



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

**Escuela Académico Profesional de Ingeniería de
Sistemas**

TESIS

**PROPUESTA DE UNA SOLUCIÓN DE REALIDAD
AUMENTADA BASADA EN TECNOLOGIA KINECT
PARA LA INTERACCIÓN EN EL PROCESO DE
PRUEBA DE PRENDAS DE VESTIR**

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

AUTOR:

BACH. INOÑAN MORALES, JOSÉ CRISTIAN

Pimentel, de 21 Febrero del 2016

Aprobación del Proyecto

“PROPUESTA DE UNA SOLUCIÓN DE REALIDAD AUMENTADA BASADA EN TECNOLOGIA KINECT PARA LA INTERACCIÓN EN EL PROCESO DE PRUEBA DE PRENDAS DE VESTIR”

Aprobación de la tesis

Zuloaga Cachay José Fortunato
Asesor metodólogo

Tuesta Monteza Víctor Alexci
Asesor especialista

Vasquez Leyva Oliver
Presidente del jurado de tesis

Celis Bravo Percy
Secretario del jurado de tesis

Tuesta Monteza Víctor Alexci
Vocal del jurado de tesis

DEDICATORIA

A mi madre que me ha guiado en la parte intelectual y siempre ha estado acompañándome con su amor y comprensión en todas las etapas de mi vida, lo que la ha convertido en mi confidente y mejor amiga con la que sé que siempre puedo contar.

A mis padres que ha guiado mis pasos por el camino del bien y que con su amor y ternura me impulsa a que sea una mejor persona cada día.

A mis asesores que ha estado presente en todo el desarrollo de mi proyecto de tesis y que me han guiado para salir adelante y no rendirme a la hora de finalizar este trabajo.

AGRADECIMIENTOS

Quiero agradecer a mis asesores del trabajo de grado Zuloaga Cachay José Fortunato y Víctor Alexci Tuesta Monteza, por sus constantes esfuerzo, dedicación, preocupación a la hora de guiarnos y paciencia por los tropiezos encontrados.

A todos los docentes que han hecho parte de nuestro proceso de formación, por brindarnos las herramientas académicas para el desarrollo de este trabajo de grado.

Y a todas las personas que de una u otra forma han hecho parte del proceso de aprendizaje por el que pasamos para obtener este trabajo con resultados satisfactorios.

ÍNDICE

ÍNDICE	iii
RESUMEN.....	IX
ABSTRACT.....	X
INTRODUCCIÓN.....	XI
CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN	1
1.1. Situación Problemática.....	1
1.2. Formulación del Problema.....	4
1.3. Delimitación de la investigación.	4
1.4. Justificación e Importancia de la Investigación	4
1.5. Limitaciones de la investigación.....	6
1.6. Objetivos de la investigación	7
CAPÍTULO II: MARCO TEÓRICO.....	8
2.1. Antecedentes de la Investigación	8
A. NIVEL INTERNACIONAL.....	8
B. NIVEL NACIONAL	10
C. NIVEL LOCAL.....	11
2.2. Estado del Arte.....	13
2.3. Bases Teórica – Científicas	21
2.4. Entorno de Desarrollo Integrado (IDE)	42
2.5. Tecnologías de programación	47
2.6. Lenguajes de programación.....	50
2.7. Metodología de Desarrollo de Software	56
2.8. Cuadro comparativo de Metodología.....	60
2.9. Metodología de desarrollo de software escogida.....	61
2.10. Normalidad	68
2.11. Evaluación Económica	80
2.12. Gestión de Riesgo de Seguridad de Información (ISO 27005)	81
2.13. Interrelación de Variables.....	84
2.14. DEFINICIÓN DE TÉRMINOS BÁSICOS	86
CAPÍTULO III: MARCO METODOLÓGICO	88
3.1. Tipo y Diseño de Investigación	88
3.2. Población y Muestra	89
3.3. Variables.....	89
3.3.1. Variable Independiente	89
3.3.2. Variable dependiente:	89
3.4. Operacionalización	90
3.5. Métodos, técnicas e instrumentos de recolección de datos.....	92
3.5.1. Métodos de la Investigación	92
3.5.2. Técnicas de Investigación	92
3.5.3. Instrumentos de recolección de datos.....	92
3.6. Procedimiento para la recolección de datos.....	93
3.6.1. Diagrama de Flujo de Procesos	93
3.6.2. Descripción de Procesos.....	95
3.7. Análisis Estadístico e Interpretación de los datos	100
3.8. Criterios éticos	100
3.9. Criterios de rigor científico.....	101
CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADO..	102
4.1. Resultado en tablas – Variable Dependiente.....	102



4.2. Contratación de la Hipótesis	107
CAPÍTULO V: DESARROLLO DE PROPUESTA	110
5.1. Propuesta.	110
5.2. Justificación.	110
5.3. Objetivos.....	111
5.3.1.Objetivos generales de la propuesta.	111
5.3.2.Objetivos específicos de la propuesta.	111
5.4. Caracterización de los Gestos de Interacción	112
5.5. Herramientas tecnológicas escogidas.	137
5.6. Metodología de desarrollo de software empleada	137
5.7. Desarrollo de la propuesta.	138
5.7.1. Aplicación Kinect	138
5.7.1.1.Etapa N° I: “Modelado del Negocio”	138
5.7.1.2.Etapa N° II: “Determinación de los Requerimientos del Software.”	144
5.7.1.3.Etapa N° III: “Análisis del Software”	147
5.7.1.4.Etapa N° IV: “Diseño de sistemas”	153
5.7.1.5.Etapa N° V: “Implementación del Sistema”	160
5.7.2. Aplicación Web.....	165
5.7.2.1.Etapa N° I: “Modelado del Negocio”	165
5.7.2.2.Etapa N° II: “Determinación de los Requerimientos del Software.”	170
5.7.2.3.Etapa N° III: “Análisis del Software”	175
5.7.2.4.Etapa N° IV: “Diseño de sistemas”	191
5.7.2.5.Etapa N° V: “Implementación del Sistema”	197
5.8. Niveles De Seguridad Y Planes De Contingencia	202
5.7.2.Creación De Usuarios	205
5.7.3.Evaluación De Riesgos Y Amenazas	207
5.7.4.Desarrollo De Planes Contingenciales	208
5.9. Componentes hardware y software	211
5.10.Análisis costo beneficio.	214
5.10.1.Inversión Inicial:	215
5.10.2.Beneficios:	218
5.11.Flujo de Caja.....	222
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES	224
6.1 Conclusiones.....	224
6.2 Recomendaciones.	225
REFERENCIAS BIBLIOGRÁFICAS.....	226
ANEXO 1.....	233
ANEXO 2.....	235
ANEXO 3.....	236
ANEXO 4.....	238
ANEXO 5.....	239
ANEXO 6.....	240
ANEXO 7.....	241
ANEXO 8.....	244



ÍNDICE DE FIGURAS

Figura 2. 1 Kinect Xbox one.....	23
Figura 2. 2 Kinect desmontada en la que se ven todas sus piezas	23
Figura 2. 3 Arquitectura de Kinect.....	25
Figura 2. 4 Gestos Innatos.....	27
Figura 2. 5 Gestos aprendidos.....	28
Figura 2. 6 Reconocimiento de voz.....	29
Figura 2. 7 Las articulaciones del esqueleto	30
Figura 2. 8 Esqueleto procedimiento de adquisición de datos.....	31
Figura 2. 9 Posición del esqueleto	32
Figura 2. 10 Realidad Aumentada	32
Figura 2. 11 Visión por Computadora	34
Figura 2. 12 Probadores	36
Figura 2. 13 Objetos Virtuales.....	37
Figura 2. 14 Metodología RUP	61
Figura 2. 15 <i>Calidad Interna y Externa de la ISO9126</i>	69
Figura 2. 16 <i>Calidad Interna y Externa de la ISO9126</i>	69
Figura 2. 17 Gestión de Riesgo de Seguridad de Información <i>ISO 27005</i>	84
Figura 2. 18 <i>Interrelación de Variables Fuente: Elaboracion Propia</i>	84
Figura 3. 1 <i>Diagrama de Flujo de Procesos</i>	93
Figura 4. 1 Grafica de ficha de usabilidad.....	104
Figura 4. 2 <i>Variable de resultados en el proceso de analisis</i>	108
Figura 5. 1: Diagrama de Caso de Uso de Negocio De API prueba de prenda de vestir.....	139
Figura 5. 2 Modelo de Objeto de Negocio seleccionar prenda de vestir.....	141
Figura 5. 3 Prueba de prenda de vestir.....	142
Figura 5. 4 Diagrama Del Modelo De Dominio del Problema prueba de prendas de vestir	143
Figura 5. 5: Modelo De Caso De Uso de Gestionar Selección de prendas	145
Figura 5. 6: Modelo De Caso De Uso De Prueba de prenda de vestir	146
Figura 5. 7: Diagrama de Colaboración seleccionar prenda de vestir	148
Figura 5. 8: Diagrama de Colaboración Probar prenda de vestir.....	148
Figura 5. 9: Diagrama de secuencia seleccionar prenda de vestir	149
Figura 5. 10: Diagrama de secuencia seleccionar prenda de vestir	150
Figura 5. 11: Diagrama de clase	151
Figura 5. 12: Diagrama de Colaboración seleccionar prenda de vestir	152
Figura 5. 13: Diagrama de patrón de diseño	152
Figura 5. 14: Modelo de datos	154
Figura 5. 15: Interfaz de Mujer	155
Figura 5. 16: Interfaz de Hombre	155
Figura 5. 17: Interfaz interacción prueba.....	156
Figura 5. 18: Interfaz de tomar fotos	156
Figura 5. 19: Interfaz de seleccionar talla	157
Figura 5. 20: Interfaz QR	157
Figura 5. 21: Interfaz Social Media	158



Figura 5. 22: Interfaz de ajuste de prenda 1.....	158
Figura 5. 23: Interfaz ajuste de prenda 2	159
Figura 5. 24: Diagrama de componentes de esquema general.....	161
Figura 5. 25: Diagrama de componentes de esquema general.....	162
Figura 5. 26: Diagrama de arquitectura de componentes.....	163
Figura 5. 27: Diagrama Componentes de Hardware	164
Figura 5. 28 Diagrama de Caso de Uso de Negocio De Web	166
Figura 5. 29 Modelo de Objeto de Negocio vender producto	168
Figura 5. 30 Modelo de Objeto de Negocio reponer stock	168
Figura 5. 31 Modelo de Objeto de Negocio realizar entrega	169
Figura 5. 32: Diagrama Del Modelo De Dominio del Problema ventas de prendas de vestir	169
Figura 5. 33: Modelo De Caso De Uso de Gestionar venta.....	171
Figura 5. 34: Modelo De Caso De Uso facturación Fuente:	171
Figura 5. 35 Modelo De Caso De Uso de gestión almacén.....	172
Figura 5. 36 : Modelo De Caso De Uso de gestión envío.....	173
Figura 5. 37: Modelo De Caso De Uso de gestión logistica	173
Figura 5. 38 Modelo De Caso De Uso de gestión marketing.....	174
Figura 5. 39 : Diagrama de Colaboración consultar producto	176
Figura 5. 40 : Diagrama de Colaboración datos del cliente - visualziar	176
Figura 5. 41: Diagrama de Colaboración datos del cliente - modificar.....	177
Figura 5. 42: Diagrama de Colaboración datos del cliente - registrar	177
Figura 5. 43: Diagrama de Colaboración gestionar compra	178
Figura 5. 44: Diagrama de Colaboración compra - visualziar.....	178
Figura 5. 45: Diagrama de Colaboración compra pedido	179
Figura 5. 46: Diagrama de Colaboración forma de pago.....	179
Figura 5. 47: Diagrama de Colaboración identificar usuario.....	180
Figura 5. 48: Diagrama de Colaboración gestionar producto	180
Figura 5. 49 : Diagrama de Colaboración producto - visualizar	181
Figura 5. 50 Diagrama de Colaboración producto - modificar	181
Figura 5. 51 : Diagrama de Colaboración categoría producto	182
Figura 5. 52: Diagrama de secuencia consultar producto	183
Figura 5. 53: Diagrama de secuencia datos cliente - visualizar.....	184
Figura 5. 54 Diagrama de secuencia datos cliente - modificar	184
Figura 5. 55: Diagrama de secuencia datos cliente registrar.....	185
Figura 5. 56 Diagrama de secuencia compra.....	185
Figura 5. 57: Diagrama de secuencia compra- visualizar.....	186
Figura 5. 58: Diagrama de secuencia compra pedido	186
Figura 5. 59: Diagrama de secuencia forma de pago	187
Figura 5. 60 Diagrama de secuencia identificación de usuario	187
Figura 5. 61: Diagrama de secuencia producto.....	188
Figura 5. 62 Diagrama de secuencia producto visualizar	188
Figura 5. 63: Diagrama de secuencia producto - modificar	189
Figura 5. 64: Diagrama de secuencia categoría producto	189
Figura 5. 65: Diagrama de clase	190
Figura 5. 66 Modelo de datos	192
Figura 5. 67: de resumen de compra	193
Figura 5. 68: Interfaz registrar o ingresar usuario.....	193
Figura 5. 69: Interfaz de llenado de datos del usuario.....	194



Figura 5. 70: Interfaz de llenado de datos del usuario.....	194
Figura 5. 71: Interfaz de datos de dirección	195
Figura 5. 72: Interfaz de resumen de compra	195
Figura 5. 73: Interfaz de cuenta de usuario.....	196
Figura 5. 74: Interfaz de agregar prendas en aplicación	196
Figura 5. 75: Interfaz de selección de prendas a agregar	197
Figura 5. 76: Diagrama de componentes de esquema general.....	198
Figura 5. 77: Diagrama de componentes de almacén.....	199
Figura 5. 78: Diagrama de componentes de ventas.....	200
Figura 5. 79: Diagrama de desplique	201



ÍNDICE DE TABLAS

Tabla 2. 1: Cuadro comparativo de Kinect 1.0 y 2.0	22
Tabla 2. 2: Entornos de Desarrollo	43
Tabla 2. 3: Comparación de Metodologías.....	60
Tabla 3. 1 Variables Dependientes e Independiente.....	90
Tabla 3. 2 Variables Dependientes e Independiente.....	101
Tabla 4. 1 Resultado en tablas y gráficos Variable independiente.	102
Tabla 4. 2: Resultado en tablas y gráficos Variable independiente.	107
Tabla 4. 3 análisis de variable de frecuencia observadas.	108
Tabla 4. 4 análisis de variable de frecuencia deseadas	108
Tabla 4. 5 Tabla de distribución del CHI-CUADRADO	109
Tabla 5. 1 Tabla tecnologías escogidas.....	137
Tabla 5. 2: Casos de uso de prueba de prenda de vestir	140
Tabla 5. 3: Casos de uso de venta de prendas de vestir.....	167
Tabla 5. 4: Costos de servicios	215
Tabla 5. 5: Costos de materiales	215
Tabla 5. 6: Costos de Personal.....	216
Tabla 5. 7: Licencias de Software de Desarrollo e Implementación	217
Tabla 5. 8: Costo de hardware para la Implementación	217
Tabla 5. 9: Gastos operativos I	218
Tabla 5. 10: Gastos operativos II	218
Tabla 5. 11: Ahorro de personal.....	219
Tabla 5. 12: Ahorro de probadores	220
Tabla 5. 13: Flujo de caja	222



RESUMEN

La presente investigación pretende desarrollar una solución de realidad aumentada basada en tecnología Kinect para la prueba de prendas de vestir, haciendo uso de las manos y gestos más adecuados como forma de entrada para controlar comandos.

La problemática responde la necesidad de encontrar una manera sencilla y natural de comunicarse con un entorno inteligente. En el mundo de la inteligencia tecnológica cada vez se pone más esfuerzo en que la interacción con el usuario sea lo más simple posible, ya que se desea que sea algo poco intrusivo, algo que no requiera aprendizaje. Por ello se decide hacer una interfaz de usuario natural usando el dispositivo Kinect para la prueba de prendas de vestir.

Para el desarrollo se consideró dos plataformas de desarrollo: web y escritorio. La aplicación web se encargará principalmente del comercio electrónico (venta de ropa), mientras que la aplicación de escritorio; de la prueba de prendas de vestir.

La tecnología utilizada para este proyecto son lenguaje de programación C#, Visual Studio 2013 y SDK Kinect 2.0. También se uso HTML, JSP, JAVASCRIPT, AJAX e IDE NETBEANS

Palabras Clave: Interacción, Kinect, prendas de vestir, prueba



ABSTRACT

This research aims to develop a solution based on augmented reality technology to the Kinect test of clothing, making use of the hands and gestures more suitable as input form to control commands.

The problematic answers the need to find a simple and natural way to communicate with a smart environment. In the world of intelligence technology is increasingly puts more effort in which the interaction with the user is as simple as possible, since it is you want it to be somewhat intrusive, something that does not require programming. For this reason we decide to make a natural user interface using the device to the Kinect test of clothing.

For the development was considered two development platforms: web and desktop. The web application will primarily be responsible of the e-commerce (selling clothes), while the desktop application; the test of clothing.

The technology used for this project are programming language C#, Visual Studio 2013 SDK and Kinect 2.0. Also use HTML, JSP, Javascript, AJAX and Netbeans IDE.

Key Words: Kinect, interaction, clothing, testing.



INTRODUCCIÓN

En la presente investigación se desarrolló una solución de entono natural para la interacción en el proceso de prueba y venta de prendas de vestir”.

Este trabajo se desarrolló basándose en la realidad aumentada (realidad física con realidad virtual) y tecnología Kinect, que permita al usuario interactuar con una aplicación de una manera más directa, ya que muchas veces, los dispositivos como el teclado, mouse no cumplen con los requisitos de una interacción más natural y adecuada.

Para el desarrollo de esta aplicación se ha utilizado el lenguaje de programación C#, ya que es más sencillo e intuitivo, más moderno y orientado a objetos (obviamente) y sobretodo muy eficiente, desde un punto de vista práctico C# es el lenguaje de .NET, por ese motivo se escogió el programa Visual Studio 2015

También se usó el SDK Kinect 2.0, ya que da libertad a sacar todo el potencial del Kinect de Xbox One, en una interfaz natural.

Además, se desarrolló un sistema web acoplado a la aplicación principal para la venta de ropa hacia el cliente, la cual permitirá agilizar



la compra de una prenda de vestir. Utilizando para ello el lenguaje de programación HTML, JSP, JAVASCRIPT, AJAX y IDE NETBEANS.

La aplicación desarrollada en este proyecto ha dado como resultado el desarrollo de una aplicación con interfaz Natural de Usuario (o NUI, Natural User Interface), que permite interactuar con las aplicaciones sin necesidad de utilizar mandos o dispositivos de entrada de datos como el ratón o el teclado, sino que se basa en movimientos gestuales de las manos o el propio cuerpo como forma de comunicación para la prueba de prendas de vestir. Y el sistema web dio como resultado una herramienta que permite agilizar la venta de una prenda y así utilizar la tecnología como un esfuerzo a hacer la vida del usuario (comprador) más fácil.

Para el análisis de los resultados se utilizó una ficha de evaluación para la interacción de probarse una prenda, la cual nos permite demostrar que el desarrollo de la API con sus funciones fue la correcta, haciendo que la interacción sea natural y rápida.



CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

1.1. Situación Problemática

El avance de la tecnología ha permitido que las aplicaciones vayan evolucionando con el tiempo y puedan tener mayor colaboración con el usuario. Con las nuevas formas de interacción en pantallas más grandes, interfaces en ambientes virtuales, etc., utilizar un teclado o un ratón no son los más adecuados.

Los dispositivos de entrada con que se interactúan también necesitan ser diferentes para adaptarse al entorno nuevo de realidad aumentada y realidad virtual. Aunque el teclado y el ratón se han vuelto inalámbricos, es poco útil en ciertas ocasiones como cuando uno se encuentra de pie y trabajando de manera colaborativa en un mismo entorno.

La utilización de un dispositivo que sense el movimiento natural de las personas, como señas y gestos de las manos, es conveniente como medio de comunicación con la aplicación. Con esto se sugiere una nueva interface para interactuar con el computador que es por medio de gestos.

Por lo tanto, la problemática responde la necesidad de encontrar una manera sencilla y natural de comunicarse con un entorno inteligente. Sin la necesidad de comunicarse mediante dispositivos como el teclado,



mouse o comando de mano, también que no se requiera aprendizaje previo para la prueba de prendas de vestir.

En la actualidad, existen diversos proyectos basados en optimizar la interacción de computador – usuario, como veremos a continuación.

A NIVEL INTERNACIONAL

En el proyecto A Consumer Level 3D Object Scanning Device using Kinect for Web-based C2C Business (Geoffrey POON, 2014) permite a los usuarios escáner objetos 3D para colocarlos en la web y así promover su producto a la venta, obteniendo una importante atención por parte de los clientes en el comercio electrónico, especialmente en el C2C.

En Kinect Banking App (Etronika, 2012), proporciona una aplicación de banca con control de movimientos para la administración de sus finanzas (pagar, transferir dinero, balance y otros) con solo gestos de la mano o control de voz.

En Kinect Enabled Shopping Cart (Mohta, 2012) proporciona una aplicación de carrito de compra con Kinect, esta aplicación escanea la tarjeta de membresía con solo colocarla en el carrito

de compra, para así reconocer al usuario comprador, luego el usuario va ingresando al carrito de compras todo lo que desea adquirir, la aplicación va escaneando y sumando para luego al finalizar el cliente la compra, la aplicación cargue todo lo comprado a la tarjeta.

A NIVEL NACIONAL

En el proyecto Vitrinas interactivas Kinect (Martinez, 2012) proporciona una interfaz de reconocimiento de gestos, utilizando el sensor Kinect para proporcionar a los usuarios una interacción de ver los productos y cuáles son sus precios.

En este otro proyecto Probador Virtual Kinect (Martinez, 2012) proporciona una aplicación con interfaz natural para que el usuario pueda interactuar con la aplicación en la prueba de prendas de vestir.

Si bien la problemática de interacción natural ya ha sido abordada con anterioridad por otros proyectos mencionados anteriormente, ninguno de estos es para el proceso de prueba y compra de prendas de vestir la cual conlleva a que muchos de los usuarios (clientes) no tengan una óptima tecnología para la interacción en el proceso de prueba de prendas de vestir.



1.2. Formulación del Problema

¿Cómo desarrollar una solución que permita apoyar el proceso de interacción de prueba de prendas de vestir?

1.3. Delimitación de la investigación.

1.3.1. Delimitación Espacial.

La tesis se desarrollará en el distrito de Chiclayo.

1.3.2. Delimitación Temporal

El estudio se ejecutará en un solo periodo en el 2015, influido por el tipo de diseño cuasi-experimental.

1.3.3. Delimitación Cuantitativa.

Elaboración de 1 tesis en el 2015.

1.4. Justificación e Importancia de la Investigación

Al analizar el problema de cómo poder comunicarse con una aplicación en ambiente virtual mediante una interacción más natural, el seguir utilizando dispositivos que requieren en algunos casos que la persona esté sentada, o que apoye los dispositivos sobre una superficie, no es muy útil en especial en entornos de visualización de tres dimensiones. Para esto hay otros métodos de interacción que se pueden utilizar, como el de un dispositivo que sense el movimiento de las manos para interactuar con las aplicaciones en 3D, implementando un conjunto de

funcionalidades que faciliten el desarrollo de gestos para este tipo de aplicaciones.

La realización del presente estudio nos permite plantear una propuesta de solución a las necesidades actuales y futuras en el proceso de interacción de prueba de prendas de vestir, aportando mejoras en los aspectos:

Justificación Tecnológica

El uso de una aplicación de este tipo permitirá que los procesos de interacción natural de prueba y compra de prendas de vestir sean más acordes a la tecnología usada, lo cual origine generar un ahorro de tiempos al personal y a los usuarios en general, dando a la empresa una imagen de una institución moderna e innovadora.

Justificación Económica

Con la presente aplicación se logrará aumentar las ganancias para la empresa, debido a que el alcance de la aplicación permitirá llegar a un mayor número de clientes. La mejora económica se verá en el incremento de ventas.

Justificación Social:

Con la implementación del nuevo modelo de ventas, se logra brindar un mejor y eficiente servicio a los clientes en la cual aumentara la capacidad de atención de clientes, reducir esfuerzos en las actividades diarias y atención de los servicios.

1.5. Limitaciones de la investigación

En la investigación corresponden al sensor Kinect, el cual tiene varias que hacen que la profundidad de ciertas regiones de la escena no se pueda estimar o si se estima, la fiabilidad de los datos no es aceptable. Estas limitaciones vienen condicionadas tanto por factores internos, debidos a la arquitectura del dispositivo; como externos, debidos a la naturaleza de la escena.

1.5.1. Limitación tecnológica

- a) Falta de hardware (adaptador Kinect).
- b) Cuando hay un cambio muy fuerte de luminosidad (luz solar) impide que la luz infrarroja pueda ser detectada por el Sensor IR.
- c) La inclinación de la superficie de los objetos respecto al proyector del haz de luz limita la detección de la profundidad.
- d) Si el rayo de luz es casi paralelo a la superficie no podrá incidir sobre la misma, haciendo imposible la estimación de la profundidad.

1.6. Objetivos de la investigación

Objetivo General

Desarrollar una solución de Probador Virtual basada en tecnología Kinect para apoyar al proceso de interacción de prueba de prendas de vestir.

Objetivos Específicos

1. Caracterizar los gestos que se usaran para la comunicación.
2. Desarrollar una aplicación bajo la tecnología Kinect, intuitiva y rápida en la NUI.
3. Desarrollar un sistema web de venta que interactúe con la aplicación.
4. Analizar resultados.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de la Investigación

A. NIVEL INTERNACIONAL

A. DESARROLLO DE UN PROTOTIPO DE UN API PARA LA INTERACCIÓN DE UN USUARIO CON APLICACIONES CON CONTENIDO 3D UTILIZANDO KINECT (López, 2013)

En el presente proyecto se presento una aplicación que permite una interacción con el computador, haciendo uso de las manos y gestos como forma de entradas para generar comandos.

El sistema es desarrollado utilizando la tecnología Kinect de Microsoft y como librería de interacción se ha utilizado **Openkinect, Openframeworks, OpenCV, OGRE 3D, QT, OpenSceneGraph**. Para integrar todo ello y realizar la implementación, se ha utilizó el entorno de desarrollo **Visual Studio 2010 C++ Express Edition**.

Este sistema tuvo como objetivo principal apoyar la interacción en el movimiento de objetos 3D

Como conclusión se puede decir que se obtuvo mejor precisión, sensibilidad y control de la aplicación y con respecto al Kinect los gestos implementados fueron los adecuados para la aplicación porque todos los que realizaron las pruebas estuvieron de acuerdo que

recordarían fácilmente los movimientos realizados si volvieran a interactuar con la aplicación.

B. SISTEMA DE ENTRENAMIENTO CON KINECT (Estévez, 2013)

Este proyecto ha desarrollado un sistema de entrenamiento, usando la captura de movimientos a través de la Kinect, para un deporte en el que los movimientos y posturas son una parte esencial como es el **Karate**.

Para el desarrollo de este sistema se utilizó el sensor Kinect, SDK Kinect, Kinect Studio, WPF y visual Studio 2010 C#

Como resultado es que al utilizar el sistema permitirá el reconocimiento de movimientos básicos y **katas** (en karate un kata es una secuencia establecida de golpes y bloqueos lanzados al aire).

En conclusión, este sistema ayuda aquellas personas que practican Karate puedan entrenar determinados katas usando tecnologías actuales (la Kinect en este caso), así como iniciar a las personas interesadas en este deporte.

C. KINECT SENSOR BASED PC CONTROL INTERFACE FOR HANDICAPPED USERS (Lee San Hyuk, 2013)

En el presente proyecto se creó un interfaz para el reconcomiendo de voz y movimientos lo cual proporcionar a los usuarios con discapacidades física poder controlar la PC sin necesidad de utilizar sus manos o brazos.

Para el desarrollo de este sistema se utilizó el sensor Kinect, SDK Kinect y visual Studio 2010.

Como resultado y conclusión es que esta API permite a los usuarios discapacitados poder interactuar con la PC, sin la necesidad de usar un teclado o mouse.

B. NIVEL NACIONAL

A. EFECTO DEL USO DEL SENSOR KINECT PARA MEJORAR LA ATENCIÓN Y CONCENTRACIÓN EN LOS NIÑOS DEL TERCER GRADO “A” DE LA INSTITUCIÓN DIVINO MAESTRO – MOLLEPAMPA, CAJAMARCA

En el presente proyecto se desarrolló una aplicación llamada Kinect Attention que incluye 4 juegos, parejas ocultas, Memoria, Sucesión de Imágenes, Asociación de Imágenes donde el jugador no necesita del teclado, mouse o joystick para interactuar con la aplicación haciendo de esta forma una experiencia de juego más divertida para los niños.

Utilizó el sensor Kinect, el IDE Visual Studio 2012 Express, Microsoft .Net Framework 4, la metodología Scrum.

Como resultado y conclusión se logró con el objetivo planteado, de mejorar la atención y concentración en los niños de tercer grado "A" de la I.E. Divino Maestro.

Así mismo es muy importante para la presente tesis porque nos ayuda a definir las herramientas a utilizar en su desarrollo.

C. NIVEL LOCAL

A. REALIDAD AUMENTADA CON KINECT PARA CONTRIBUIR EN EL PROCESO DE TERAPIAS DE LOS PACIENTES CON PARÁLISIS CEREBRAL INFANTIL (López D. K., 2014)

En este proyecto se desarrolló un software interactivo utilizó la metodología Extreme Programig, el Sensor Kinect, logrando una constante interacción con el usuario para determinar las funcionalidades del juego que mejor se adapte a la realidad.

Como resultado final se obtuvo un producto que ayuda ampliar la cantidad de ejercicios terapéuticos específicos para las extremidades superiores, involucrando en cada ejecución de movimientos la activación de zonas cerebrales como las visuales, auditivas y sensitivas de los pacientes contribuyendo a la eficacia del tratamiento,

además de aumentar la duración de la sesión y brindar información relevante de lo que el paciente realizó en su sesión para su posterior evaluación.

Relevante para la presente tesis como ayuda para definir la metodología a utilizar.

B. DESARROLLAR UNA APLICACION BASADA EN REALIDAD AUMENTADA COMO MATERIAL DE APRENDIZAJE DEL CUERPO HUMANO CON TECNOLOGIA MOVIL ANDROID (Neria, 2014)

Este proyecto pretende crear Realidad Aumentada en un dispositivo móvil actual para la ayuda al aprendizaje del cuerpo humano. Para el desarrollo de este API se utilizó modelos 3D diseñados en maya, desarrollados en Metaio SDK y codificado a través del SDK Android, usando Eclipse.

El resultado es que la aplicación innova muchísimo respecto a otras API sobre el cuerpo humano por el tratamiento de las capas virtuales, además que realiza un estudio de la realidad aumentada en dispositivos móviles.

En conclusión, la implementación de una aplicación de RA ha servido para migrar un proyecto que originalmente estaba diseñado para equipos de sobremesa.

2.2. Estado del Arte

Realidad Aumentada.

Tecnología que combina técnicas de reconocimiento de formas y visualización 3D para añadir virtualidad a una imagen real de forma coherente y en función de la localización de la escena real. (Serrano Mamolar, 2012)

Se dice que la RA es un híbrido entre el mundo real y el mundo virtual. Paul Milgram clasificó por primera vez los distintos espacios de realidad “mixta” desde el punto de vista de continuidad del contexto. Este concepto describe que existe una escala continua entre lo completamente real y lo completamente virtual. Entre ambos existe la virtualidad aumentada (está más próxima al entorno virtual) y la RA (más próxima al entorno real). (Serrano Mamolar, 2012)

Evolución de la Realidad Aumentada

1992.

Tom Caudell y David Mizell acuñan el término “realidad aumentada” para referirse a la superposición del mundo real con información generada por ordenador. IBM desarrolla el primer smartphone que saldría al mercado en 1993. El teléfono tenía 1MB de memoria y una pantalla táctil en blanco y negro de 160x293 píxeles de resolución. (Caudell, 1992)



1999.

Kato y Billinghurst presentan ARToolKit, una librería de seguimiento con 6DOF, utilizando markers para el reconocimiento de patrones. ARToolKit está disponible como código abierto bajo la licencia GPL y es todavía muy popular en la comunidad RA. Hollerer et al. Presentan el primer sistema de RA móvil basado en GPS y sensores inerciales. (Kato, 1999) Scale Invariant Feature Transform (SIFT), metodología que fue desarrollada por David Lowe y sus estudiantes.

2000.

Julier. Presenta BARS (Battlefield Augmented Reality system). El sistema consiste en un sistema portátil con conexión wifi y HMD. El sistema muestra de forma virtual una escena de batalla con información adicional sobre la infraestructura del entorno y sobre posibles enemigos. (Technology, 2000)

2004.

Mohring. Presenta un sistema para el posicionamiento con marcadores 3D en teléfonos móviles. Rohs y Gfeller presentan Visual Codes, un sistema de marcadores 2D para teléfonos móviles. Estos marcadores pueden utilizarse sobre objetos físicos para superponer información virtual sobre dicho objeto. (Möhring, 2004)

2009.

Kimberly Spreen. Desarrolla ARhrrr!, el primer videojuego de RA con una calidad gráfica al nivel de los juegos comerciales. Esta aplicación utiliza el kit de desarrollo Tegra de Nvidia, optimizado para las GPU's del momento. (Lab, 2010)

2013.

Google desarrolla las Google Glass Explorer Edition (gafas de realidad aumentada (Head-mounted display, HMD)) fueron lanzadas para los desarrolladores de Google I/O el presente año, mientras que la versión para consumidores estará lista en el 2014. Su propósito es mostrar información, como un teléfono inteligente, con acceso a internet mediante órdenes de voz pero con Realidad Aumentada. (Inc, 2013)

2015.

Microsoft desarrolla los **Holo Lens** (casco inalámbrico que crea ilusiones holográficas en el campo de visión del usuario y además permite "tocarlas") se espera que esté listo a finales de 2015. Su propósito no es solo mostrar información, como un teléfono inteligente, con acceso a internet mediante órdenes de voz pero con Realidad Aumentada. Holo Lens no solo crea realidad aumentada, sino que también permite interactuar con los objetos virtuales que están en el campo de visión. (Mundo, 2015)

Kinect

Kinect es un dispositivo creado por Microsoft que originalmente se diseñó para la consola de videojuegos Xbox con el propósito de que los jugadores pudiesen interactuar con la consola de una manera más natural e intuitiva de manera que la jugabilidad fuese mucho más ágil y los juegos no requiriesen apenas de aprendizaje por parte de los usuarios para poder dominarlos. (Kinect, 2014)

Este dispositivo consta de tres partes diferenciadas. Un sistema de detección de profundidad a base de un sensor de rayos infrarrojos, una cámara de vídeo RGB y un array de cuatro micrófonos para reconocimiento de voz. Gracias al sistema de profundidad y a la cámara RGB el dispositivo es capaz de reconocer hasta seis jugadores y localizarlos en un espacio delante del dispositivo posicionándolos en un sistema de ejes X, Y, Z pudiendo además hacer un seguimiento del esqueleto de dos de los jugadores posicionando frente al dispositivo Kinect. (Kinect, 2014)

Evolución del KINECT

Kinect 1.0

El 3 de junio de 2009 se lanzó la primera versión del Kinect, en lo cual muchos vieron en él una simple respuesta de los de Redmond ante el indudable éxito que Nintendo estaba cosechando con Wii y su sistema de control. El dispositivo en forma de barra usaba un

proyector infrarrojo y una cámara que escaneaban la escena y enviaban la información a un microchip especialmente preparado para capturar en tres dimensiones el movimiento de objetos y personas. A ellos se les unía una fila de micrófonos capaz de reconocer la voz del usuario. Todos estos elementos en conjunto permitían la captura de movimiento en 3D junto al reconocimiento facial, de gestos y de voz. (windows, 2013)

Para tal tarea las especificaciones de Kinect no suponían nada del otro mundo. La cámara tenía resolución VGA y actuaba a 640×480 por defecto, aunque era capaz de trabajar a 1280×1024 píxeles a costa de una menor tasa de refresco. El microchip incluido apenas realizaba parte del trabajo de procesar la información dejando buena parte de la tarea a la propia consola. (windows, 2013)

Kinect 2.0

El 22 de noviembre de 2013 fue el lanzamiento del Kinect 2.0. La gran diferencia del nuevo Kinect respecto a su predecesor reside en la nueva cámara principal. La segunda generación del dispositivo de captura de movimientos **incorpora una cámara time-of-flight (TOF) de alta resolución** que permite al próximo Kinect de Xbox One capturar más detalles con gran precisión y mayor resolución. El nuevo modo de profundidad proporcionado

por esta cámara TOF permite reproducir una escena con tres veces más fidelidad que el primer Kinect. (windows, 2013)

No es la única ventaja de utilizar este tipo de cámara. Con ella además se logra un campo de visión un 60% más grande, lo que permite registrar un espacio mayor y posibilita que más personas puedan ser registradas al mismo tiempo y a una menor distancia del dispositivo. Con la nueva consola pueden aparecer hasta 6 personas en escena reconociendo y distinguiendo todos sus movimientos. Es un avance importante respecto a su predecesora que sólo era capaz de registrar el movimiento de 2. (windows, 2013)

El segundo gran cambio en la nueva generación de Kinect viene de la mano del **nuevo sensor de infrarrojos** que logra reconocer objetos y personas en condiciones de muy poca luz. El sensor es ahora tan potente que permite identificar elementos en una habitación totalmente a oscuras. La precisión es tal que puede reconocer gente y registrar cuerpos incluso sin ninguna luz visible para el ojo humano. Con poca luz reconoce la pose de la mano hasta a cuatro metros de distancia, distinguiendo con precisión cada uno de los dedos. (windows, 2013)

E-commerce

Siguiendo la definición operativa del EUSTAT, Business, como actividad que incluye el eCommerce, es el conjunto de

transacciones comerciales realizadas mediante Internet con el fin de comprar o vender productos o servicios, sea entre empresas, familias, personas, gobiernos u otros organismos públicos o privados. Los pedidos se efectúan por Internet, en tanto que el pago y la entrega, puede ser online u offline. (telecos.cnmc.es, 2011)

El eCommerce aporta numerosas ventajas para las empresas que la practican o utilizan como son las mejoras en la distribución, en las comunicaciones comerciales, en los planes de operaciones, en el marketing, en la fidelización de clientes, en el acceso geográfico global, en la logística, en los soportes técnicos, en el incremento de las interacciones internacionales, etc. (telecos.cnmc.es, 2011)

Origen y evolución histórica

A principio de los años 1970, aparecieron las primeras relaciones comerciales que utilizaban una computadora para transmitir datos, tales como órdenes de compra y facturas. Este tipo de intercambio de información, si bien no estandarizado, trajo aparejado mejoras de los procesos de fabricación en el ámbito privado, entre empresas de un mismo sector.

A mediados de 1980, con la ayuda de la televisión, surgió una nueva forma de venta por catálogo, también llamada venta directa. De esta manera, los productos son mostrados con mayor realismo, y con la dinámica de que pueden ser exhibidos

resaltando sus características. La venta directa es concretada mediante un teléfono y usualmente con pagos de tarjetas de crédito. (Guevara, 2014)

En 1995 los países integrantes del G7/G8 crearon la iniciativa Un Mercado Global para PYMEs, con el propósito de acelerar el uso del comercio electrónico entre las empresas de todo el mundo durante el cual se creó el portal pionero en idioma español Comercio Electrónico Global. (Guevara, 2014)

A finales de los años 1990, el comercio a través de Internet creció de manera considerable. (Guevara, 2014)

En el año 2000 se estrenó la primera web de compra colectiva y en 2008 nació el concepto de compra flash, gracias a la venta online de cupones de descuento en base a la geolocalización. Una iniciativa que en 2004 intentó Woot.com, con la promoción diaria de ofertas. (Guevara, 2014)

En el año 2014, Amazon y Twitter unían fuerzas para permitir que los consumidores comprar y añadir productos a sus cestas online mediante tuits. Además, PayPal se convierte el primer medio de pago del mundo que adopta la tecnología de autenticación de huella dactilar de Samsung.

Estos hechos junto con las compras a través de los dispositivos móviles y las tablets han permitido el boom de las compras online tal y como las conocemos hoy en día. (Guevara, 2014)

2.3. Bases Teórica – Científicas

A. Kinect

El sensor Kinect (Dept, 2012) es nuestra principal herramienta del proyecto, es en la que nos basamos para llevarlo a cabo.

Kinect conocido en sus principios como "Proyecto Nata", creado por el brasileño Alex Kipman y desarrollado por la compañía Microsoft para su videoconsola Xbox 360. Kinect permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador de videojuegos tradicional, mediante una interfaz natural de usuario que reconoce gestos, comandos de voz, y objetos e imágenes.

El sensor Kinect (Tapia, 2010) estaba destinado únicamente a su utilización junto a la videoconsola. Para extraer la información tan valiosa que obtiene, hubo que recurrir a ingeniería inversa. Con este proceso se pudo empezar a utilizar el dispositivo con ordenadores para diferentes aplicaciones como es el reconocimiento de objetos.

Kinect (2012) dispone de una serie de sensores que le permiten percibir el mundo a detalle. Ubicados a lo largo de una barra horizontal.

Cuadro comparativo entre Kinect 1.0 y 2.0

Funciones	Kinect 1 (Xbox 360, PC)	Kinect 2 (Xbox One, PC)
Resolución	640x480 30 fps 4:3	1920x1080 30 fps 16:9
Ángulos de visión	57 grados horizontal, 43 grados vertical	70 grados horizontal, 60 grados vertical
Distancia mínima de uso	1.82 metros	1.37 metros
IR Activo (visión nocturna)	No	Sí
Latencia	102 ms	20 ms
Ajuste manual del motor	Sí	No
Detección simultánea de personas	4	6
Puntos del cuerpo simultáneos	20	25
Detección de dedos y muñecas	No	Sí
Detección de músculos	No	Sí
Medidor de pulsaciones	No	Sí

Tabla 2. 1: Cuadro comparativo de Kinect 1.0 y 2.0 Fuente: (123kinect, 2014)





Figura 2. 1 Kinect Xbox one Fuente: (ifixit.com, 2013)



Figura 2. 2 Kinect desmontada en la que se ven todas sus piezas Fuente: (Zahumenszky, 2013)

Breve descripción del Kinect Xbox one: (Montserrat, 2015)

Son muchas las diferencias que hay con su antecesor y muchas las posibilidades de desarrollo que se generan con ellas. A continuación, destacamos las más significativas a nivel de desarrollo.



- **Mayor campo de visión.** 70° en horizontal (antes 57°) y 60 en vertical (antes 43°).

Esto permite poder detectar a más personas dentro de un mismo campo de visión. Hasta 6 jugadores pueden ser detectados simultáneamente.

A destacar que esta versión de Kinect no tiene motor de inclinación.

- **Mayor resolución. 1920 x 1080 Full HD** (antes 640 x 480).

Más resolución, más detalle.

Permite detectar con más precisión todo el entorno.

Capacidad de diferenciar la orientación del cuerpo incluyendo sus manos y pudiendo diferenciar sus dedos.

El face tracking tiene mucho más detalle y permite captar los gestos de la cara.

Más calidad de imagen.

- **Mejora el rango de profundidad del sensor.**

El rango de actuación pasa a ser de 0,5 a 4,5 metros.

- **USB 3.0**

Al aumentar la velocidad de la comunicación con el ordenador los datos fluyen más rápido y esto disminuye la latencia del sensor. Pasa de 90ms a 60ms.

- Mejora de la captación de sonidos.

Esta versión de Kinect viene dotada de una gran mejora en cuanto al reconocimiento de voz y la captación de sonidos. Se ha mejorado la eliminación del ruido ambiente y esto permite captar con más detalle las instrucciones vocales.

- Captación de movimiento a oscuras.

Ahora Kinect 2 es capaz de reconocer y captar los movimientos, aunque la sala este a oscuras.

- Kinect 2 permite calcular/analizar la fuerza de nuestros músculos y medir el ritmo cardíaco.

Aunque no lo hemos probado empíricamente. Hay que destacar y remarcar estas dos nuevas posibilidades que nos ofrece Kinect 2 y que, en un futuro, no tan lejano, nos aportaran nuevas experiencias de interacción sorprendentes.

Arquitectura Kinect 2.0

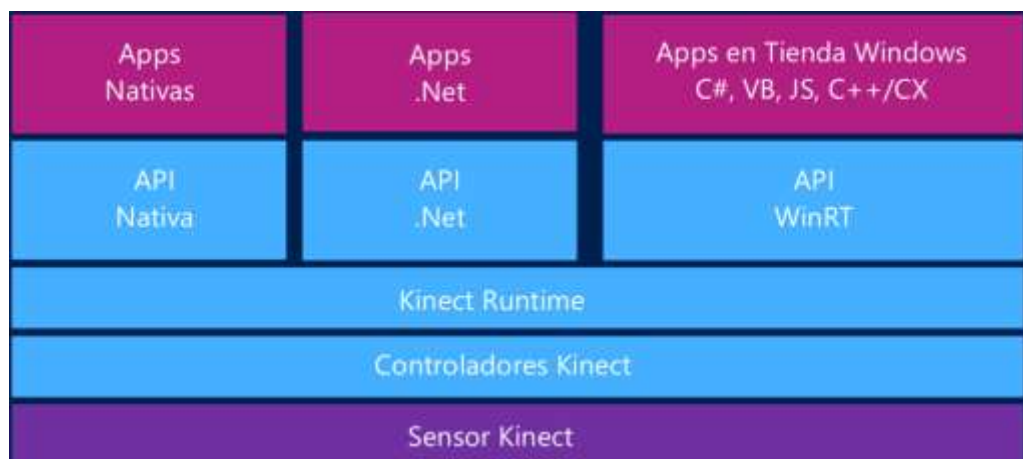


Figura 2. 3 Arquitectura de Kinect Fuente: (msdn.microsoft, 2012)



B. Interacción Kinect - Usuario

Gesto y voz son dos entradas nuevas y emocionantes que Kinect pone a su disposición. Las secciones siguientes describen cómo trabajan, cómo mejor utilizarlas, desafíos para esperar y cómo retroalimentar con sus usuarios que entiendan. (ABREGO, 2012)

Gesto

Gesto es un divertido método de interacción para explorar, pero también presenta retos difíciles que deben abordarse con el fin de hacer una interfaz natural, utilizable y adecuada para todos los usuarios de su aplicación. Esta sección cubre algunos tipos básicos de gesto, cómo dar a los usuarios ver los comentarios acerca de lo que el Kinect y algunas consideraciones de diseño de interacción de gesto importante. (ABREGO, 2012)

Tipos de gestos empleados

Gestos pueden tomar muchas formas, uso de la mano al destino algo en la pantalla, concretas, aprendido patrones de movimiento, para largos recorridos de movimiento continuo. A continuación, se presentan algunos ejemplos de tipos de gestos comúnmente utilizados y las descripciones que le ayudará a entender los gestos que se está diseñando. (ABREGO, 2012)

Gestos innatos

Gestos innatos son los que el usuario sabe intuitivamente o eso tiene sentido basado en la comprensión de los usuarios del mundo. Ejemplos

1. Señalar al objeto
2. Agarrar para coger
3. Presionar para elegir

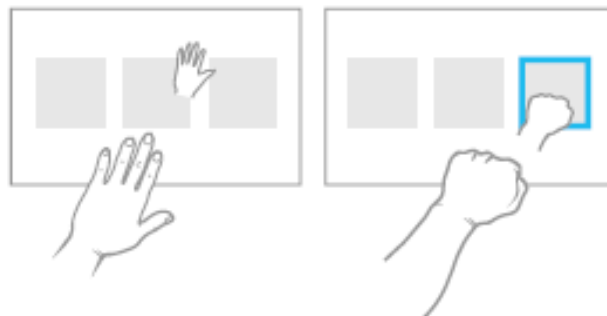


Figura 2. 4 Gestos Innatos Fuente: (ABREGO, 2012)

Gestos aprendidos son los que se les debe enseñar al usuario con el fin de saber cómo utilizarlos para interactuar con el sistema. Ejemplos

1. Agitar a participar
2. Hacer una pose específica para
3. cancelar una acción

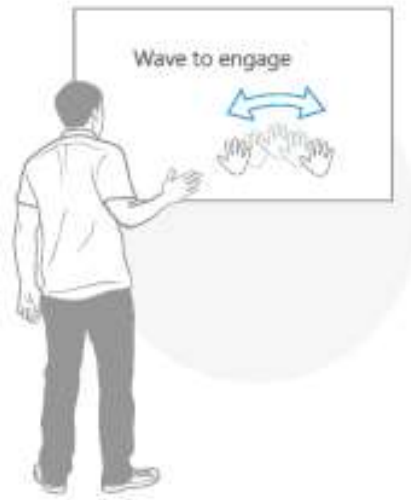


Figura 2. 5 Gestos aprendidos Fuente: (ABREGO, 2012)

Reconocimiento De Voz

La voz es otra entrada que permite tener al usuario experiencias nuevas y una sensación natural, pero es todavía limitado en cuanto a lo que es compatible. El uso de voz en alguna aplicación permite elegir determinadas palabras o frases para escuchar y utilizar como disparadores. Palabras o frases habladas como comandos no es conversacional y no puede parecer como una forma natural de interactuar, pero cuando la entrada de voz se diseña y se integra bien puede hacer experiencias en las que el usuario se sienta a gusto y aumentar su confianza en la intención usuario-Kinect. (ABREGO, 2012)



Palabra clave / gatillo

El sensor sólo escucha una sola palabra clave o frase. Esto es la mejor manera de reducir los falsos activos. La palabra clave que se elija debe ser muy distinta de modo que no sea fácilmente malinterpretado. Por ejemplo, en Xbox360, "Xbox" es la palabra clave. No hay muchas palabras que suenen como "Xbox", por lo que es una buena palabra clave elegida. (ABREGO, 2012)

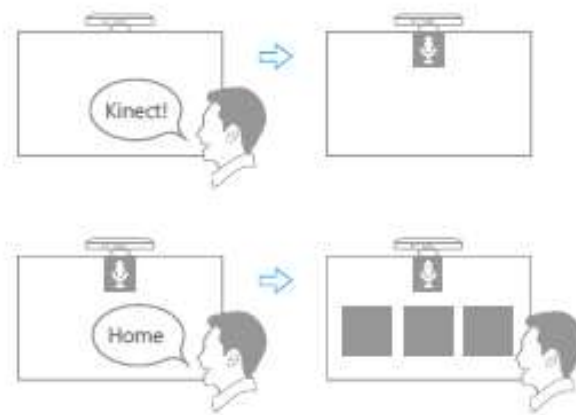
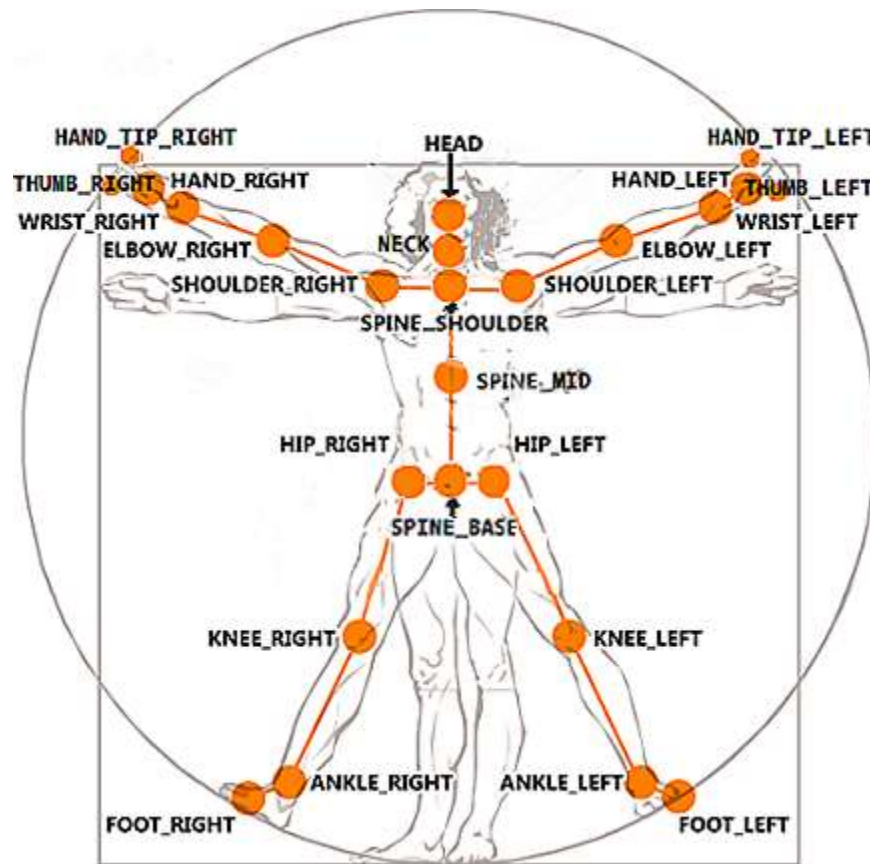


Figura 2. 6 Reconocimiento de voz Fuente: (ABREGO, 2012)

C. Esqueleto detectado por Kinect

Kinect tiene la capacidad de detectar automáticamente los esqueletos, con solo quedándose con **todo el cuerpo** en frente del sensor. (Kerkhove, 2014)



Cada esqueleto tiene 25 articulaciones:

Figura 2.7 Las articulaciones del esqueleto Fuente: (Kerkhove, 2014)

- **Seguimiento Mano** - Cada parte de un cuerpo tiene ahora una indicación de en qué estado se encuentra, *abierto, cerrado, Lazo* donde está apuntando con dos dedos
- **Actividades** - Indicación lo que la emoción facial del usuario, ojo izquierdo cerrado, la boca abierta, etc. (*Más podrían añadirse más adelante*)
- **Inclinándose** - Indicación de si el usuario está inclinando hacia la izquierda o hacia la derecha
- **Apariencia** - Le dice más sobre el usuario si él / ella lleva gafas (*Más podrían añadirse más adelante*)
- **Expresiones** - Expresión de la persona actual, Fe feliz o neutral (*Más podrían ser añadido más tarde*)

- **Comprometido** - Indicación de si el usuario está buscando en el sensor o no
- Clases Body**

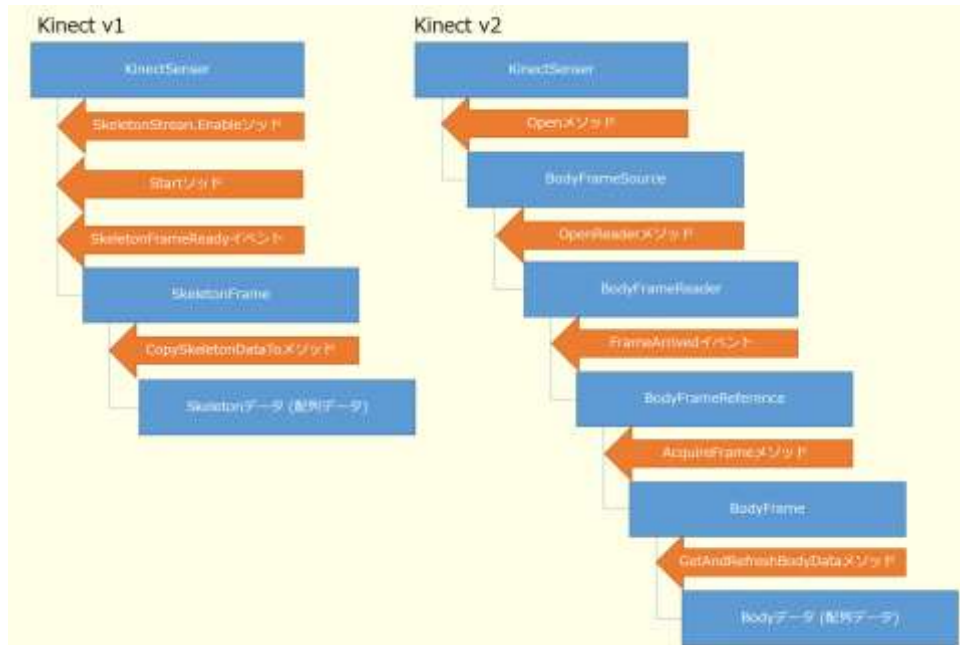


Figura 2. 8 Esqueleto procedimiento de adquisición de datos **Fuente:** (buildinsider.net, 2014)

Posición Del Esqueleto

La posición del esqueleto se representa en metros y el sistema de coordenadas que utiliza un **diestro**, es decir, el eje Y positivo es hacia arriba y el eje X positivo es a **la izquierda**.



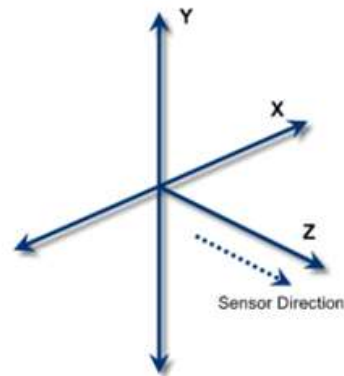


Figura 2. 9 Posición del esqueleto Fuente: (ABREGO, 2012)

D. Realidad Aumentada

La Realidad Aumentada, según Basogain (X. Basogain, 2013), es una tecnología que complementa la percepción e interacción con el mundo real y permite al usuario estar en un entorno real aumentado con información adicional generada por el computador. – un ejemplo de esto se muestra en la siguiente figura.



Figura 2. 10 Realidad Aumentada, Fuente: (Mocholi, 2014)



Para Barfield y Caudell (Barfield, 2011), la Realidad Aumentada es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, que se combinan con elementos virtuales para la creación de una Realidad Aumentada en tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente; ésta es la principal diferencia con la Realidad Virtual, puesto que no sustituye la realidad física, sino que sobreimprime los datos informáticos al mundo real.

Algunas características de la Realidad Aumentada son (Fabregat, 2012):

- a. Combina lo real y lo virtual: la información digital es combinada con la realidad.
- b. Funciona en tiempo real: la combinación de lo real y lo virtual se hace en tiempo real.
- c. Registra en tres dimensiones: en general la información aumentada se localiza o registra en el espacio. Para conservar la ilusión de ubicación real y virtual, ésta última tiende a conservar su ubicación o a moverse respecto a un punto de referencia en el mundo real.

E. Visión por Computadora



Figura 2. 11 Visión por Computadora, **Fuente: (Valdes, 2013)**

El potencial de esta tecnología está basado en la habilidad de aprovechar las características visuales del ambiente capturado naturalmente por la cámara. Una de sus ventajas es que no es necesario modificar el ambiente; para esto, se tienen en cuenta dos técnicas principales: Sistemas de marcadores (Marker Systems) basados en patrones de referencia impresos y Sistemas sin marcadores (Markerless Systems) basados en patrones encontrados en la imagen pura (como patrones faciales) los cuales requieren de la implementación de algoritmos mucho más complejos. Es importante tener en cuenta que se debe hacer un registro para el alineamiento del contenido virtual con respecto a los objetos del mundo real. (OLMEDO, 2012)

F. Probadores

Un **probador** (Parrado, 2012) es una habitación destinada a cambiarse de ropa en los establecimientos comerciales.

Los probadores se encuentran en tiendas de ropa en las que se estima conveniente probarse las prendas de vestir antes de comprarlas. De ahí, su nombre. Su objeto es el de preservar la intimidad de los clientes mientras se están desvistiendo. Así pues es común encontrar probadores en camiserías, grandes almacenes y tiendas de ropa en general. Pero no así en zapaterías, sombrererías y otros establecimientos en los que no es necesario desnudarse para realizar la prueba.

Los probadores son estancias individuales que constan de al menos un espejo de grandes dimensiones y varias perchas fijas para colgar la ropa. Pueden contar también con banquetas u otros pequeños muebles y se cierran mediante una puerta o una cortina corredera. Están situados al fondo del establecimiento en un punto apartado y discreto.



Figura 2. 12 Probadores, Fuente: (Estanteriasmoro, 2014)

G. Objetos Virtuales

Los objetos virtuales (José Luis Sierra, 2005) son generados por un ordenador y un software de modelación, como por ejemplo Maya o Blender. Estos programas vienen cargados con un gran número de formas en 3 dimensiones, los cuales son los bloques de construcción de objetos más complejos. Por ejemplo, se puede modelar un coche conectando cubos, cilindros, pirámides y esferas de diferentes formas y tamaños. Al ser objetos en 3D, están modelados en tres ejes (X, Y y Z), y pueden ser objetos y visionados desde cualquier ángulo. Cuando empiezas a modelar un objeto, no tiene ningún color de superficie o textura. Todo lo que ves en tu pantalla es el esqueleto del objeto – las líneas de los cubos, bloques y esferas individuales que han sido usados para construirlo. Esto se llama “*wireframe*”. Cada forma que está compuesta



por las líneas de un wireframe se llama “*polígono*”. Por ejemplo, una pirámide está hecha de cuatro polígonos en forma de triángulo.



Figura 2. 13 Objetos Virtuales, Fuente: (Luis, 2014)

H. Interfaz Natural de Usuario (NUI)

Las NUI's (Peralta, 2012) son aquellas en las que se interactúa con un sistema y/o aplicación, sin utilizar dispositivos de entrada, como ratón, teclado, lápiz óptico. En lugar de esto se utilizan las manos o las yemas de los dedos.

El término natural se refiere, en términos computacionales, a lo que le es más fácil o común hacer a una persona en su vida diaria, en la cual en todo momento existe retroalimentación inmediata y continua al momento de interactuar con algún objeto o persona. Un ejemplo puede verse claramente en un automóvil,

el conductor sabe que tiene que girar el volante para dar vuelta y al mismo tiempo debe acelerar para poder avanzar.

La NUI se enfoca en la construcción de una interfaz adaptada a nuestras capacidades, que permita concentrar nuestra atención en realizar una tarea, en lugar de como realizar una tarea. La interacción se alcanza de forma natural al mejorar la experiencia de usuario con una interfaz de tal forma que se vuelve invisible al usuario en un proceso de aprendizaje por repetición.

I. Sistemas E-commerce

El comercio electrónico, también conocido como e-commerce (electronic commerce en inglés), consiste en la compra y venta de productos o de servicios a través de medios electrónicos, tales como Internet y otras redes informáticas.

La mayor parte del comercio electrónico consiste en la compra y venta de productos o servicios entre personas y empresas

Un porcentaje considerable del comercio electrónico consiste en la adquisición de artículos virtuales (software y derivados en su mayoría), tales como el acceso a contenido "Premium" de un sitio web. (Leder Rivas Andrés, 2011)



Ventajas para las empresas

- ✓ Mejoras en la distribución
- ✓ Comunicaciones comerciales por vía electrónica
- ✓ Beneficios operacionales
- ✓ Facilidad para fidelizar clientes

Funcionalidades

- ✓ Catálogo
- ✓ Clientes
- ✓ Pedidos
- ✓ Pago
- ✓ Transporte
- ✓ Estadísticas
- ✓ Traducciones
- ✓ Ubicación
- ✓ Referenciamiento
- ✓ Seguridad
- ✓ Gestión
- ✓ Front-Office modulable

Tipos De E-Commerce

Business to Consumer B2C.

El B2C (Business to Consumer) consiste en la venta de productos y/o servicios de la empresa a consumidores finales (no empresariales). Mediante este tipo de e-

commerce se mejora el servicio brindado a los clientes y se genera oportunidades para incrementar las ventas e ingresos. (Leder Rivas Andrés, 2011)

La mayoría de los usuarios que compran en internet buscan artículos a precios competitivos. A medida que aumenta el número de empresas que ofrecen productos en línea, los clientes disponen de mayor libertad para investigar y comparar precios. Con una solución B2C, la empresa ofrece a sus clientes la comodidad de efectuar compras en Internet sin tener que acudir al establecimiento, además se puede atraer nueva clientela. (Leder Rivas Andrés, 2011)

Business to Business B2B.

El B2B consiste en la automatización en las relaciones comerciales y cadenas de suministros que se da entre dos empresas mediante el uso de redes de telecomunicaciones. Este tipo de e-commerce brinda la oportunidad de reducir los costes e incrementar los ingresos. Mediante el acceso a la web y con los: (García, 2013)

Medios de pago

Unos de los elementos fundamentales en el comercio en general y en el comercio electrónico en particular, es la realización del pago. En este contexto de los pagos, en

general hay 3 puntos importantes para realizar una transacción: (García, 2013)

1. El comprador necesita la garantía sobre calidad, cantidad y características de los bienes.
2. El vendedor necesita garantía del pago.
3. La transacción debe tener un alto nivel de confidencialidad

Tarjeta de débito y de crédito: Son el medio más popular y tradicionalmente usado en el comercio electrónico. El comprador adeuda su tarjeta con el pago al momento de realizar la transacción (débito) o más tarde con o sin devengo de intereses (crédito). El vendedor recibe un cobro rápido, a cambio de una comisión que le descuenta el banco. (García, 2013)

PayPal

PayPal es una forma más segura y fácil de pagar y recibir pagos en el ámbito del comercio electrónico. El servicio permite a cualquier persona pagar en la forma que ésta prefiera, incluyendo tarjetas de crédito, cuentas bancarias o saldos de cuentas, sin compartir información financiera.

PayPal se ha convertido rápidamente en un líder global en soluciones de pago en línea con más de 153 millones de cuentas en todo el mundo.

Disponible en 190 mercados y 24 divisas en todo el mundo, PayPal permite el comercio electrónico mundial mediante el pago posible a través de diferentes. (García, 2013)

E-commerce en el Perú

En el Perú también se alinea con las tendencias globales y empieza a impulsar el comercio electrónico o **e-commerce en Perú**. De hecho, según cifras brindadas por Asbanc (2015), Asociación de Bancos del Perú, en los últimos dos años, las ventas a través de plataformas web prácticamente se duplicaron. Pasaron de S/. 2,102 millones a S/. 5,875 millones. Si nos referimos al número de operaciones, se incrementaron de 2.2 millones a 4.1 millones.

Es necesario tener en cuenta que estas transacciones contabilizadas por la asociación, fueron solo las que emplearon tarjetas de crédito y no las que fueron con tarjeta de débito o efectivo. (webtilia, 2015)

2.4. Entorno de Desarrollo Integrado (IDE)

Para poder desarrollar una aplicación es necesario contar con Entorno de Desarrollo Integrado, este es un programa el cual cuenta con un editor de código, compilador, depurador y un constructor de interfaz gráfica; estos pueden estar orientados a un lenguaje de programación o puede ser multilenguaje. En la **Tabla 2.2** se muestra los IDEs recomendados.

Tabla 2. 2: Entornos de Desarrollo, Fuente: Elaboración Propia

Sistema Operativo	Entorno de Desarrollo
Windows OS, LINUX	Eclipse IDE, Netbeans, Visual Code
OS, LINUX	Processing, Netbeans, Visual Code
Windows OS	Microsoft Visual Studio, Visual Code

Frameworks y librerías para Realidad Aumentada

Para el desarrollo de una aplicación con realidad aumentada y tecnología Kinect además del IDE, es necesario un SDK, que no es más que una interfaz de programación de aplicaciones; la cual permite el uso de algún lenguaje de programación, dependiendo del sistema operativo al cual este destinada la aplicación. Así mismo, para desarrollar una aplicación en Realidad Aumentada con Kinect se necesita un conjunto de herramientas ya sean librerías o SDK orientado al desarrollo de una aplicación con Realidad Aumentada, a continuación, veremos algunas librerías y SDK utilizados para el desarrollo de aplicaciones con Realidad Aumentada y Kinect.

a) Airkinect (Wouter, 2012)

AIRKinect Extension es una extensión nativa para utilizar con Adobe AIR 3.0 + en las ventanas y OSX. AIRKinect AIR permite a los desarrolladores aprovechar la rica experiencia del Microsoft Kinect y empujan interactividad a un nuevo



nivel. La extensión AIRKinect proporciona a los desarrolladores de Flash la misma información y la configuración como con el SDK de Kinect proporcionada por Microsoft.

Las versiones futuras incluyen:

- a. AIRKinect Componentes de interfaz de usuario - especialmente para su uso con el Kinect y NUI
 - b. Utilidades de gestión de Extensión
 - c. Gestos
 - d. Los comandos de voz
 - e. Muchos más
- b) **Kinect Sdk** (Research.microsoft, 2011)

El Kinect para Windows SDK beta incluye controladores, API rica para el acceso a los flujos de datos en bruto del sensor de profundidad, acceso al sensor de la cámara de color, seguimiento del esqueleto, capacidades de audio avanzada, los documentos de la instalación, y recursos materiales. Proporciona capacidades de Kinect a los desarrolladores que crean aplicaciones con C++, C#, Visual Basic o mediante el uso de Microsoft Visual Studio 2010.

c) Jnect (Duncan, 2012)

El SDK de Kinect requiere que el programador escribe la aplicación en C ++ / C # para tener acceso al SDK, y hasta la fecha, no ha habido soporte para Java. El único Puente es Jnect que conecta la brecha y crea la interoperabilidad entre el SDK de Microsoft y el mundo Java. El proyecto ofrece un plugin de Eclipse que utiliza JNI para conectar con el Microsoft Kinect SDK. El plug-in ofrece una API de Java para utilizar las funciones seleccionadas que ofrece el SDK de Microsoft original.

d) Opencv (Gracia, 2013) es una librería de computación visual creada por Intel, esta librería está disponible para múltiples plataformas como Windows, Linux, Mac, Android, además cuenta con soporte para diferentes lenguajes como Python, Java, C/C++, entre otros.

Opencv puede ser usado bajo licencia BSD para proyectos escolares o comerciales, opencv puede ser usado en la robótica, análisis de imágenes o vídeo, seguimiento de objetos, detección y reconocimiento de rostros, reconocimiento de placas de vehículos y más.

e) Freenect (Landerpfc, 2012)

Desarrolla un driver para la cámara Kinect, coordinando a un gran grupo de personas que trabajan con este nuevo

periférico. El driver se llama Freenect, y hasta el momento es capaz de capturar la imagen de la cámara con una resolución VGA (640x480), así como los datos de profundidad, y controlar el pequeño motor en su base. Sin embargo, con kinectComp (interfaz diseñada para la cámara Kinect) sólo podemos acceder a los datos de profundidad e imagen. Una versión experimental de Freenect consigue capturar el audio de sus micrófonos y cambiar el estado del LED frontal. Y trabajan en un sistema llamado Fakenect con el cual puedas grabar una sesión con una kinect, para trabajar posteriormente con esa sesión sin necesidad de tenerla conectada al PC.

f) OpenNI (Landerpfc, 2012)

OpenNI (Interacción Natural Abierta) es multi-idioma, multi-plataforma, define un marco para las API donde poder escribir aplicaciones que utilizan Interacción Natural. El API de OpenNI se compone de un conjunto de interfaces para escribir aplicaciones de Interacción Natural. El objetivo principal de OpenNI es formar una API estándar que permite la comunicación con ambos:

- ❖ Sensores de visión y audio (los dispositivos que "ven" y "escuchan" las figuras y sus alrededores.)



- ❖ Middleware (los componentes de software que analizan los datos de audio y vídeo grabados desde el escenario y lo interpretan) para la percepción de la visión y del audio. Por ejemplo, un software que recibe información visual, como una imagen, devuelve la ubicación de la palma de una mano dentro de la imagen.

2.5. Tecnologías de programación

Sistema E-commerce

Html

HTML (Álvarez, 2001) es el lenguaje con el que se definen las páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la que se presenta el texto y otros elementos de la página.

Javascript / Jscript

JavaScript (Valdés, 2007) es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales

sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo.

Vbscript

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de Javascript.

Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows. Tanto su sintaxis como la manera de trabajar están muy inspiradas en él. (Microsoft., 2007)

Applets Java

Es una manera de incluir programas complejos en el ámbito de una página Web. Estos applets se programan en Java y por tanto se benefician de la potencia de este lenguaje para la Red. La principal ventaja de utilizar applets consiste en que son mucho menos dependientes del navegador que los scripts en Javascript, incluso independientes del sistema operativo del ordenador donde se ejecutan. (Vaquero, 2014)

Componentes Activex

ActiveX (Microsoft, A, 2007) es una tecnología de Microsoft para el desarrollo de páginas dinámicas. Tiene presencia en la programación del lado del servidor y del lado del cliente, aunque existan diferencias en el uso en cada uno de esos dos

casos. En el cliente: Son pequeños programas que se pueden incluir dentro de páginas Web y sirven para realizar acciones de diversa índole.

En el servidor: También existen controles ActiveX del servidor. Por ejemplo, cuando realizamos una conexión con una base de datos, estamos utilizando un control ActiveX del servidor.

Bootstrap

Es framework de Twitter que permite crear interfaces web con CSS y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive Design. (Solis, 2014)

Ajax

Ajax son las siglas de Asynchronous JavaScript And XML y es una tecnología que permite a una página web actualizarse de forma dinámica sin que tenga que recargarse completamente. JavaScript es el encargado de comunicarse con el servidor enviando y recibiendo datos desde la página web, en el servidor la solicitud es



procesada y se envía una respuesta que es interpretada de nuevo por JavaScript en la página web.

JQuery

JQuery (Mikoluk, 2013) es una librería JavaScript open-source, que funciona en múltiples navegadores, y que es compatible con CSS3. Su objetivo principal es hacer la programación “scripting” mucho más fácil y rápida del lado del cliente. Con jQuery se pueden producir páginas dinámicas, así como animaciones parecidas a Flash en relativamente corto tiempo, también permite agregar interacción con la tecnología AJAX

2.6. Lenguajes de programación

Sistema E-commerce

JAVA SERVER PAGE (JSP)

JSP (Bautista, 2011) es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java ServePages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor. JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET desarrollado para la creación de aplicaciones web potentes.

ASP NET

Es el lenguaje de scripting del lado del servidor creado por Microsoft ASP (Active Server Pages) es la tecnología desarrollada para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página ASP NET. (Julius, 2008)

PHP

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor. (TILATAM, 2010)



Probador Virtual

C#

C# (Binstock, 2014) (pronunciado si sharp en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por la notación musical, donde # (sostenido, en inglés sharp) indica que la nota (C es la nota do en inglés) es un semitono más alto, sugiriendo que C# es superior a C/C++. Además, el signo '#' viene de cuatro '+' pegados.¹

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee

el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows, Unix y GNU/Linux.

C++

C++ (villalobos, 2014) es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma.

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Existen también algunos intérpretes, tales como ROOT.

Una particularidad del C++ es la posibilidad de redefinir los operadores (sobrecarga de operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales.

El nombre C++ fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". En C++, la expresión "C++" significa "incremento de C" y se refiere a que C++ es una extensión de C.

Autodesk 3ds Max

Autodesk 3ds Max (Gonzalo, 2010) (anteriormente **3D Studio Max**) es un programa de creación de gráficos y animación 3D desarrollado por Autodesk, en concreto la división Autodesk Media & Entertainment (anteriormente Discreet). Creado inicialmente por el Grupo Yost para Autodesk, salió a la venta por primera vez en 1990 para DOS.

3ds Max, con su arquitectura basada en plugins, es uno de los programas de animación 3D más utilizado, especialmente para la creación de videojuegos, anuncios de televisión, en arquitectura o en películas.

Blender 3D

Blender (2011) es un programa gratuito para el **modelado, animación y renderizado de gráficos 3D**. Se trata de un proyecto de código abierto con una potencia comparable a la de los paquetes comerciales más destacados, como Maya o 3D Studio.

Las posibilidades de Blender son inmensas: sus herramientas permiten crear objetos, esculpirlos, iluminarlos, pintarlos con texturas y animarlos

en escenas complejas. **La modalidad Game Blender**, incluida en el paquete principal, es un editor de videojuegos 3D avanzado.

Base de Datos

SQL Server

Es un sistema para la gestión y análisis de bases de datos creado por Microsoft, el mismo se basa en el modelo relacional. Sel Server utiliza como lenguajes de consulta T-SQL y ANSI SQL. (Minayo, 2011)

PostgreSQL

PostgreSQL (Martinez R. , 2010) es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales.

Servidores web

Apache

Apache (Geneura, 2008) es un poderoso servidor web, cuyo nombre proviene de la frase inglesa “a patchy server” y es completamente libre, ya que es un software Open Source y con licencia GPL. Una de las ventajas más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede

trabajar con diferentes sistemas operativos y mantener su excelente rendimiento.

Glassfish

Glassfish (Herrera, 2013) es un servidor de aplicaciones que implementa la plataforma JavaEE5, por lo que soporta las últimas versiones de tecnologías como: JSP, JSF, Servlets, EJBs, Java API para Servicios Web (JAX-WS), Arquitectura Java para Enlaces XML (JAXB), Metadatos de Servicios Web para la Plataforma Java 1.0, y muchas otras tecnologías.

2.7. Metodología de Desarrollo de Software

Proceso Unificado de Desarrollo de Software RUP

Según **Stephen R. Schach** (2009); El Proceso Unificado es más que un simple proceso de desarrollo de software, es un marco de trabajo genérico que puede ser especializado para el desarrollo de sistemas grandes de diferentes áreas, diferentes tipos de organización, diferentes niveles de competencia y diferentes tamaños de proyectos.

Es un proceso basado de componentes, lo cual significa que el software en desarrollo está formado por componentes interconectados por interfaces bien definidas.

Este proceso usa UML cuando se prepara los diagramas de modelamiento del sistema de software. De hecho, el UML es una parte integral del proceso unificado.

Características del Proceso Unificado:

- Es un proceso iterativo, basado en el refinamiento sucesivo del sistema.
- Es un proceso controlado, donde juegan un papel de primordial importancia la gestión de requisitos y el control de cambios
- Basado en la construcción de modelos visuales del sistema.
- Centrado en el desarrollo de la arquitectura por lo que maneja el concepto de desarrollo basado de componentes.
- Conducidos por casos de uso.
- Soporta técnicas orientadas a objetos y en particular uso de UML.
- Formatea el control de calidad.
- Soportado por herramientas.

Programación Extrema (eXtreme Programing) XP.

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. (Valencia, 2006)

El enfoque al desarrollo de software que propone XP, pretende ser:

- Útil en proyectos de alto riesgo caracterizados por una constante variación de los requerimientos.
- Promueve la integración del cliente en el proceso de desarrollo, la programación de a pares y el trabajo en equipo como medios para lograrlo.
- XP se basa en realimentación continua entre el cliente y el equipo de desarrollo.
- Comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.
- XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Modelo en Espiral

El desarrollo en espiral es un modelo de ciclo de vida del software definido por primera vez por **Barry Boehm** en 1988, utilizado generalmente en la Ingeniería de software. Las **actividades** de este modelo se conforman en una **espiral**, en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a priori, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior. (Aspiazu, 2010)



Los principios básicos del modelo espiral son los siguientes:

- ❖ **La atención se centra en la evaluación y reducción del riesgo del proyecto** dividiendo el proyecto en segmentos más pequeños y proporcionar más facilidad de cambio durante el proceso de desarrollo, así como ofrecer la oportunidad de evaluar los riesgos y con un peso de la consideración de la continuación del proyecto durante todo el ciclo de vida.
- ❖ **Cada viaje alrededor de la espiral atraviesa cuatro cuadrantes básicos:** (1) Determinar objetivos, alternativas, y desencadenantes de la iteración; (2) Evaluar alternativas; Identificar y resolver los riesgos; (3) Desarrollar y verificar los resultados de la iteración, y (4) Plan de la próxima iteración.
- ❖ Cada ciclo comienza con la **identificación de los interesados** y sus **condiciones de ganancia**, y termina con la revisión y examinación.

2.8. Cuadro comparativo de Metodología

Tabla 2. 3: Comparación de Metodologías: Fuente: Elaboración Propia

CRITERIOS	METODOLOGIA		
	ESPIRAL	RUP	XP
Tamaño del proyecto	Enfocado a cualquier tipo de proyecto	Enfocado a cualquier tipo de proyecto	Enfocado a proyectos cortos
Obtención de requisitos	Se basa en los Usecase	Se basa en los Usecase	Se basa en los Usecase
Relación con el cliente	Mientras se avanza las fases del proyecto se podrá realimentar la fase o proceso anterior, y con esto se conseguirá tener un mayor balance entre el producto final entregado y el producto final esperado	Al final de cada fase se le presenta al cliente los artefactos finales de dicha fase, para que sean evaluados por este y se pueda generar las iteraciones necesarias para la siguiente fase.	La comunicación con el cliente es fluida (a través de su representante) después de cada iteración el cliente recibe una pieza de programa funcional
Desarrollo	Aquí los tres procesos están basados en iteraciones, lo que les permite acercarse poco a poco a las soluciones sin tener que entrar demasiado rápido a los detalles, la diferencia están en que los programadores de XP tiene menor carga a parte del desarrollo del software entonces les permite hacer las iteraciones con una menor duración		

Después de haber realizado una evaluación de dichas metodologías ya mencionadas se llegó a optar por la metodología **RUP**, la cual se ajusta a los diferentes requisitos necesarios para desarrollar el sistema web de ventas y aplicación de Realidad Aumentada Basada en Tecnología Kinect para la Interacción en el proceso de prueba y compra de prendas de vestir



2.9. Metodología de desarrollo de software escogida

Metodología RUP (Rational Unified Process)

El Proceso Unificado de Rational (Ivar, Grady, & James, 1999). es permitir la producción de un software de la mayor calidad que satisfaga las necesidades de los usuarios finales, dentro de planificaciones y presupuesto predecibles. El Proceso Unificado de Rational captura algunas de las mejores prácticas de desarrollo de software, de una forma que es adaptable a un amplio rango de proyectos y organizaciones. En el aspecto de la gestión, El Proceso Unificado de Rational proporciona un enfoque disciplinado sobre como asignar tareas y responsabilidades dentro de una organización de desarrollo de software.

Ciclo de vida:

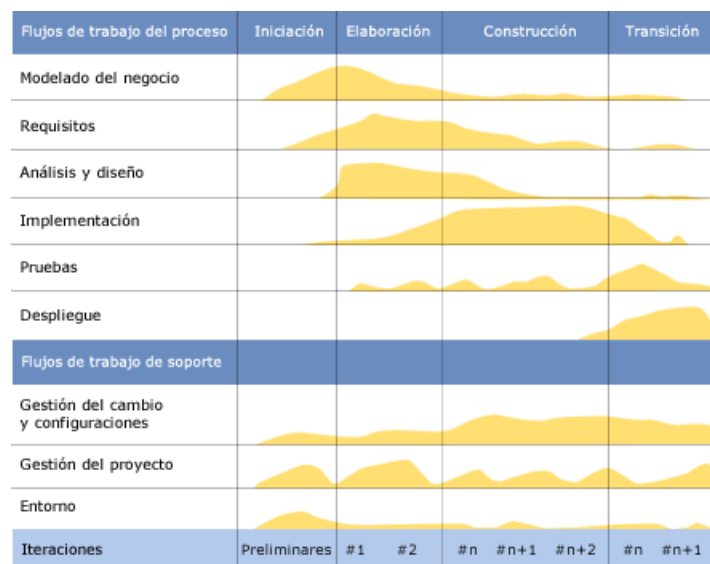


Figura 2. 14 Metodología RUP Fuente: (Rosales, 2010)



En el ciclo de vida RUP veremos una implementación del desarrollo en espiral. Con el ciclo de vida se establecen tareas en fases e iteraciones. El RUP maneja el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una base de inicio

Descripción de las Fases

➤ **Fase de Inicio**

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar que recursos deben ser asignados al proyecto.

Los objetivos en esta fase son:

- ✓ Establecer el ámbito del proyecto y sus límites.
- ✓ Encontrar los Casos de Uso críticos del sistema, los escenarios básicos que definen la funcionalidad.
- ✓ Mostrar al menos una arquitectura candidata para los escenarios principales.
- ✓ Estimar el coste en recursos y tiempo de todo el proyecto.



- ✓ Estimar los riesgos, las fuentes de incertidumbre.

Los resultados de la fase de inicio deben ser:

- ✓ Un documento de visión: Una visión general de los
- ✓ requerimientos del proyecto, características clave y restricciones
- ✓ principales.
- ✓ Modelo inicial de Casos de Uso (10-20% completado).
- ✓ El caso de negocio.
- ✓ Modelo de negocio, si es necesario
- ✓ Prototipos exploratorios para probar conceptos o la arquitectura
- ✓ candidata.

Al terminar la fase de inicio se deben comprobar los criterios de evaluación para continuar:

- ✓ Todos los interesados en el proyecto coinciden en la definición del
- ✓ ámbito del sistema y las estimaciones de agenda.
- ✓ Entendimiento de los requisitos, como evidencia de la fidelidad de los
- ✓ Casos de Uso principales.
- ✓ Las estimaciones de tiempo, coste y riesgo son creíbles.
- ✓ Comprensión total de cualquier prototipo de la arquitectura desarrollado.

- ✓ Los gastos hasta el momento se asemejan a los planeados.
- ✓ Si el proyecto no pasa estos criterios hay que plantearse abandonarlo o repensarlo profundamente.

Fase de Elaboración

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final.

Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

Los objetivos de esta fase son:

- ✓ Definir, validar y cimentar la arquitectura.
- ✓ Completar la visión.
- ✓ Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costes si procede.
- ✓ Demostrar que la arquitectura propuesta soportará la visión con un coste razonable y en un tiempo razonable.

Al terminar deben obtenerse los siguientes resultados:

- ✓ Un modelo de Casos de Uso completa al menos hasta el 80%: todos los casos y actores identificados, la mayoría de los casos desarrollados.
- ✓ Requisitos adicionales que capturan los requisitos no funcionales y cualquier requisito no asociado con un Caso de Uso específico.

En esta fase se debe tratar de abarcar todo el proyecto con la profundidad mínima. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes.

En la fase de elaboración se actualizan todos los productos de la fase de inicio.

Los criterios de evaluación de esta fase son los siguientes:

- ✓ La visión del producto es estable.
- ✓ La arquitectura es estable.
- ✓ Se ha demostrado mediante la ejecución del prototipo que los principales elementos de riesgo han sido abordados y resueltos.
- ✓ El plan para la fase de construcción es detallado y preciso. Las estimaciones son creíbles.
- ✓ Todos los interesados coinciden en que la visión actual será alcanzada si se siguen los planes actuales en el contexto de la arquitectura actual.

- ✓ Los gastos hasta ahora son aceptables, comparados con los previstos.

Si no se superan los criterios de evaluación quizá sea necesario abandonar el proyecto o replanteárselo considerablemente.

➤ **Fase de Construcción**

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones.

Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

Los objetivos concretos según incluyen:

- ✓ Minimizar los costes de desarrollo mediante la optimización de recursos y evitando el tener que rehacer un trabajo o incluso desecharlo.
- ✓ Conseguir una calidad adecuada tan rápido como sea práctico.
- ✓ Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea práctico.

Los resultados de la fase de construcción deben ser:

- ✓ Modelos Completos (Casos de Uso, Análisis, Diseño, Despliegue e Implementación)
- ✓ Arquitectura íntegra (mantenida y mínimamente actualizada)

✓ Riesgos Presentados Mitigados

✓ Caso del Negocio Actualizado

Los criterios de evaluación de esta fase son los siguientes:

✓ El producto es estable y maduro como para ser entregado a la comunidad de usuario para ser probado.

✓ Todos los usuarios expertos están listos para la transición en la comunidad de usuarios.

✓ Son aceptables los gastos actuales versus los gastos planeados.

➤ **Fase de Transición**

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

Los principales objetivos de esta fase son:

✓ Conseguir que el usuario se valga por sí mismo.

✓ Un producto final que cumpla los requisitos esperados, que funcione y satisfaga suficientemente al usuario.

✓ Los resultados de la fase de transición son:

✓ Prototipo Operacional

✓ Documentos Legales

- ✓ Caso del Negocio Completo
- ✓ Línea de Base del Producto completa y corregida que incluye todos los modelos del sistema
- ✓ Descripción de la Arquitectura completa y corregida
- ✓ Las iteraciones de esta fase irán dirigidas normalmente a conseguir una nueva versión.

Los criterios de evaluación de esta fase son los siguientes:

- ✓ El usuario se encuentra satisfecho.
- ✓ Son aceptables los gastos actuales versus los gastos planificados.

2.10. Normalidad

Con respecto a la normatividad esta tesis se basará en el uso de la Norma ISO/IEC 9126, que es un estándar ISO de calidad en el desarrollo de software. (Karina Melendez, 2009)

Definición.

Es un estándar internacional para la evaluación del Software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software.

Define un marco conceptual de calidad que considera los siguientes factores: Calidad del Proceso, Calidad del Producto de Software (Calidad Interna y Calidad Externa) y Calidad en Uso.

Según el marco conceptual, la calidad de un proceso contribuye a mejorar la calidad del producto, y a su vez, la calidad del producto contribuye a mejorar la calidad en uso.

La sección ISO/IEC 9126-1 describe un modelo de dos partes para la calidad de productos de software. (Karina Melendez, 2009)

Calidad interna Y Calidad Externa

- La primera parte del modelo especifica seis características para la calidad interna y externa, las cuales son a su vez subdivididas en sub características.
- Estas sub características se manifiestan externamente cuando el software es utilizado como parte de un sistema de cómputo, y son el resultado de los atributos internos del software

Calidad en el uso

- La segunda parte del modelo especifica cuatro características de calidad en el uso
- El concepto de calidad en el uso se puede definir como la extensión a la cual un producto utilizado por usuarios específicos cumple la necesidad de alcanzar metas específicas con efectividad, productividad y satisfacción en un contexto de uso definido.

ESQUEMA DE CALIDAD INTERNA Y EXTERNA

Figura 2. 15 *Calidad Interna y Externa de la ISO9126 Fuente: (Karina Melendez, 2009)*



ESQUEMA DE CALIDAD EN USO

Figura 2. 16 *Calidad Interna y Externa de la ISO9126 Fuente: (Karina Melendez, 2009)*



Características de Calidad Interna y Externa

El modelo de calidad para la calidad interna y externa ha sido establecido en categorías de atributos de calidad del software en seis características (funcionalidad, confiabilidad, usabilidad, eficiencia, capacidad de mantenimiento y portabilidad), que se subdividen a su vez en sub características.

Las sub características se pueden medir mediante métricas internas o externas. (Karina Melendez, 2009)

1 Funcionalidad.

Es la capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas

Esta característica se refiere a lo que hace el software para satisfacer necesidades, mientras que las otras características se refieren principalmente a cuándo y a cómo satisfacen las necesidades

Para un sistema que es operado por un usuario, la combinación de la funcionalidad, fiabilidad, usabilidad y eficiencia puede ser medida externamente por su calidad en uso

1.1 Adecuación

Es la capacidad del producto de software para proveer un adecuado conjunto de funciones para las tareas y objetivos especificados por el usuario

1.2 Exactitud

Es la capacidad del producto de software para proporcionar los resultados o efectos acordados con un grado necesario de precisión.

1.3 Interoperabilidad

Es la capacidad del producto de software de interactuar con uno o más sistemas especificados. La interoperabilidad se utiliza en lugar de compatibilidad para evitar una posible ambigüedad con la reemplazabilidad.

1.4 Seguridad

Es la capacidad del producto de software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos, y a las personas o sistemas autorizados no se les niegue el acceso a ellos

La seguridad en un sentido amplio se define como característica de la calidad en uso, pues no se relaciona con el software solamente, sino con todo un sistema

1.5 Conformidad de la Funcionalidad

La capacidad del producto de software de apegarse a los estándares, convenciones o regulaciones legales y prescripciones similares referentes a la funcionalidad

2 Confiabilidad

La capacidad del producto de software para mantener un nivel específico de funcionamiento cuando se está utilizando bajo condiciones específicas

El desgaste o envejecimiento no ocurre en el software. Las limitaciones en confiabilidad son debido a fallas en los requerimientos, diseño, e implementación

Las fallas debido a estos errores dependen de la manera en que se utiliza el producto de software y de las opciones del programa seleccionadas, más que del tiempo transcurrido

2.1 Madurez

Es la capacidad del producto de software para evitar fallas como resultado de errores en el software.

2.2 Tolerancia a errores

Es la capacidad del producto de software para mantener un nivel especificado de funcionamiento en caso de errores del software o de incumplimiento de su interfaz especificada.

2.3 Recuperabilidad

Es la capacidad del producto de software para restablecer un nivel especificado de desempeño y recuperar los datos afectados directamente en el caso de una falla Después de una falla, un producto de software a veces estará no disponible por

cierto período del tiempo, intervalo en el cual se evaluará su recuperabilidad.

2.4 Conformidad de la Confiabilidad

Es la capacidad del producto de software para apegarse a las normas, convenciones o regulaciones relativas a la confiabilidad

3 Usabilidad

Es la capacidad del producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo las condiciones especificadas.

Algunos aspectos de funcionalidad, confiabilidad y eficiencia también afectarán la usabilidad, pero para los propósitos de la ISO/IEC 9126 estos no son clasificados como usabilidad

3.1 Entendimiento.

Es la capacidad del producto de software para permitir al usuario entender si el software es adecuado, y cómo puede ser utilizado para las tareas y las condiciones particulares de la aplicación.

3.2 Aprendizaje.

Es la capacidad del producto de software para permitir al usuario aprender sobre su aplicación.

3.3 Operabilidad.

La capacidad del producto de software para permitir al usuario operarlo y controlarlo.



3.4 Atracción.

La capacidad del producto de software de ser atractivo al usuario. Esto se refiere a las cualidades del software para hacer el software más atractivo al usuario, tal como el uso del color y la naturaleza del diseño gráfico.

3.5 Conformidad de Usabilidad

La capacidad del producto de software para apegarse a los estándares, convenciones, guías de estilo o regulaciones relacionadas a su usabilidad.

4 Eficiencia.

Es la capacidad del producto de software para proveer un desempeño adecuado, de acuerdo a la cantidad de recursos utilizados y bajo las condiciones planteadas

Los recursos pueden incluir otros productos de software, la configuración de hardware y software del sistema, y materiales.

4.1 Comportamiento de Tiempos

Es la capacidad del producto de software para proveer tiempos adecuados de respuesta y procesamiento, y niveles de rendimiento cuando realiza su función bajo las condiciones establecidas

4.2 Utilización de Recursos

Es la capacidad del producto de software para utilizar cantidades y tipos adecuados de recursos cuando éste funciona bajo las condiciones establecidas

Los recursos humanos están incluidos dentro del concepto de productividad.

4.3 Conformidad de Eficiencia

La capacidad del producto de software para adherirse a estándares o convenciones relacionados a la eficiencia.

5 Capacidad de Mantenimiento

Es la capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y especificaciones de requerimientos funcionales.

5.1 Capacidad de ser Analizad.

Es la capacidad del producto de software para apegarse a diagnósticos de deficiencias o causas de fallas en el software o la identificación de las partes a ser modificadas.

5.2 Cambiabilidad.

Es la capacidad del software para permitir que una determinada modificación sea implementada. Implementación incluye codificación, diseño y documentación de cambios.

Si el software va a ser modificado por el usuario final, la cambiabilidad podría afectar la Operabilidad.

5.3 Estabilidad.

Es la capacidad del producto de software para evitar efectos inesperados debido a modificaciones del software.

5.4 Facilidad de Prueba

Es la capacidad del software para permitir que las modificaciones sean validadas.

5.5 Conformidad de Facilidad de Mantenimiento

Es la capacidad del software para apegarse a estándares o convenciones relativas a la facilidad de mantenimiento.

6 Portabilidad

Es la capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software.

6.1 Adaptabilidad.

Es la capacidad del producto de software para ser adaptado a diferentes entornos especificados sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerado. La adaptabilidad incluye la escalabilidad de capacidad interna (Ejemplo: campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

6.2 Facilidad de Instalación.

La capacidad del producto de software para ser instalado en un ambiente especificado.

6.3 Coexistencia.

La capacidad del producto de software para coexistir con otros productos de software independientes dentro de un mismo entorno, compartiendo recursos comunes.

6.4 Reemplazabilidad

La capacidad del producto de software para ser utilizado en lugar de otro producto de software, para el mismo propósito y en el mismo entorno.

6.5 Conformidad de Portabilidad

La capacidad del software para apegarse a estándares o convenciones relacionados a la portabilidad.

Características de Calidad en Uso.

Los atributos de la calidad en uso están categorizados en cuatro características: Eficacia, Productividad, Seguridad y Satisfacción. La calidad en uso es la visión de calidad del usuario. Alcanzar la calidad en uso depende de alcanzar la calidad externa necesaria que a su vez depende de alcanzar la calidad interna necesaria.

1. Eficacia

Es la capacidad del producto de software para permitir a los usuarios lograr las metas especificadas con exactitud e integridad, en un contexto especificado de uso

2. Productividad

Es la capacidad del producto de software para permitir a los usuarios emplear cantidades apropiadas de recursos, en relación a la eficacia lograda en un contexto especificado de uso
Los recursos relevantes pueden incluir: tiempo para completar la tarea, esfuerzo del usuario, materiales o costo financiero.

3. Seguridad

Es la capacidad del producto de software para lograr niveles aceptables de riesgo de daño a las personas, institución, software, propiedad (licencias, contratos de uso de software) o entorno, en un contexto especificado de uso

Los riesgos son normalmente el resultado de deficiencias en la funcionalidad (incluyendo seguridad), confiabilidad, usabilidad o facilidad de mantenimiento

4. Satisfacción

Es la capacidad del producto de software para satisfacer a los usuarios en un contexto especificado de uso

La satisfacción es la respuesta del usuario a la interacción con el producto, e incluye las actitudes hacia el uso del producto.



2.11. Evaluación Económica

Tasa Interna de Retorno

Es la tasa de rendimiento que hace el valor presente neto igual a 0.

$$VPN = -I_0 + \sum_{t=1}^{t=n} \frac{FT_t}{(1+r)^t} = 0 \quad \dots \dots \dots \quad I_0 = \sum_{t=1}^{t=n} \frac{FE_t}{(1+r)^t}$$

Dónde: r= TIR

El criterio de aceptación utilizando este método es aceptar aquellas inversiones independientes cuya TIR sea igual o mayor al costo de los recursos asignados a la inversión.

Obtención del TIR:

La TIR no puede obtenerse de manera directa, se acostumbra determinar el valor por medio de la construcción de la gráfica.

Restricciones en el cálculo de la TIR:

El cálculo de una TIR equivale a obtener la raíz de la ecuación de grado n, es posible que, para una misma inversión.

- No existe TIR.
- Solo hay una TIR positiva en el dominio de los números reales o
- Que se presente más de TIR positiva.



Interpretación de la TIR:

La TIR tiene diversas interpretaciones, entre las que destacan:

- Es la rentabilidad que se obtendrá por la inversión.
- Es la tasa de crecimiento de una inversión.
- La TIR es la máxima tasa de interés que se pagara a un banco.
- Cuando se conoce el pago de un crédito por el total de interés y capital, la TIR del pago totales una tasa de interés.

Valor Actual Neto

Es la cantidad que un inversionista podría pagar por una inversión en exceso de su costo.

También se conoce como valor presente neto VPN.

$$VPN = -I_0 + \sum_{t=1}^{t=n} \frac{FE_t}{(1+r)^t}$$

I_0 = inversión inicial.

FE_t = flujo de efectivo en el periodo t

Σ = suma de los flujos descontados

r = tasa de descuento

t = índice de tiempo

$1/(1+r)$ = factor de valor presente

2.12. Gestión de Riesgo de Seguridad de Información (ISO 27005)

El ISO/IEC 27005:2008 (Cruz, 2014) provee lineamientos para la gestión del riesgo para la seguridad de información. Apoya los conceptos generales especificados en el ISO/IEC 27001 y está diseñada para asistir a la implementación adecuada de seguridad de la información basada en un enfoque de gestión del riesgo y aplica a todo tipo de organización, que intente



gestionar sus riesgos que pudieran comprometer la seguridad de información de la empresa.

El uso de las **tecnologías de la información** se ha intensificado en las organizaciones independientemente de la naturaleza, giro u actividad. El entorno está en constante evolución, viéndose las empresas en la necesidad de adaptarse a las circunstancias como el uso de distintas fuentes de información, redes sociales, dispositivos móviles, información en la nube, etc. Todas estas herramientas, son necesarias para seguir en competencia y no tener un rezago significativo que pueda comprometer la vida de la misma.

Desafortunadamente esto también implica **riesgos** a los que puede estar expuesta una organización. El robo de información, la suplantación de identidad, o ataques dirigidos, son ejemplos que uno podría suponer están dirigidos sólo a las grandes corporaciones, pero, a través del tiempo la evidencia nos demuestra que, todos estamos expuestos a amenazas que pueden comprometer el activo más importante con el que contamos: la **información**.

En estos días, conectarse a la red y el uso de tecnologías es indispensable, el envío de información es tan natural y forma parte de nuestra vida cotidiana a tal nivel, que simplemente no podemos no estar prevenidos. La misma información que

utilizamos se replica y a menudo, muchas veces a través de una empresa e incluso más empresas; haciendo impredecible conocer en qué manos puede caer la información sensible.

Por esta cuestión, es importante implementar un programa de **gestión de riesgos** de la seguridad de la información. **ISO 27005** nos puede llevar a través de un proceso de gestión de dichos riesgos.

Pero, ¿Por qué es importante?

- Nos permite identificar las necesidades de la organización sobre la seguridad de la información.
- Se identifican los riesgos y el contexto de los mismos.
- Aborda los riesgos de manera eficaz y oportuna, donde y cuando sea necesario.
- Prioriza las acciones a tomar respecto al tratamiento de los riesgos.
- El proceso de gestión de riesgos que nos presenta **ISO 27005** nos presenta las siguientes cláusulas:



Figura 2. 17 Gestión de Riesgo de Seguridad de Información ISO 27005 Fuente: (Karina Melendez, 2009)



2.13. Interrelación de Variables



Figura 2. 18 Interrelación de Variables Fuente: Elaboracion Propia



2.14. Definición de términos básicos

SDK. (Software Development Kit - Kit de desarrollo de software o devkit). Un SDK (Alegsa.Screenshoot, 2013) es un conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware, sistema de computadora, consulta de videojuego, sistema operativo o similar.

Librería (Alegsa, 2013) es un conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tienen una interfaz bien definida para el comportamiento que se invoca.

API (Alegsa, 2013) en inglés una "Application Programing Interface", lo que en español quiere decir "Interfaz de Programación de Aplicaciones", y se refiere a un protocolo de programación para comunicarse con un software en particular, en principio su finalidad es simple: permite a los desarrolladores de software crear programas que interactúen con otras aplicaciones

IDE. (Integrated development environment o Entorno de Desarrollo Integrado) (**Mentoring, 2012**) Es un programa para el



desarrollador de software que combina las funciones de un editor de texto, un intérprete o un compilador y ejecutar run time para simplificar la codificación y depuración.

Sistema operativo: Un sistema operativo (SO, frecuentemente OS, del inglés Operating System) es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación.

SDK - KINECT.

El **SDK (Software Development Kit)** (Murillo, 2013) se trata de una **librería** que nos **facilita** diferentes funciones que nos ayudan a **interactuar** con el **dispositivo Kinect**. Básicamente pedirle información sobre los distintos esqueletos y de sus articulaciones. Kinect, una vez detecta el esqueleto humano, es capaz de facilitarnos información detallada de la posición exacta en el plano (X,Y,Z) de todas y cada una de las articulaciones en las que divide el esqueleto humano. Es gracias a esa información lo que permite que podamos desarrollar aplicaciones que funcionen con la interacción del cuerpo humano, sin necesidad de teclados, ratones ni touchpads.

CAPÍTULO III: MARCO METODOLÓGICO

3.1. Tipo y Diseño de Investigación

3.1.1. Tipo de investigación

De acuerdo a la clasificación general de la investigación (Grajales, 2000) nuestro trabajo es de tipo descriptivo (de desarrollo) y cuasi experimental

De acuerdo a Zorrilla, 1993:43 (Cherrez, 2012) Es **aplicada** por lo que aplica conocimientos de la visión artificial para aplicar en la prueba de los productos virtuales de prendas de vestir y **mixta** porque busca combinar la revisión de documentos en actividades directas sobre el objeto de estudio que es la tecnología computacional.

3.1.2. Diseño de la Investigación

Según Campbell y Stanley nuestra investigación es un diseño **Experimental cuasi experimental.**

El diseño pre experimental “solamente después”

GRUPO EXPERIMENTAL (A) X O1

Donde (A es una asignación no controlada) X es el tratamiento y O1 es la observación al grupo

3.2. Población y Muestra

Unidad de análisis (OA): Sesiones de prueba de prendas de vestir usando la aplicación propuesta

Población: La población está determinada por el número total de prendas de vestir que se incluirá en el sistema que hacen un total de 200 imágenes.

Muestra: La muestra es poblacional ya que se realizará las pruebas con la totalidad de las imágenes que se encuentran en el sistema.

Hipótesis

Mediante la Solución de un probador virtual basada en tecnología Kinect apoyara en la interacción de interfaz natural a la prueba de prendas de vestir.

3.3. Variables

3.3.1. Variable Independiente

Solución de un probador virtual basada en tecnología Kinect.

3.3.2. Variable dependiente:

Interacción de interfaz natural.

3.4. Operacionalización

Tabla 3. 1 Variables Dependientes e Independiente. Fuente: Elaboración Propia

Variable	Dimensión	Indicador	Unidades de medición	Formula	Técnicas e instrumentos de recolección de datos	Descripción
Independiente						
Solución de un probador virtual basada en tecnología Kinect.	Evaluación Tecnológica.	Grado de Confiabilidad del Producto.	%	$GC = NA/NP$	Plantilla de evaluación de software	Es el número porcentual de aciertos para identificar errores del producto. GC= Grado Confiabilidad, NA= Número de Aciertos, NP= Número de Pruebas.



Dependiente						
Interacción de interfaz natural.	Adaptabilidad del sistema con el usuario final.	Precisión	Número	NP/ TP	Observación	<p>Este indicador permite medir el número de precisiones que tiene la aplicación en la prueba de prendas de vestir</p> <p>NP=Numero de pruebas TP=Total de pruebas P= Precisión</p>
		Usabilidad	%	$U=NCE*VC$	Ficha de evaluación de uso	<p>Este indicador mide el Nivel de confiabilidad de la interacción en el usuario final.</p> <p>U=USABILIDAD NCE=Numero De Criterio Evaluado VC=Valor de Criterio</p>



3.5. Métodos, técnicas e instrumentos de recolección de datos

Los métodos y procedimientos para la recolección de datos o información, consistirán en:

3.5.1. Métodos de la Investigación

De Observación: puesto que es un método cuya función primaria e inmediata es recoger información sobre el objeto que se tomara en consideración, para este caso es la API que se está desarrollado.

3.5.2. Técnicas de Investigación

Las técnicas de recolección de datos que se utiliza en el estudio, es:

1. Observación:
2. Encuestas
3. Plantilla
4. Ficha

3.5.3. Instrumentos de recolección de datos

Cuestionario:

Los cuestionarios se utilizan para recopilar información en forma verbal, a través de preguntas que propone el entrevistador. Quienes responden son las personas que están involucradas en el proceso de prueba de prendas de vestir.

Plantilla de evaluación: Este instrumento tiene como objetivo evaluar el nivel de calidad del software, para ello se usará uno

que ya ha sido estudiado y evaluado por un experto, en este caso sería el de Pressman Roger S. Ingeniería de Software un enfoque práctico (Pressman, 2002)

Ficha de evaluación: Este instrumento tiene como objetivo evaluar el nivel de satisfacción del usuario en el software.

Guía de Observación:

Son los análisis que pueden hacer los usuarios, asesores, y jurado calificador de la presente tesis hacia el sistema mientras se hace el desarrollo del software y las pruebas al prototipo.

3.6. Procedimiento para la recolección de datos

3.6.1. Diagrama de Flujo de Procesos

Figura 3.1 Diagrama de Flujo de Procesos Fuente: Elaboracion Propia



3.6.2. Descripción de Procesos.

1. Análisis de Requerimientos del Sistema.

Fuente de Consulta:

Clientes que compran

Datos a Conseguir:

- ❖ Requerimientos funcionales del sistema.
- ❖ Requerimientos no funcionales del sistema.
- ❖ Expectativa de usuarios.

Materiales y Herramientas:

- ❖ Materiales de Oficina
- ❖ Cuaderno de Apuntes
- ❖ Lapiceros

Técnica de Recolección de Información

- ❖ Entrevista: A los interesados
- ❖ Observación. Requisitos mínimos de hardware y software de la PC para que responda con normalidad al sistema.

2. Diseño del Prototipo.

Fuente de Consulta:

Fuentes escritas: Libros de Programación de sistemas C#, JSP, HTML.

Datos a Conseguir:

- ❖ Interfaces del Probador Virtual.
- ❖ Interfaces del Sistema E-commerce
- ❖ Diseñar la Base de Datos.

Materiales y Herramientas:

- ❖ Materiales de Oficina
- ❖ Cuaderno de Apuntes
- ❖ Lapiceros
- ❖ Computadora personal. (IDE NETBEANS y MS Visual Studio)

Técnica de Recolección de Información

- ❖ Análisis: Realizar la programación extrema para el sistema.

3. Desarrollo del Prototipo.

Fuente de Consulta:

Fuentes escritas: Libros de Programación de sistemas Windows, Libros de desarrollo de Realidad Aumentada y E-commerce.

Datos a Conseguir:

- ❖ Programación de los procesos de Realidad Aumentada.
- ❖ Programación de los procesos de E-commerce
- ❖ Programación de los procesos de Probador Virtual.
- ❖ Conectar con la Base de Datos.

Materiales y Herramientas:

- ❖ Materiales de Oficina
- ❖ Cuaderno de Apuntes
- ❖ Lapiceros
- ❖ Computadora personal.(IDE NETBEANS y MS Visual Studio)

Capacidades.

- ❖ Razonamiento y Análisis para el desarrollo de los módulos del sistema E-commerce y probador virtual.
- ❖ Observación:

4. Pruebas al Prototipo.

Fuente de Consulta:

Fuentes escritas: Libros de Programación RUP, Libros de Programación C#, JSP, HTML.

Datos a Conseguir:

- ❖ Poner a prueba al sistema en diferentes escenarios.
- ❖ Depurar los posibles errores.

Materiales y Herramientas:

- ❖ Materiales de Oficina
- ❖ Cuaderno de Apuntes.
- ❖ Lapiceros
- ❖ Computadora personal. (IDE NETBEANS y MS Visual Studio)

Capacidades.

- ❖ Análisis de las limitaciones del sistema en el desarrollo de los módulos del sistema.
- ❖ Observación: Depurar errores funcionales del Sistema.

5. Análisis Económico del Software.

Fuente de Consulta:

Fuentes escritas: Libros de evaluación de requerimientos de software para el desarrollo de sistemas informáticos.

Datos a Conseguir:

- ❖ Evaluación Económica del Software.
- ❖ Análisis de la Viabilidad del Sistema.
- ❖ Evaluación Económica de Recurso Humano.

Materiales y Herramientas:

- ❖ Materiales de Oficina
- ❖ Cuaderno de Apuntes
- ❖ Lapiceros
- ❖ Computador personal. (Microsoft Office Project)

Capacidades.

- ❖ Razonamiento y Análisis para el desarrollo de la evaluación económica del sistema.
- ❖ Observación: Determinar si el sistema resulta viable.

3.7. Análisis Estadístico e Interpretación de los datos

El procesamiento estadístico de la información que se ingresó al presente estudio fue recogido mediante el software estadístico denominado SPSS. Posteriormente se hizo un análisis para obtener los balances preliminares y luego se realizaron las conclusiones finales de la información estudiada.

Se empleó una PC CORE I7, con ambiente de Windows 8. Los textos se procesaron con MS Word 2013, y las tablas y gráficos resultantes, a los cuales se ingresó la información pertinente. Asimismo, se realizaron con SPSS. Posteriormente, adecuamos todo el trabajo final a un solo formato tecnológico.

3.8. Criterios éticos

Los criterios éticos que se respetan en el presente proyecto de tesis es el Código Deontológico del Colegio de Ingenieros del Perú en su Capítulo III “Faltas Contra la Ética Profesional y Sanciones” y su Sub Capítulo II “De la Relación con El Público” en su Artículo 106 expresa:

Los ingenieros, al explicar su trabajo, méritos o emitir opiniones sobre temas de ingeniería, actuarán con seriedad y convicción, cuidando de no crear conflictos de intereses, esforzándose por ampliar el conocimiento del público a cerca de la ingeniería y de los servicios que presta a la sociedad.

El presente proyecto de investigación expresara en la medida de lo posible lo más claro y conciso su contenido con el fin de generar un aporte al desarrollo de E-commerce y Realidad Aumentada en la colectividad Lambayecana, al estar disponible para el público en general la presente tesis.

3.9. Criterios de rigor científico

Criterios	Características éticas de los criterios
Consentimiento informado.	El Tesista deberá estar de acuerdo con ser informante y conocer sus derechos y responsabilidades durante la investigación.
Confidencialidad.	Asegurar la protección de identidad de sus fuentes, como también de las personas que participan como informantes de la investigación.
Manejo de Riesgos.	La investigación requiere de una eficiencia y no de un beneficio personal para realizar una investigación consistente.
Observación Participante.	La participación del Tesista requiere una responsabilidad ética por los efectos y consecuencias que pueden surgir durante la investigación.
Entrevistas.	La comunicación con el Tesista y el entrevistado no debe provocar actitudes que induzcan a coincidir con las respuestas del investigador.
Grabaciones de Audio y Video.	Los archivos deben ser guardados cautelosamente para fines de la investigación, preservando el anonimato, la confidencialidad y el respeto de los participantes.

Tabla 3. 2 Variables Dependientes e Independiente. **Fuente:** Elaboración Propia



CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADO.

4.1. Resultado en tablas – Variable Dependiente.

Tabla 4. 1 Resultado en tablas y gráficos Variable independiente. Fuente:

Elaboración Propia

Ficha de usabilidad						
N°	Criterio	Valor				Análisis de Datos
		Si		No		
		F	%	F	%	
1	Es una interfaz limpia, sin ruido visual	90	90%	10	10%	De acuerdo a los consultados, el 90% piensan que la interfaz es limpia y sin ruido visual, mientras que el 10% dijo todo lo contrario, es decir, algunas dicen que la interfaz no es muy limpia.
2	Uso correcto del espacio visual de la API	95	95%	5	5%	Los resultados obtenidos muestran que un 95% del espacio visual de la API es correcta, lo que facilito la interacción y la participación de los usuarios. Así mismo en un 5% es considerada como significativas
3	Uso consistente de los controles de la interfaz	85	85%	15	15%	De acuerdo a los resultados un 85% de los encuestados respondió Que la interfaz tiene controles consistentes, aunque un 15% refieren que se desarrollaron controles no muy consistentes.
4	Se emplea un gestos claros y concisos	93	93%	7	7%	Del total de sujetos encuestados un 93% está de acuerdo con los gestos que fue especializado la API, un 7% refiere que no son



						%concisos.
5	Interacción amigable, familiar y cercano	95	95%	5	5%	Los datos obtenidos refieren que en un 95% la interacción es amigable, familiar y cercana, solo un 5% refiere que no lo es..
6	Interacción de gestos comprensibles por cualquier usuario	90	90%	10	10%	Según los resultados obtenidos del total de encuestados un 90% respondió que las interacciones de los gestos son comprensibles, y el 10 % restante refiere que no son comprensibles.
7	El usuario tiene todo el control sobre la interfaz	90	90%	10	10%	El 90% de los encuestados refiere que tienen un control total de la interfaz, pero el 10% refiere lo contrario.
8	Se ha controlado el tiempo de respuesta	87	87%	13	13%	Los datos registrados evidencian que el 87% hay un tiempo de respuesta controlado, aunque hubo 13% que dijo todo lo contrario.

Los ítems con que se midió cada indicador fueron los números 1, 2, 3, 4, 5, y 8 contenidos en el cuestionario aplicado (ver anexo N° 1), el cual fue respondido por usuarios que interactuaron con la aplicación.

El análisis descriptivo de la variable independiente, consistió en el estudio de la frecuencia absoluta y relativa de los datos, con la finalidad de determinar el comportamiento y/o tendencia de los mismos.

El primer nivel SI equivale a 10 y representa la presencia del atributo.

El segundo nivel NO equivale a 0 y representa la ausencia del atributo.



Figura 4. 1 Grafica de ficha de usabilidad Fuente: Elaboracion Propia



Interpretación:

De acuerdo al gráfico anterior la tendencia de los datos tiene un comportamiento aceptable, lo que significa que según referencia de los usuarios se ha hecho uso de una tecnología y gestos adecuados, por lo tanto, es evidente que el uso de la aplicación es muy aceptable en los usuarios que quieren mejorar la interacción entre usuario y hardware.

Tabla 4.2. Resultado en tablas y gráficos del Software Variable dependiente.

Fuente: (Pressman, 2002)

Plantilla de evaluación de Uso

Dimensiones	Indicadores	Deficiente		Regular		Bien		Excelente	
		F	%	F	%	F	%	F	%
Eficacia	Al utilizar varios objetos la API fue:			2	80	1	20		

Dimensiones	Indicadores	Deficiente	Regular	Bien	Excelente
-------------	-------------	------------	---------	------	-----------



		F	%	F	%	F	%	F	%
Productividad	El tiempo que se usó para completar la tarea fue:			2	80	1	20		
	El esfuerzo del usuario para completar la tareas fue:			2	80	1	20		

Dimensiones	Indicadores	Bajo		Alta		Bien		Excelente	
		F	%	F	%	F	%	F	%
Satisfacción	La satisfacción que el usuario siente al usar la API fue:			2	80	1	20		

Interpretación:

De acuerdo a la tabla anterior la tendencia de los datos tiene un comportamiento aceptable, lo que significa que según referencia de los encuestados se ha hecho uso de los gestos adecuados, por lo tanto, es evidente que el uso de la aplicación es muy aceptable en los usuarios que quieren mejorar la interacción entre usuario y hardware.

DISCUSIÓN DE RESULTADOS

En cuanto a la dimensiones sobre adaptabilidad de la interacción de interfaz natural a los usuarios en el uso de la API objeto de estudio, se analizó su primer indicador el cual fue la exactitud, para medir la interacción por la API, se pudo determinar que al momento del ingreso de los gestos en la API los usuarios se sienten complacidos al poder interactuar con una aplicación sin tener que usar un teclado o un mouse.



Al satisfacer a los usuarios durante la interacción en la aplicación, se presentó una inspiración de confianza para que puedan interactuar, el autor López (2013) plantea que los gestos implementados fueron los adecuados para la aplicación porque todos los que realizaron las pruebas estuvieron de acuerdo que recordarían fácilmente los movimientos realizados si volvieron a interactuar con la aplicación. En este caso los usuarios en esta aplicación experimentaron una interacción satisfactible.

Siguiendo con la dimensión de confiabilidad, el segundo indicador medido fue el de consistencia, donde los usuarios experimentaron al momento de controlar los comandos con los gestos tuvo un excelente recibimiento, de la misma manera los usuarios manifiestan que cuando tienen algún inconveniente referido a los gestos, la persona encargada de estar al lado de la aplicación se esmera y se interesa en explicar cualquier inconveniente que se presente durante la estancia en la prueba de esta API, igualmente los usuarios refirieron que la aplicación estuvo muy entendible.

El autor Lee San Hyuk (2013), refiere a este punto que al usuario hay que brindarle una interfaz natural eficiente y agradable para reconocer comandos de voz y el movimiento del cuerpo, especialmente de las manos, los brazos y movimiento de la cabeza con el fin de controlar el ordenador. En este caso los usuarios refirieron que la aplicación tiene gestos muy entendibles.



4.2. Contrastación de la Hipótesis

Contrastación de la Hipótesis Principal

Mediante la Solución de un probador virtual basada en tecnología Kinect apoyara en la interacción de interfaz natural a la prueba de prendas de vestir.

Tabla 4. 2: Resultado en tablas y gráficos Variable independiente. **Fuente:**
Elaboración Propia

Item	Aceptable	No aceptable
Eficiencia	65	35
Eficacia	75	25
Precisión	55	45
Sensibilidad	63	37
Control	61	39
Desempeño	78	22

Modelo Lógico

H0 NO apoyara en la interacción de interfaz natural a la prueba de prendas de vestir.

H1 SI apoyara en la interacción de interfaz natural a la prueba de prendas de vestir.

Modelo Matemático

$$H0: O = E \quad H1: O = E$$

Modelo estadístico

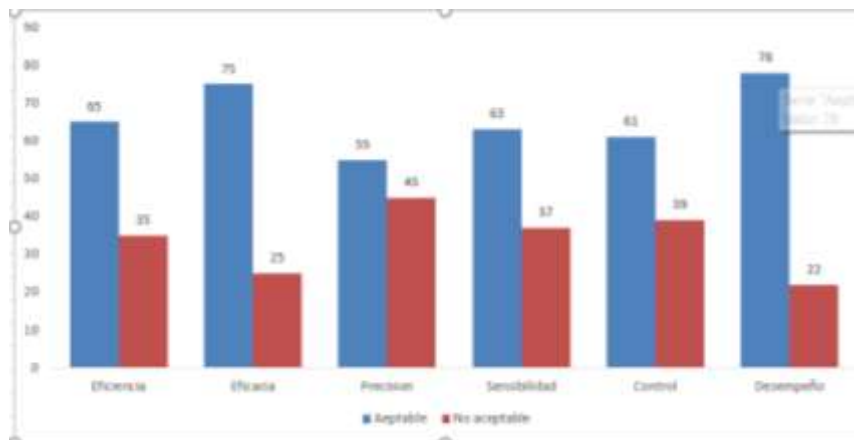


Figura 4. 2 Variable de resultados en el proceso de análisis Fuente: Elaboracion

Propia

**ANALISIS DE VARIABLES
FRECUENCIAS OBSERVADAS**

	Si	No	Totales
Interacción agradable	9.00	70.00	79
Interacción no agradable	7.00	14.00	21
Totales	16	84	100

Tabla 4. 3 análisis de variable de frecuencia observada. Fuente: Elaboración

Propia

**ANALISIS DE VARIABLES
FRECUENCIAS DESEADAS**

	Si	No	Totales
Interacción agradable	12.64	66.36	79
Interacción no agradable	3.36	17.64	21
Totales	16	84	100

Tabla 4. 4 análisis de variable de frecuencia deseada. Fuente: Elaboración

propia

Nivel de Significación

95% de confiabilidad

$\alpha=0,05$

Zona de Rechazo de H0

gl= Grado de Libertad

$gl = (c-1)(f-1) \quad gl = (2-1)(2-1) \quad gl = 1$



El percentil de la distribución χ^2 con $(2-1)(2-1)=1$ Grado de Libertad $\chi_{0,952}(1)=3,84$ según a Tabla de Distribución del CHI – CUADRADO

Observados	Esperados	O-E	(O-E) ²	(O-E) ² /E
9	12.64	-3.64	13.25	1.05
7	3.36	3.64	13.25	3.94
70	66.36	3.64	13.25	0.20
14	17.64	-3.64	13.25	0.75
100	100		χ^2	5.94

Tabla 4. 5 Tabla de distribución del CHI-CUADRADO. Fuente: Elaboración propia

Con 1 grados de libertad y 95% de confiabilidad, aplicando la prueba χ^2 (Chi-Cuadrada) se tiene que el valor tabular es igual a 3.841; de acuerdo a los resultados obtenidos con los datos tomados de la encuesta se ha calculado el valor de χ^2 que alcanza a 5.94; lo que implica que se rechaza la hipótesis nula, aceptando la hipótesis alterna o de trabajo que dice: Mediante la Solución de un probador virtual basada en tecnología Kinect Si apoyara en la interacción de interfaz natural a la prueba de prendas de vestir.



CAPÍTULO V: DESARROLLO DE PROPUESTA

5.1. Propuesta.

En el presente capítulo se suministrará de manera detallada el proceso de la aplicación a realizar sobre Realidad Aumentada basada en Tecnología Kinect para optimizar el proceso de la interacción en la prueba de y compra de prendas de vestir.

Este capítulo está compuesto por los objetivos del desarrollo, la justificación de la misma, así como el impacto y otros detalles de la aplicación.

5.2. Justificación.

Aprovechando la disponibilidad y gran avance de las tecnologías de reconocimiento gestos humanos en tres dimensiones, se proyecta el desarrollo de una aplicación con el sensor Kinect. La interfaz del sensor de Kinect presenta grandes ventajas para el desarrollo de la aplicación ya que reconoce gestos, comandos de voz, objetos e imágenes.

El proyecto es factible debido a que este dispositivo es asequible, presenta tecnología de punta, mismo que puede ser utilizado como herramienta, y ser aplicados en distintas áreas como: Educación, Entretenimiento, Médicas, Comercial, Industrial, permitiendo a los usuarios controlar e interactuar sin la necesidad de tener contacto físico con el dispositivo.

Con esta aplicación se estima lograr un cambio en el sistema de compra de ropa para el beneficio de todos los que interviene en dicha compra como clientes, personal de venta, y que esta herramienta pueda ser parte de una evolución en el proceso de probarse una prenda de vestir y comprar dicha prenda.

5.3. Objetivos.

5.3.1. Objetivos generales de la propuesta.

Construir una aplicación que pueda interactuar con el ser humano de una manera más eficaz e intuitiva para el proceso de prueba y compra de prendas de vestir.

5.3.2. Objetivos específicos de la propuesta.

1. Debe ser capaz de detectar los gestos de una persona, implementados en el API
2. Posibilitar a la aplicación para cambiar entre varios modelos de prenda de vestir.
3. Posibilitar a la aplicación de una manera independiente agregue objetos virtuales a la base de datos.
4. Habilitar a la aplicación para hacer una captura de pantalla del usuario con la prenda.
5. Habilitar al sistema para colocar modelos tridimensionales de prendas en el usuario.
6. Disminuir los tiempos de prueba de prenda de vestir.

7. Habilitar a la aplicación para seleccionar las tallas de las prendas a comprar.
8. Reducir el tiempo en la atención asía el cliente
9. Reducir el tiempo en el registro de compra de prendas de vestir

5.4. Caracterización de los Gestos de Interacción

El sistema a construir se basa en el reconocimiento de gestos con las manos para la manipulación de un objeto virtual, específicamente de un modelo tridimensional de una prenda de vestir.

Por tal motivo, entonces definiremos un gesto el cual puede llegar a ser un término especializado para diferentes disciplinas de estudio. Por ejemplo, en semiótica (estudio de los signos), los gestos pueden significar, palabras, imágenes, fórmulas matemáticas, mapas.

En las artes, un gesto es usado para describir aspectos expresivos de la danza [40]. Finalmente, en diseño de interacción, los gestos se enfocan en la manipulación de la experiencia de usuario en NUI.

Eric Hulteen y Gord Kurenbach en 1990 en su publicación llamada: *“Gestures in Human Computer Interaction”*, lo definieron como:

“Un gesto es un movimiento del cuerpo que contiene información. Mover la mano para decir adiós es un gesto. Presionar una tecla en un teclado no es un gesto porque el movimiento del dedo para presionar la tecla no es observado y es insignificante.”

Para reiterar bajo el contexto de interfaces de usuario, un gesto:

- Expresa un comando simple,
- Es arbitrario y natural, • Está basado en convención y
- Puede ser malinterpretado.

Etimológicamente la palabra gesto viene del latín *gestus*, la cual era usada para referirse al lenguaje corporal de las personas. Es decir, la expresión facial, movimiento de las manos y postura del cuerpo que indican su estado de ánimo. *Gestus* viene del verbo *gerere*, que significa traer o llevar a cabo. *Gerere* también tiene el significado de administrar (llevar a cabo las cosas).

Una vez definido un gesto, y entendiendo que estos pueden ser malinterpretados como cuando alguien levanta la mano, que puede entenderse que una persona requiere atención (de llevarse a cabo en un restaurante) u otra cosa (si fuese en otro contexto).

Gestos con las manos, son aquellos que se realizan por una o dos manos, esta a su vez es la categoría con mayor número de gestos debido a la capacidad de la mano humana para adquirir un gran número de configuraciones claramente discernibles, por eso la importancia del lenguaje de los signos.

Según cada escenario de aplicación, los gestos con las manos se pueden dividir en varias categorías, tales como:

- Gestos de conversación.
- Gestos de control.
- Gestos de manipulación.
- Gestos comunicativos.

La categoría en la cual estará basada esta aplicación son los gestos de control, y en la interfaz basada en la visión (VBI) con el objetivo de manipular objetos virtuales mediante el análisis de los gestos que registran. Algunas aplicaciones de control de visualización demuestran el potencial de señalar gestos en HCI.

Se definirá el gesto para referirse a cualquier forma de movimiento que puede ser utilizado como una entrada o interacción para controlar o influir en una aplicación.

Gestos de interacción en el diseño de la aplicación

Básicamente cuando se menciona reconocimiento de gestos con las manos, se trata de asignar un cierto número de movimientos de la mano a una determinada acción en este caso deslizar y elegir una prenda de vestir.

Como se mencionó con anterioridad los gestos son convención, tal es el caso de Samsung con su SMART TV [58], en la integración de comandos empleando gestos con las manos.

Dichos comandos empleados en la SMART TV nos ayudarán a definir cada uno de los gestos que emplearemos en dicha aplicación, puesto que son similares a los gestos que maneja el *Kinect* para la interacción con los videojuegos.

Para elaborar el diseño de cada uno de los gestos que la aplicación es capaz de reconocer existen diferentes técnicas [59]. La primera técnica es utilizar redes neuronales, las cuales deben entrenarse para tener mejor precisión y calidad de detección. La segunda técnica es definir algorítmicamente cada gesto y la última técnica es comparando con una serie de plantillas preestablecidas.

Deslizar

La aplicación debe ser capaz de detectar el *swipe* o deslizamiento. Este gesto básicamente, se trata de mover la mano simulando una barra de desplazamiento vertical para que el usuario pueda elegir la pieza arqueológica de su preferencia y poder manipularla.

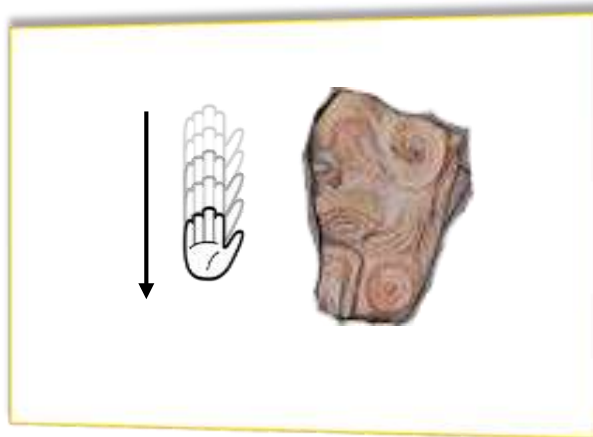
Para que este gesto pueda ser válido se deben capturar 20 coordenadas de la mano derecha o izquierda, la mano derecha para avanzar a la siguiente pieza o la mano izquierda para seleccionar la pieza anterior.

Este gesto está compuesto por dos acciones para darle mayor libertad de uso al usuario. La primera parte del gesto se realizara únicamente con la mano derecha moviéndola de abajo hacia arriba en línea recta, para poder desplazar y elegir las piezas disponibles, como en la Ilustración 24.



Deslizamiento abajo-arriba

La segunda parte del gesto implica solamente el movimiento de la mano izquierda está desplazándose de arriba hacia abajo en línea recta, para poder desplazar y así mismo elegir la pieza de nuestra preferencia, véase Ilustración 25.



Deslizamiento arriba-abajo

Con cada una de los métodos que componen el gesto de deslizamiento, el gesto queda completado y el usuario pueda cambiar o elegir la pieza arqueológica de su preferencia.

En el Capítulo 5 “Evaluación y Resultados”, se mostrará con más detalle que después de concluir las pruebas integrales los resultados arrojaron que este gesto no era comprendido, además que el usuario presentaba ciertas dificultades por lo cual el gesto sufrió ciertas modificaciones.

Para replantear el gesto, este se definió en dos partes (ver Ilustración 26): *SwipeLeftToRight* y *SwipeRightToLeft*. El mismo ahora se ejecutara solo con el uso de una mano, específicamente la mano derecha.

Este gesto para ser válido tiene que tener una duración entre 500 y 1500 milisegundos con una distancia promedio de 20 cm.



Modificación del *Swipe* (deslizamiento)

Además de la retroalimentación, en la parte superior de la pantalla el usuario podrá ver la galería de piezas disponibles para la interacción así como identificar la pieza arqueológica con la cual está trabajando (ver Ilustración 27).



Menú de objetos disponibles y seleccionada

Diseño Conceptual

El diseño conceptual, se desarrolla como primer acercamiento para contemplar las bases de desarrollo en las cuales se agrupara el sistema total. El diseño conceptual es una visión general del sistema, en el cual se muestran todos los módulos y las posibles relaciones entre ellos [56].

En la Ilustración se muestra el diseño conceptual general del sistema en el cual se engloba cada uno de los módulos a construir para lograr el funcionamiento del sistema abarcando cada uno de los requerimientos o gestos que el sistema será capaz de detectar.

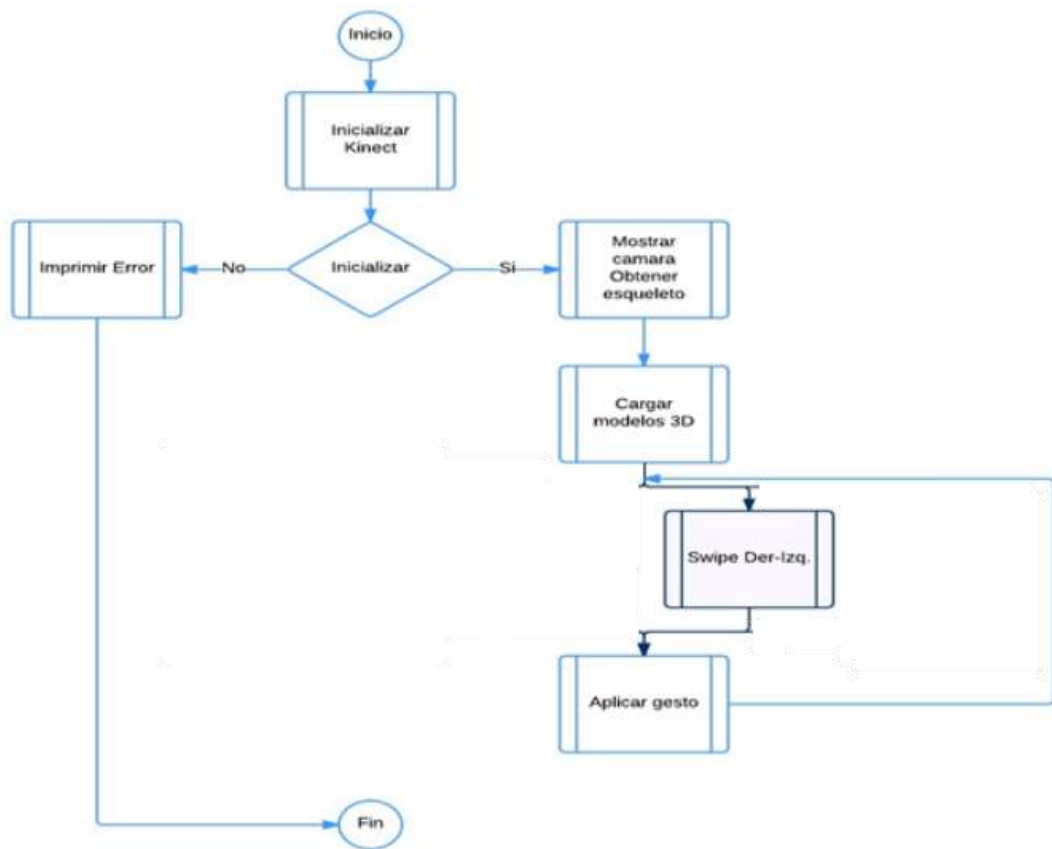


Diagrama conceptual del sistema de reconocimiento de gestos

Plantilla de Especificación Operacional (OST)

Deslizamiento

El OST para este gesto lo definimos en dos partes por tanto es importante tener para cada una su OST correspondiente.

Primero se plantea el funcionamiento normal de detección del gesto deslizamiento con la mano derecha e izquierda dado que es similar el comportamiento del sistema, véase Tabla 8.



Escenario #:1		Objetivo Usuario: Gesto de <i>SwipeDowntoUp</i> (DeslizarAbajo- <i>SwipeUptoDown</i> (Deslizar Arriba-Abajo)	
Objetivo del Escenario: Dadas 20 [40] coordenadas de la mano derecha detectar el <i>SwipeDowntoUp</i> 0 20 coordenadas de la mano izquierda para detectar el <i>SwipeUptoDown</i> .			
Fuente	Paso	Acción	Comentarios
Usuario	1	Ejecuta el sistema.	
Sistema	2	Obtiene las coordenadas de la mano derecha o mano izquierda.	Estas coordenadas se obtienen cada segundo.
Usuario	3	Mueve la mano derecha de abajo hacia arriba de forma recta o la mano izquierda de arriba hacia debajo de forma recta.	
Sistema	4	Almacena 20 coordenadas de la mano derecha o 20 coordenadas de la mano izquierda.	Estas coordenadas serán almacenadas en un vector.
Sistema	5	Analiza y determina si las coordenadas corresponden a un <i>SwipeDowntoUp</i> o a un <i>SwipeUptoDown</i> .	



Sistema	6	Cambia la pieza arqueológica en la pantalla.	
---------	---	--	--

OST *SwipeDowntoUp* o *SwipeUptoDown* detectado

Seguidamente se describe el escenario en la Tabla **anterior** cuando el sistema no sea capaz de detectar el gesto debido a que el movimiento de la mano derecha o izquierda no es el adecuado para poder definir el gesto de deslizamiento.

Escenario #:	2	Gesto de <i>SwipeDowntoUp</i> (Deslizar Abajo-Arriba) y <i>SwipeUptoDown</i> (Deslizar Arriba-Abajo) no detectado	
Objetivo del Escenario:	Dadas 20 coordenadas de la mano derecha o 20 coordenadas de la mano izquierda verificar que no es un gesto deslizamiento válido.		
Fuente	Paso	Acción	Comentarios
Usuario	1	Ejecuta el sistema.	
Sistema	2	Obtiene las coordenadas de la mano derecha o mano izquierda.	Estas coordenadas se obtienen cada segundo.
Usuario	3	Mueve la mano derecha de abajo hacia arriba de forma recta o la mano izquierda de arriba hacia debajo de forma recta.	



Sistema	4	Almacena 20 coordenadas de la mano derecha o 20 coordenadas de la mano izquierda.	Estas coordenadas serán almacenadas en un vector.
Sistema	5	Analiza y determina si las coordenadas corresponden a un <i>SwipeDowntoUp</i> o a un <i>SwipeUptoDown</i> .	
Sistema	6	No realiza una acción.	

OST Deslizamiento no detectado

Los escenarios OST de deslizamiento nos servirán para verificar que este gesto se implementó de forma adecuada.

Plantilla Especificación Funcional (FST)

Deslizamiento

El deslizamiento maneja dos métodos en la anterior se muestran las variables para la detección del *SwipeDowntoUp*:

Clase de Prueba	<i>SwipeDowntoUp</i>
Variables	
Declaración	Descripción



<pre>public struct VectorG { public float X; public float Y; public float Z; public DateTime date; }</pre>	<p>Estructura de datos que almacena las coordenadas de la mano derecha en los ejes X, Y Z, además del tiempo en el que se capturan dichas coordenadas.</p>
<pre>const int SwipeMininalDuration = 250</pre>	<p>Almacena la duración mínima del gesto en milisegundos.</p>
<pre>const int SwipeMaximalDuration = 1500</pre>	<p>Almacena la duración máxima del gesto en milisegundos.</p>
<pre>const float SwipeMinimalLength = 0.4f</pre>	<p>Almacena la longitud mínima del gesto.</p>
<pre>const float SwipeMaximalHeight = 0.2f</pre>	<p>Almacena la elevación máxima del gesto.</p>
<pre>List positionList</pre>	<p>Lista que almacena los puntos de la mano derecha en la estructura de datos</p>



	VectorG.
Boolean swipe	Bandera que indica si el gesto fue aceptado.
Métodos	
Void Swipe1(Boolean s)	Aplica el cambio de pieza arqueológica cuando el gesto es aceptado.

FST *SwipeDowntoUp* detectado

De forma similar la **anterior** tabla muestra el FST para el reconocimiento del gesto *SwipeUptoDown*.

Clase de Prueba	<i>SwipeDowntoUp</i>
Variables	
Declaración	Descripción
<pre>public struct VectorG { public float X; public float Y; public float Z; public DateTime date; }</pre>	Estructura de datos que almacena las coordenadas de la mano izquierda en los ejes X, Y Z, además del tiempo en el que se capturan dichas coordenadas.
<pre>const int SwipeMininalDuration =</pre>	Almacena la duración mínima del gesto



250	en milisegundos.
const int SwipeMaximalDuration = 1500	Almacena la duración máxima del gesto en milisegundos.
const float SwipeMinimalLength = 0.4f	Almacena la longitud mínima del gesto.
const float SwipeMaximalHeight = 0.2f	Almacena la elevación máxima del gesto.
List positionList2	Lista que almacena los puntos de la mano izquierda en la estructura de datos VectorG.
Boolean swipe	Bandera que indica si el gesto fue aceptado.
Métodos	
Void Swipe2(Boolean s)	Aplica el cambio de pieza arqueológica cuando el gesto es aceptado.

FST *SwipeUptoDown* detectado



Plantilla de Especificación de Estados (SST) Deslizamiento

El deslizamiento se compone por 6 estados de inicio a fin para completar el gesto, los cuales se muestran en la Tabla:

Nombre Estado	Descripción
Inicio	Define el estado inicial de la máquina de estados, obteniendo las coordenadas de las manos.
Verificar_Swipe	Verifica que ya se tengan almacenados 20 puntos de la mano derecha o izquierda.
Swipe1	Verifica que los puntos de la mano derecha sean válidos para <i>SwipeDowntoUp</i> .
Swipe2	Verifica que los puntos de la mano izquierda sean válidos para <i>SwipeUpDown</i> .
Aplicar_Swipe	Muestra la siguiente pieza arqueológica
Fin	Termina la detección del gesto
Función/Parámetro	Descripción
List positionList	Lista que contiene los valores de las coordenadas <X, Y, Z> correspondientes a la mano derecha.
List positionList2	Lista que contiene los valores de las coordenadas <X, Y,Z> correspondientes a la mano izquierda.
Boolean Swipe	Almacena el valor para saber qué tipo de Swipe se



	detectó.
--	----------

Swipe1()	Verifica que sea un <i>SwipeDown</i> to <i>Up</i> .	
Swipe2()	Verifica que sea un <i>SwipeUp</i> to <i>Down</i> .	
Estados/Siguientes Estados	Condición de transición	Acción
Inicio		Capturar 20 puntos de mano derecha o mano izquierda
Inicio	No hay transición	
Verificar_Swipe	Tener 20 puntos almacenados	
Swipe1	No hay transición	
Swipe2	No hay transición	
Aplicar_Swipe	No hay transición	
Fin	No hay transición	
Verificar_Swipe		Tener 20 puntos de la mano derecha o izquierda almacenados.
Inicio	No hay transición	
Verificar_Swipe	No hay transición	



Swipe1	Verificar si los 20 puntos almacenados en positionList de la mano derecha corresponden al SwipeDowntoUp	
Swipe2	Verificar si los 20 puntos almacenados en positionList2 de la mano izquierda corresponden al SwipeUptoDown	
Aplicar_Swipe	No hay transición	
Fin	No hay transición	
Swipe1		Validar punto a punto para saber si es <i>swipe=1</i>
Inicio	No hay transición	
Verificar_Swipe	No hay transición	
Swipe1	No hay transición	



Swipe2	No hay transición	
Aplicar_Swipe	Si <i>swipe</i> =1	
Fin	Si <i>swipe</i> =0	
Swipe2		Validar punto a punto para saber si es <i>swipe</i> =2
Inicio	No hay transición	
Verificar_Swipe	No hay transición	
Swipe1	No hay transición	
Swipe2	No hay transición	
Aplicar_Swipe	Si <i>swipe</i> =2	
Fin	Si <i>swipe</i> =0	
Aplicar_Swipe		
Inicio	No hay transición	
Verificar_Swipe	No hay transición	
Swipe1	No hay transición	
Swipe2	No hay transición	
Aplicar_Swipe	<i>swipe</i> =1 <i>swipe</i> =2	o Mostrar la pieza anterior o siguiente de las piezas arqueológicas disponibles



Fin	No hay transición	
-----	-------------------	--

SST para deslizamiento

Plantilla de Especificación Lógica (LST)

Deslizamiento

La implementación del deslizamiento según las especificaciones, la lógica de desarrollo para este gesto se encuentra en el siguiente LST en la Tabla 21, los cambios realizados después de las pruebas se observan en negritas.

Parámetros	Descripción
<pre>public struct VectorG { public float X; public float Y; public float Z; public DateTime date; }</pre>	Estructura de datos que almacena las coordenadas de la mano derecha en los ejes X, Y, Z, además del tiempo en el que se capturan dichas coordenadas.
<code>const int SwipeMininalDuration = 250</code>	Almacena la duración mínima del gesto en milisegundos.
<code>const int SwipeMaximalDuration = 1500</code>	Almacena la duración máxima del gesto en milisegundos.
<code>const float SwipeMinimalLength = 0.4f</code>	Almacena la longitud mínima del gesto.



const float SwipeMaximalHeight = 0.2f	Almacena la elevación máxima del gesto.
List positionList	Lista que almacena los puntos de la mano derecha en la estructura de datos <i>VectorG</i> .
List positionList2	Lista que almacena los puntos de la mano izquierda en la estructura de datos <i>VectorG</i> .
Boolean swipe	Bandera que indica si el gesto fue aceptado.

Declaración de librerías: Microsoft.Kinect, System;
Definir constantes
Obtener las coordenadas de las manos usando Kinect.
<pre>void Sensor_Skeleton_FrameReady(object sender, SkeletonFrameReadyEventArgs e);</pre>



```

void SwipeDowntoUp()
{
    int start = 0;
    for index = 0 until 19 do
//La comprobación que se realiza es para ver si el movimiento de la mano sigue
siempre //la misma altura y en la dirección correcta.

        if ((Abs(positionList[0].X - positionList[index].X) >
SwipeMaximalHeight) || (positionList[index].Y - positionList[index + 1].Y >
-0.01f))
            if ((Math.Abs(positionList[0].Y - positionList[index].Y) >
SwipeMaximalHeight) || (positionList[index].X - positionList[index + 1].X >
0.01f))
                do
                    start = index;
//La siguiente comprobación es para ver si el movimiento de la mano tiene la
misma //longitud

                if ((Abs(positionList[index].Y - positionList[start].Y) >
SwipeMinimalLength))
if ((Math.Abs(positionList[index].X - positionList[start].X) >
SwipeMinimalLength)
    
```

```

do
    totalMilliseconds = (positionList[index].date -
positionList[start].date).TotalMilliseconds;

//La ultima comprobación es para determinar si cumple con el tiempo minino y
máximo //de la duración del gesto
    if (totalMilliseconds >= SwipeMininalDuration && totalMilliseconds <=
SwipeMaximalDuration) do
        figseleccionada = ((figseleccionada + 1) % 5);           end-if
    end-if
end-if end-of-for
}
    
```



```

void SwipeUptoDown()
{
    int start = 0;
    for index = 0 until 19 do
//La comprobación que se realiza es para ver si el movimiento de la mano sigue
siempre //la misma altura y en la dirección correcta.

        if ((Abs(positionList2[0].X - positionList2[index].X) >
SwipeMaximalHeight) || (positionList2[index].Y - positionList2[index + 1].Y <
-0.01f))
            if ((Math.Abs(positionList[0].Y - positionList[index].Y)
>SwipeMaximalHeight) || (positionList[index].X - positionList[index + 1].X < -
0.01f))
                do
                    start = index;

//La siguiente comprobación es para ver si el movimiento de la mano tiene la
misma //longitud

                if ((Math.Abs(positionList2[index].Y - positionList2[start].Y)
> SwipeMinimalLength))
if ((Math.Abs(positionList[index].X - positionList[start].X) >

```

SwipeMinimalLength))

```

do
    totalMilliseconds = (positionList2[index].date -
positionList2[start].date).TotalMilliseconds;
//La ultima comprobación es para determinar si cumple con el tiempo minino y
máximo //de la duración del gesto

    if (totalMilliseconds >= SwipeMininalDuration && totalMilliseconds <=
SwipeMaximalDuration) do
        figseleccionada= ((figseleccionada - 1 + 5) % 5);    end-if
end-if

end-if    end-of-for
}
    
```

LST para deslizamiento

5.5. Herramientas tecnológicas escogidas.

ITEM	DESCRIPCION	
Equipos	Computadora personal.	
	Dispositivo Móvil Personal	
	KINECT XBOX ONE	
	Adaptador Kinect	
Herramientas	Software Para Probador Virtual	a) VISUAL C#
		b) AUTODESK 3DS MAX
		c) SDK KINECT 2.0
		d) KINECT SPEECH LANGUAGE PACK 11.0
		e) Visual Studio 2015
	Software Para Pagina Web	a) HTML
		b) JAVASCRIPT
		c) JAVA SERVER PAGE (JSP)
		d) JQuery
		e) Ajax
		f) Bootstrap
		g) NETBEANS 8.1
	Base de Datos	a) SQL SERVER 2012
	Servidor web	a) Glassfish

Tabla 5. 1 Tabla tecnologías escogidas . Fuente: Elaboración

5.6. Metodología de desarrollo de software empleada

Para este desarrollo de software se tomó la metodología RUP la cual fue escogida porque provee un enfoque estructurado para realizar tareas y responsabilidades en una organización de desarrollo. Su principal objetivo es asegurar la producción de software de alta calidad, que cumpla las necesidades de sus usuarios finales, que sea realizado en las fechas acordadas y con el presupuesto disponible.



5.7. Desarrollo de la propuesta.

En este punto realizaremos el desarrollo de nuestra investigación, basada en la metodología RUP.

5.7.1. Aplicación de prueba de prendas de vestir

5.7.1.1. Etapa N° I: “Modelado del Negocio”.

Objetivo:

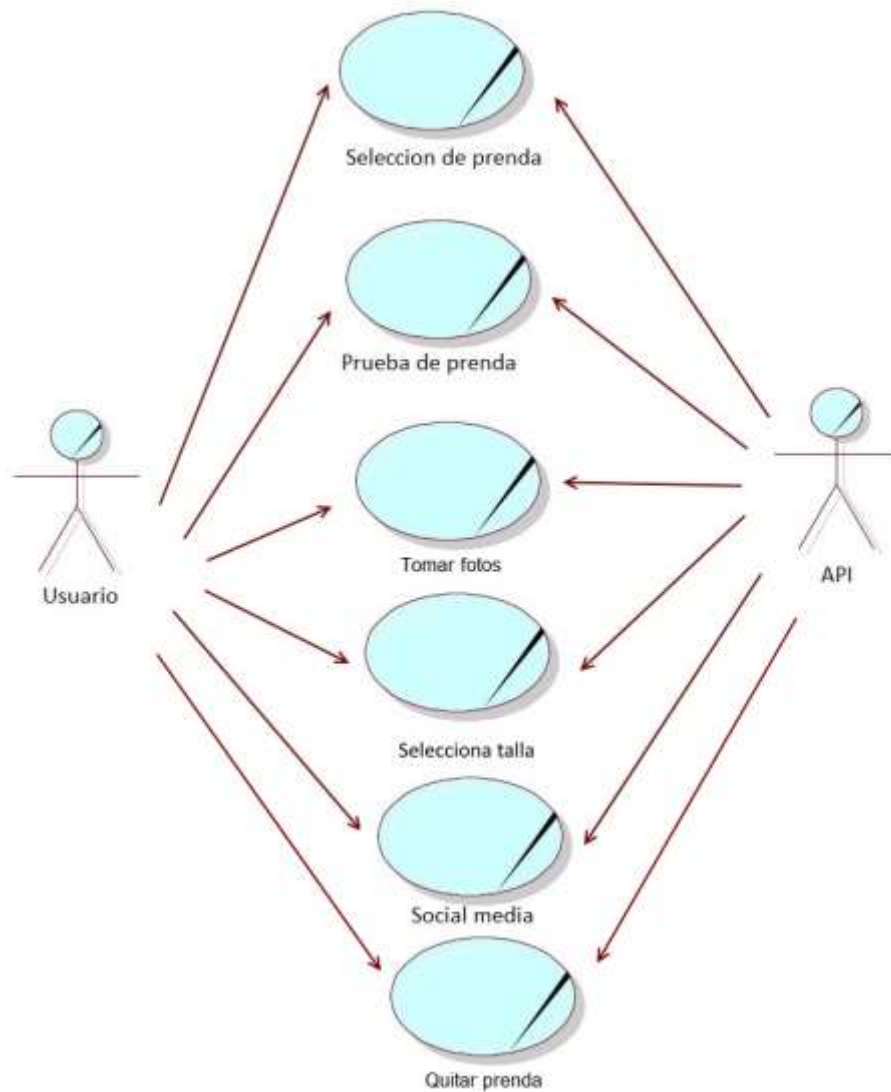
Comprender los procesos del Negocio de la Empresa, con el fin de obtener información relevante y así poder llevar a cabo la implementación del Sistema de Información.

A continuación, mostramos el desarrollo de las actividades que realizamos en esta etapa:

- **Modelo de Casos de Uso del Negocio.** - Representan los modelos genéricos de un sistema, es el primero en realizarse.
- **Modelo de Objetos del Negocio.** - Es aquella que va a plasmar el contexto de alguna de las actividades.
- **Modelo del Dominio del Problema.** - Representa la visión lógica del sistema, relaciona todas las entidades del modelo anterior.

Modelo de Caso de Uso de Negocio (MCUN):

Figura 5. 1: Diagrama de Caso de Uso de Negocio De API prueba de prenda de vestir Fuente: Elaboración propia



Specificaciones De Casos De Uso De Negocio:



Tabla 5. 2: Casos de uso de prueba de prenda de vestir **Fuente:** Elaboración propia

Nombre	Realizar prueba
Descripción	En este caso de uso empieza con el comprador al realizar una selección de la prenda de vestir y termina en la entrega de la prenda
Actores del Negocio	Intervienen en este proceso el COMPRADOR
Entradas	selección de prenda
Entregables	Se entrega la prenda al comprador
Prueba	Realiza las pruebas necesarias
Ajustar	Ajustar tamaño de la prenda
Fotos	Toma foto al usuario con la prenda
Quitar	Quita la prenda puesta
Talla	Seleccionar talla de prenda
Social media	Seleccionar la red social
Categoría	Caso de uso principal



**Modelo de Objeto de Negocio (MON): Gestión
Seleccionar prenda**

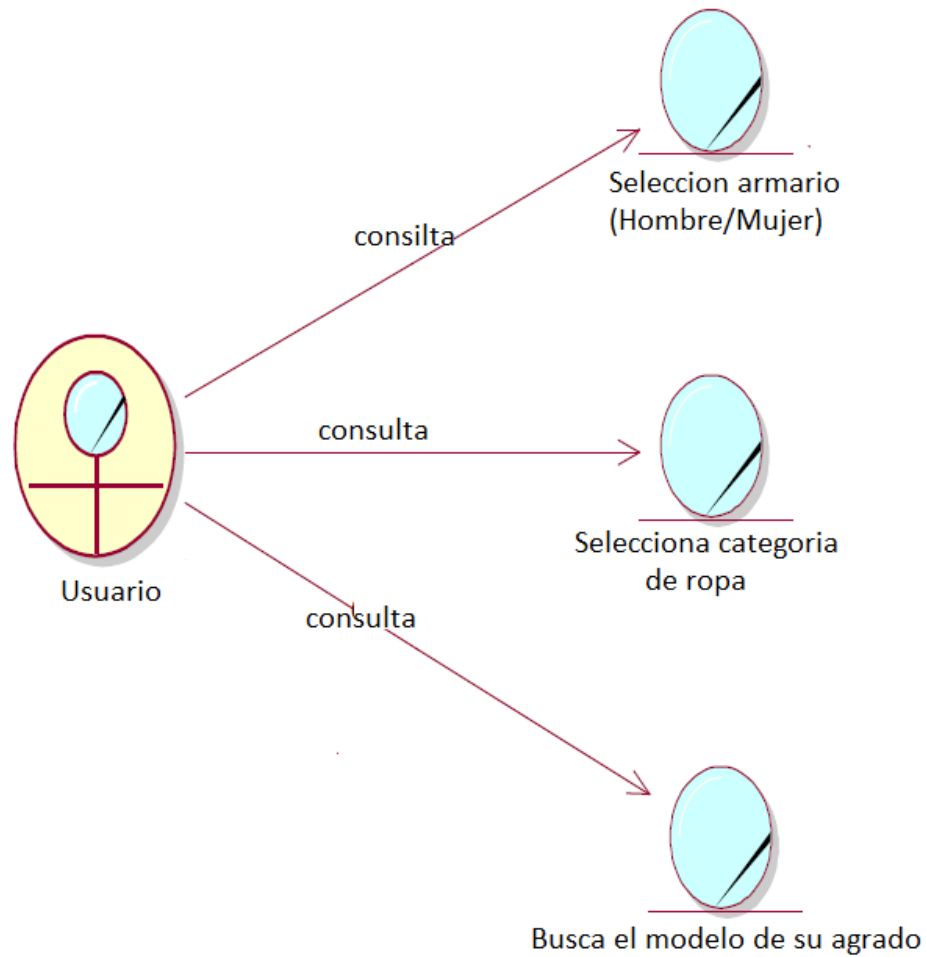


Figura 5.2 Modelo de Objeto de Negocio seleccionar prenda de vestir **Fuente:**
Elaboración propia



Modelo de Objeto de Negocio (MON): Gestión Prueba de prenda

Este MON, tiene como actor principal al cliente, quien participa en el proceso de prueba de prenda de vestir.

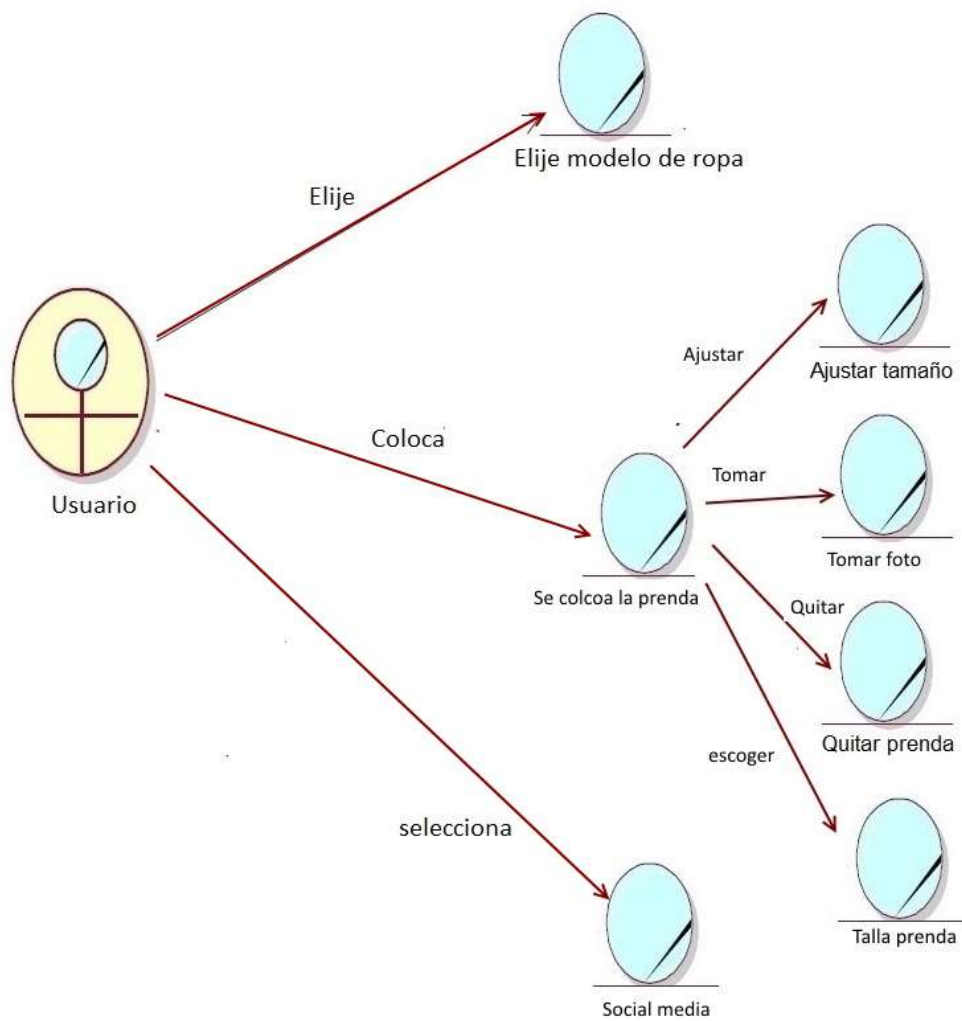


Figura 5. 3 Prueba de prenda de vestir **Fuente:** Elaboración propia



Modelo de Dominio Del Problema

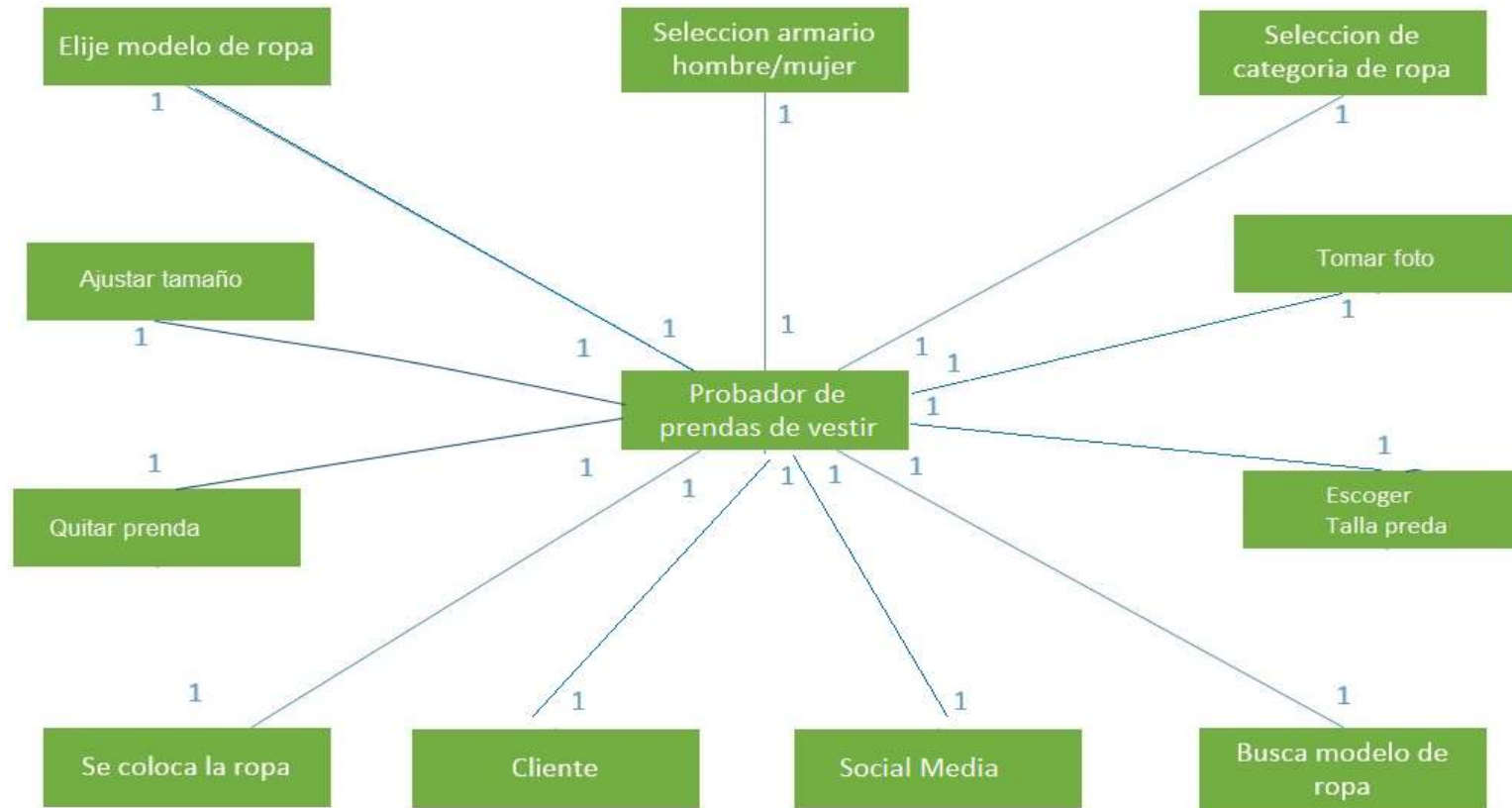


Figura 5. 4 Diagrama Del Modelo De Dominio del Problema prueba de prendas de vestir Fuente: Elaboración propi

5.7.1.2. Etapa N° II: “Determinación de los Requerimientos del Software.”

Objetivo:

Obtener las características que describen el comportamiento final del sistema que el usuario espera que realice, esto nos va a permitir obtener los requerimientos funcionales y no funcionales del sistema

A continuación mostramos los productos finales obtenidos los cuales son:

➤ **Modelo de Casos de Uso o Diagramas de Casos de**

Uso:

Captura todos los requerimientos funcionales y no funcionales del sistema.

Diagrama De Case Use: Gestión Seleccionar prenda

Caso de uso Seleccionar Prenda

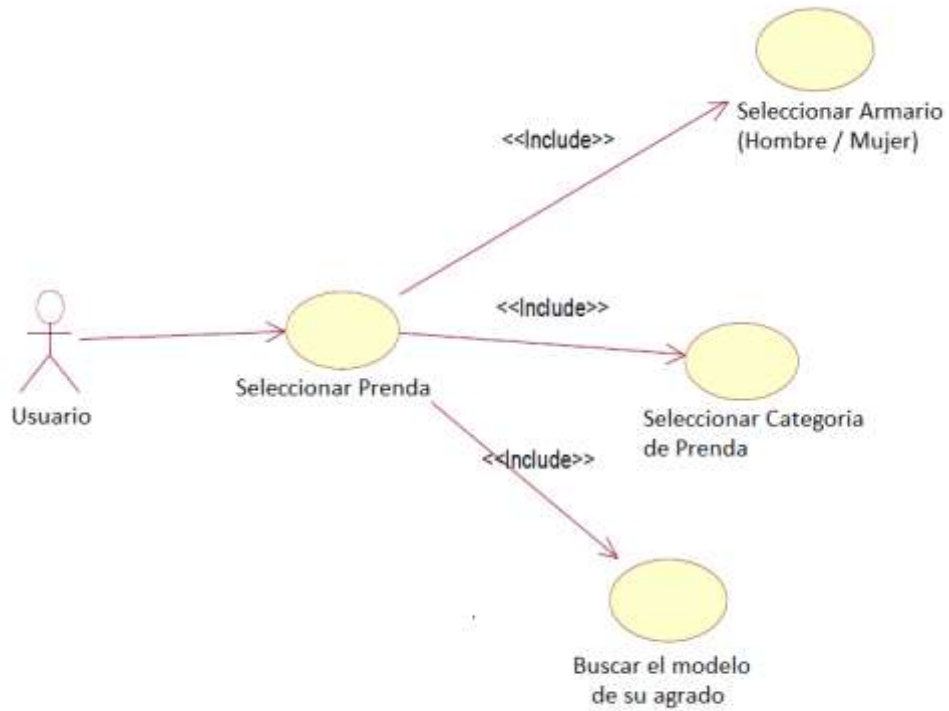


Figura 5. 5: Modelo De Caso De Uso de Gestionar Selección de prendas Fuente:
 Elaboración propia



Diagrama De Case Use: Gestión Prueba de prenda de vestir

Caso de uso Probar prenda de vestir

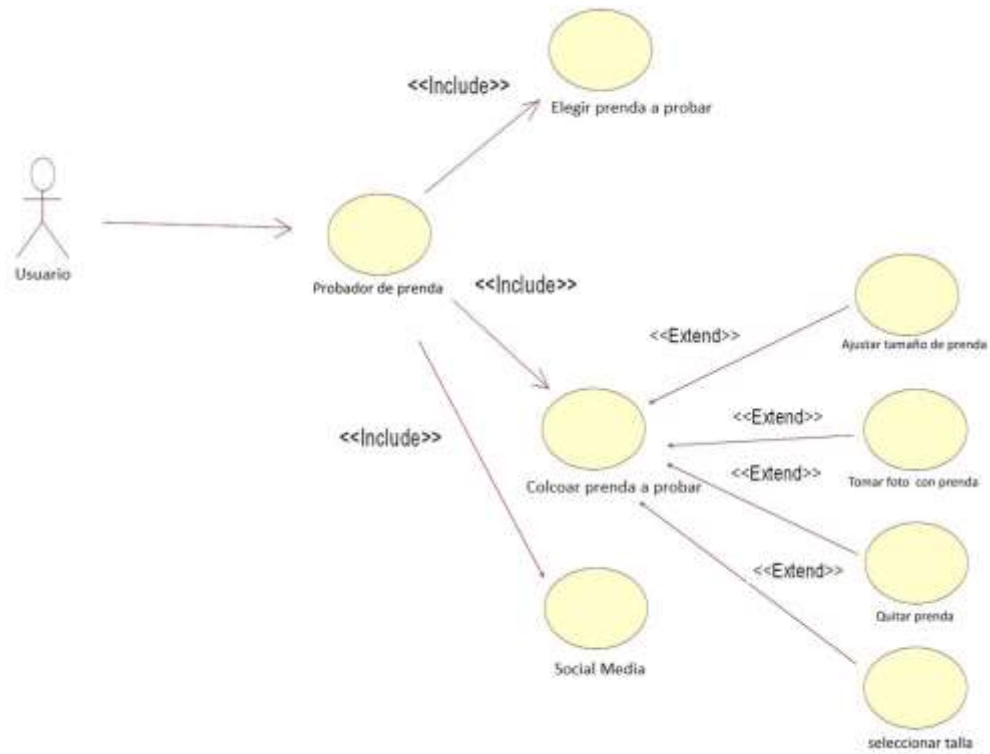


Figura 5. 6: Modelo De Caso De Uso De Prueba de prenda de vestir Fuente: Elaboración propia



5.7.1.3. Etapa N° III: “Análisis del Software”

Objetivo:

Representar el comportamiento del sistema con base en los conceptos del dominio del problema sin entrar en detalles de la solución.

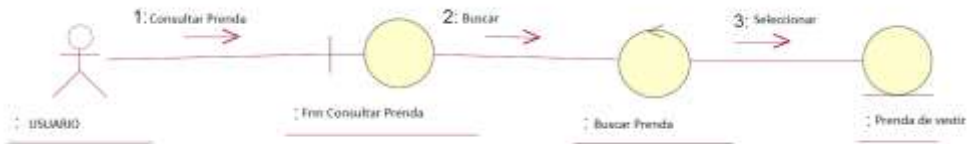
A continuación mostramos los productos finales obtenidos los cuales son:

- **Diagrama de Clases del Análisis.-** Rompe la relación de muchos a muchos del dominio del problema y se agrega atributos.
- **Diagrama de Secuencia.-** Igual que el diagrama de colaboraciones, pero diferente en cuanto a su estructura.
- **Diagrama de Colaboraciones.-** Modela los objetos y los enlaces implicados en la implementación de una interacción (involucra recibir información y responder a usuarios y sistemas externos.).

DIAGRAMA DE COLABORACION:

A. GESTIONAR SELECCIÓN DE PRENDA

Figura 5. 7: Diagrama de Colaboración seleccionar prenda de vestir **Fuente:** Elaboración propia



B. GESTIONAR PRUEBA DE PRENDA

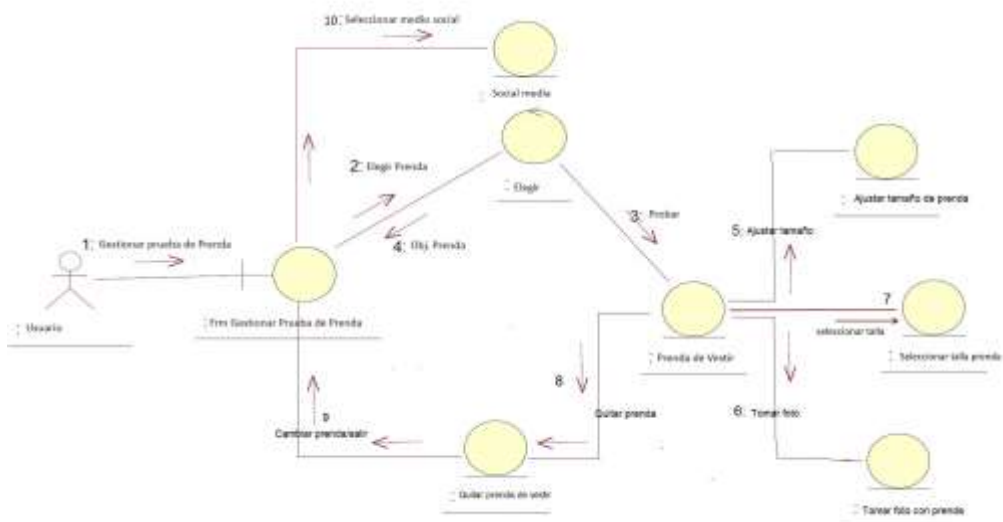


Figura 5. 8: Diagrama de Colaboración Probar prenda de vestir **Fuente:** Elaboración propia



DIAGRAMA DE SECUENCIA:

A. GESTIONAR SELECCIÓN DE PRENDA:

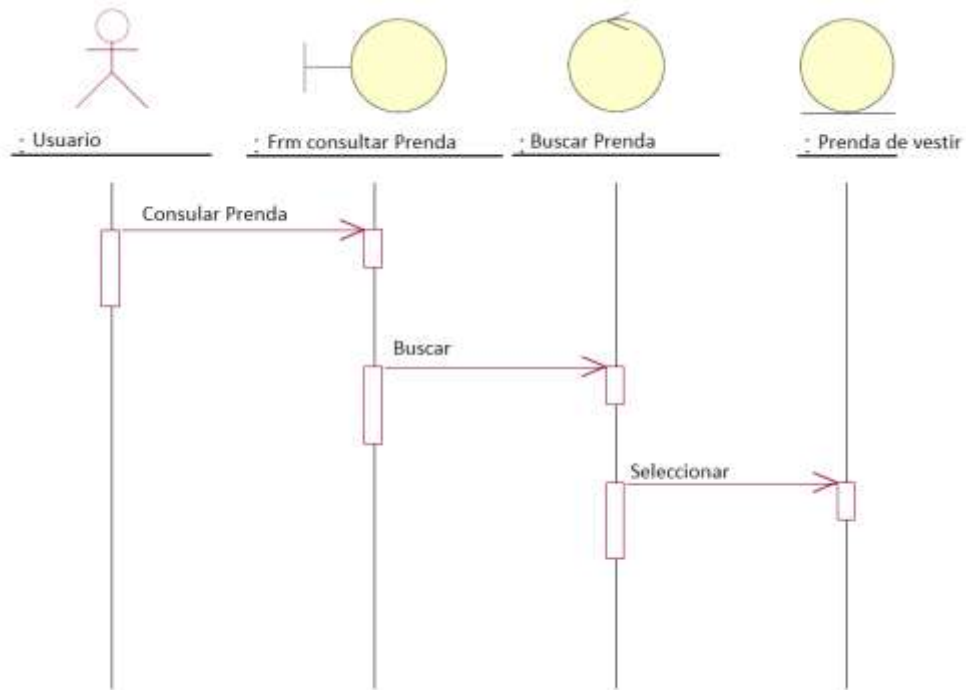


Figura 5. 9: Diagrama de secuencia seleccionar prenda de vestir Fuente: Elaboración propia



B. GESTIONAR SELECCIÓN DE PRENDA:

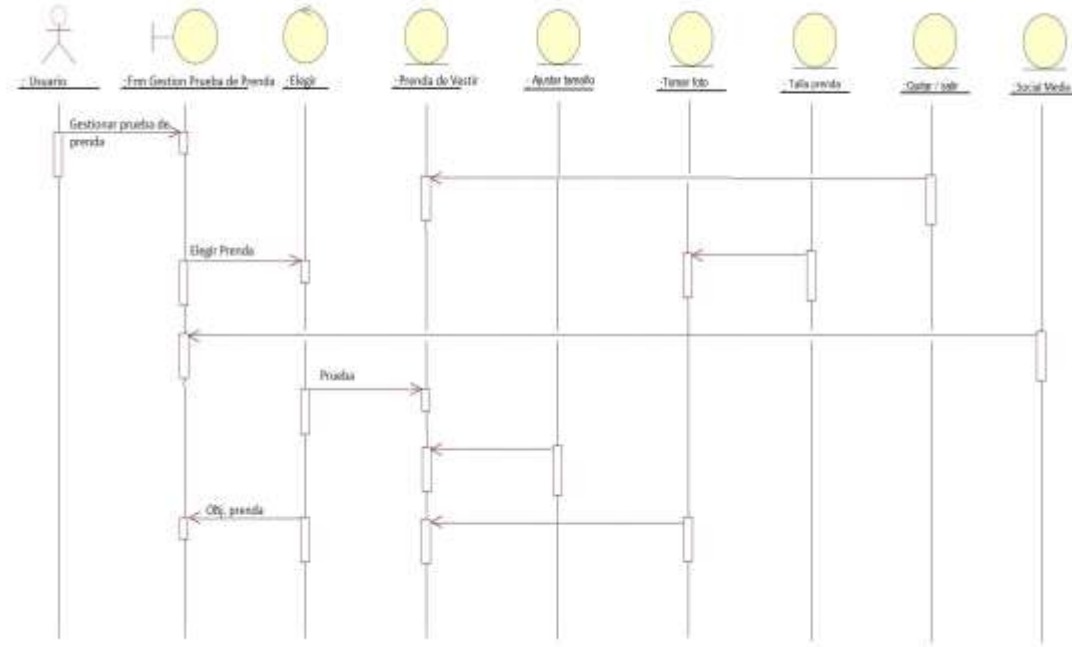


Figura 5. 10: Diagrama de secuencia seleccionar prenda de vestir **Fuente:** Elaboración propia



DIAGRAMA DE ARQUITECTURA DE LA APLICACIÓN

Figura 5. 12: Diagrama de Colaboración seleccionar prenda de vestir Fuente: Elaboración propia

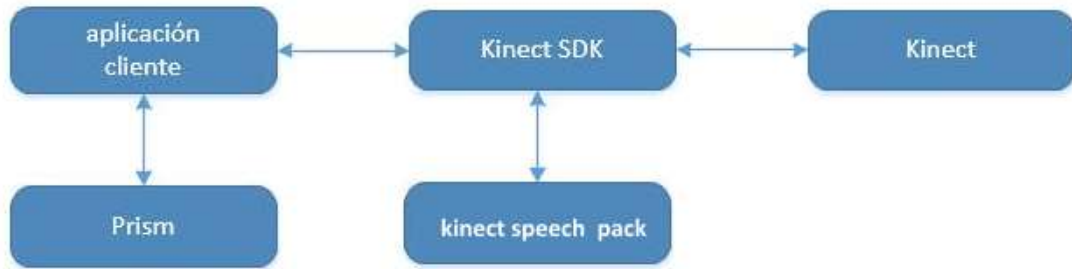


DIAGRAMA DE PATRON DE DISEÑO MVVM – APLICACION

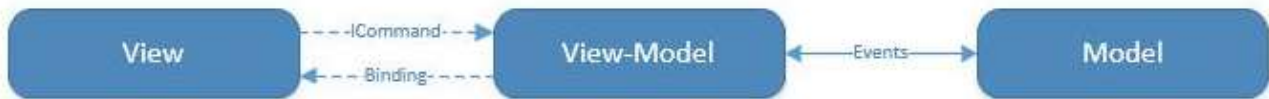


Figura 5. 13: Diagrama de patrón de diseño Fuente: Elaboración propia

5.7.1.4. Etapa N° IV: “Diseño de sistemas”

Objetivo:

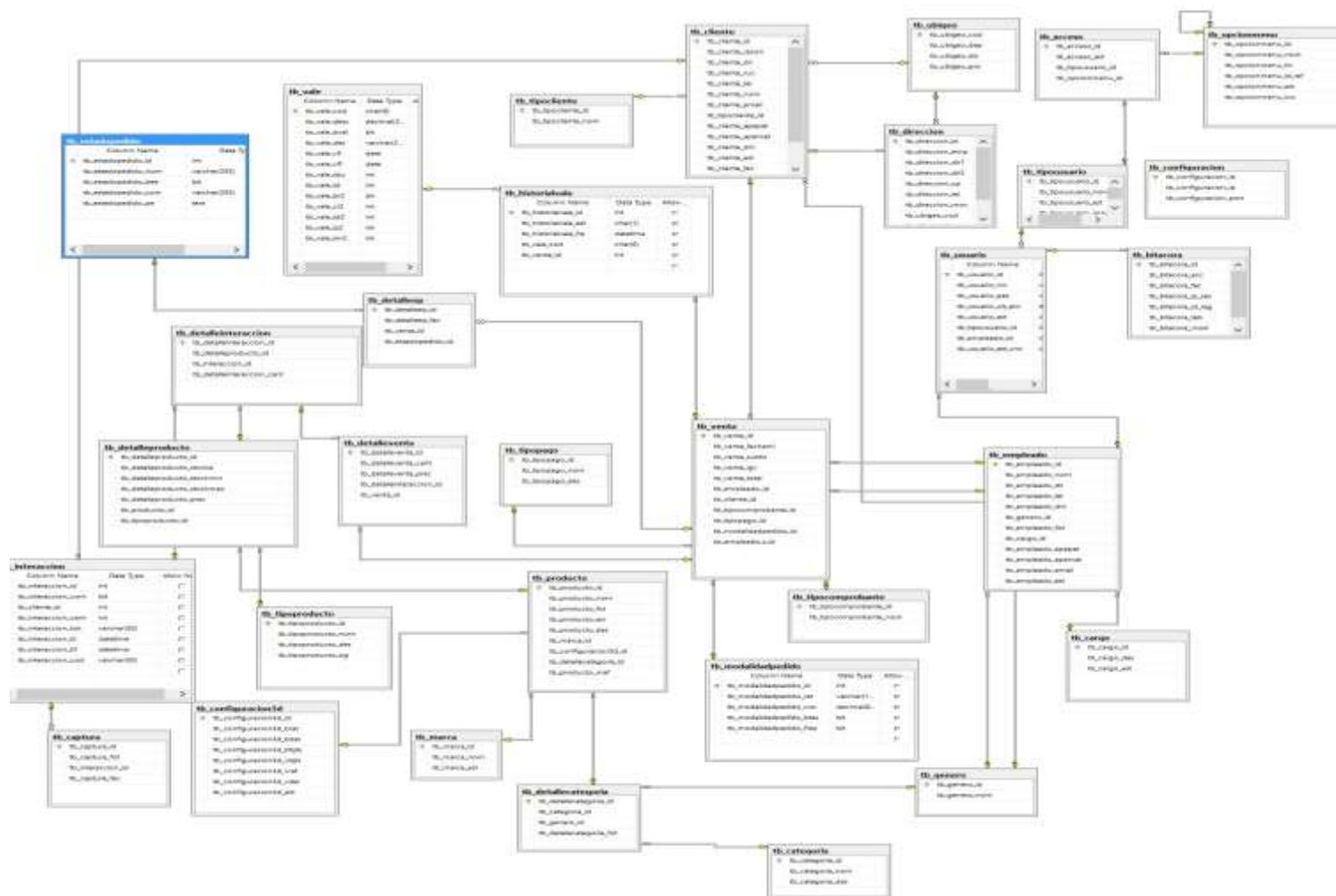
Representar los componentes con los cuales se va a construir el software.

A continuación mostramos los productos finales obtenidos los cuales son:

- **Modelo de Datos (Modelo Relacional en SQL).-**
Modela la base de datos, diseñada en un Gestor de base de datos.
- **Interface Gráfica de Usuarios (GUI).-** Presenta las interfaces del sistema, diseñada en un Lenguaje de Programación.

MODELO DE DATOS

Figura 5. 14: Modelo de datos Fuente: Elaboración propi



INTERFACES DE LA PROPUESTA DE LA APLICACIÓN

Figura 5. 15: Interfaz de Mujer Fuente: Elaboración propia



Figura 5. 16: Interfaz de Hombre Fuente: Elaboración propia

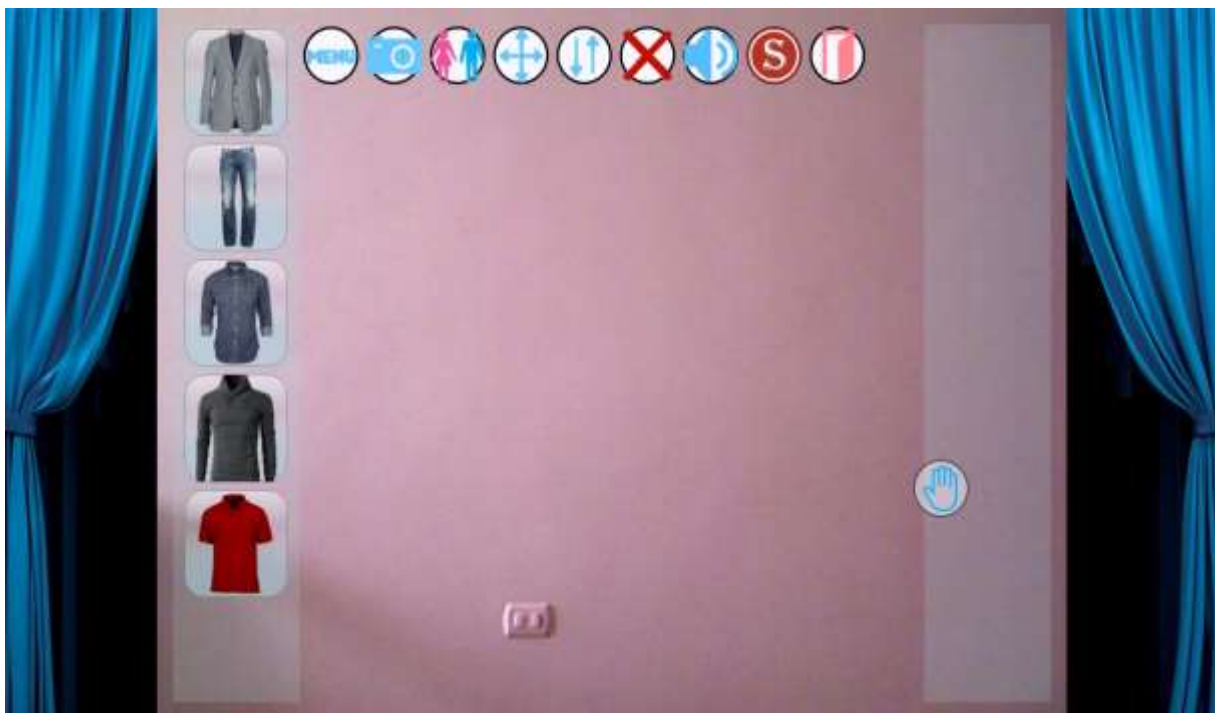


Figura 5. 17: Interfaz interacción prueba Fuente: Elaboración propia



Figura 5. 18: Interfaz de tomar fotos Fuente: Elaboración propia

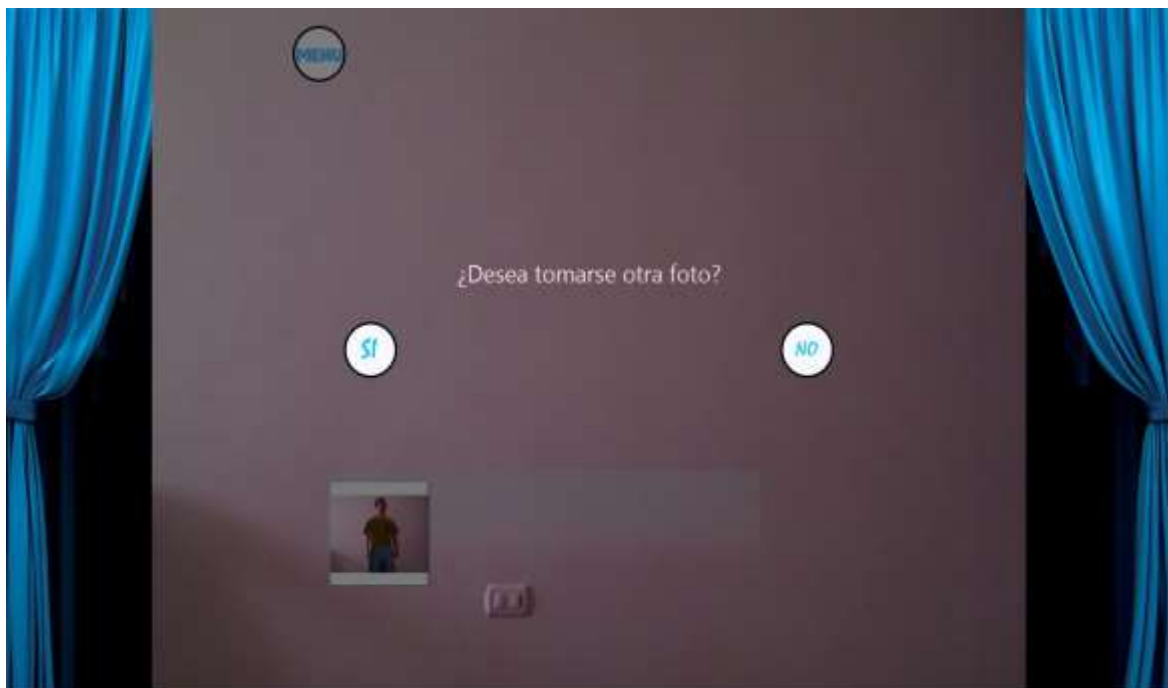


Figura 5. 19: Interfaz de seleccionar talla Fuente: Elaboración propia



Figura 5. 20: Interfaz QR Fuente: Elaboración propia



Figura 5. 21: Interfaz Social Media Fuente: Elaboración propia

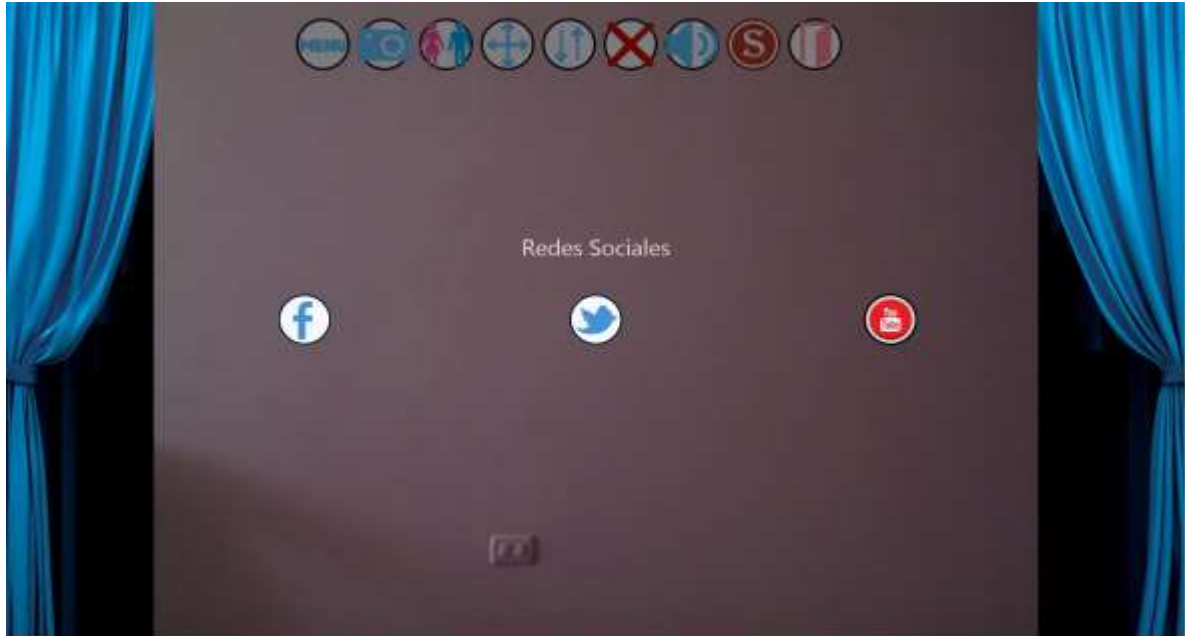


Figura 5. 22: Interfaz de ajuste de prenda 1 Fuente: Elaboración propia

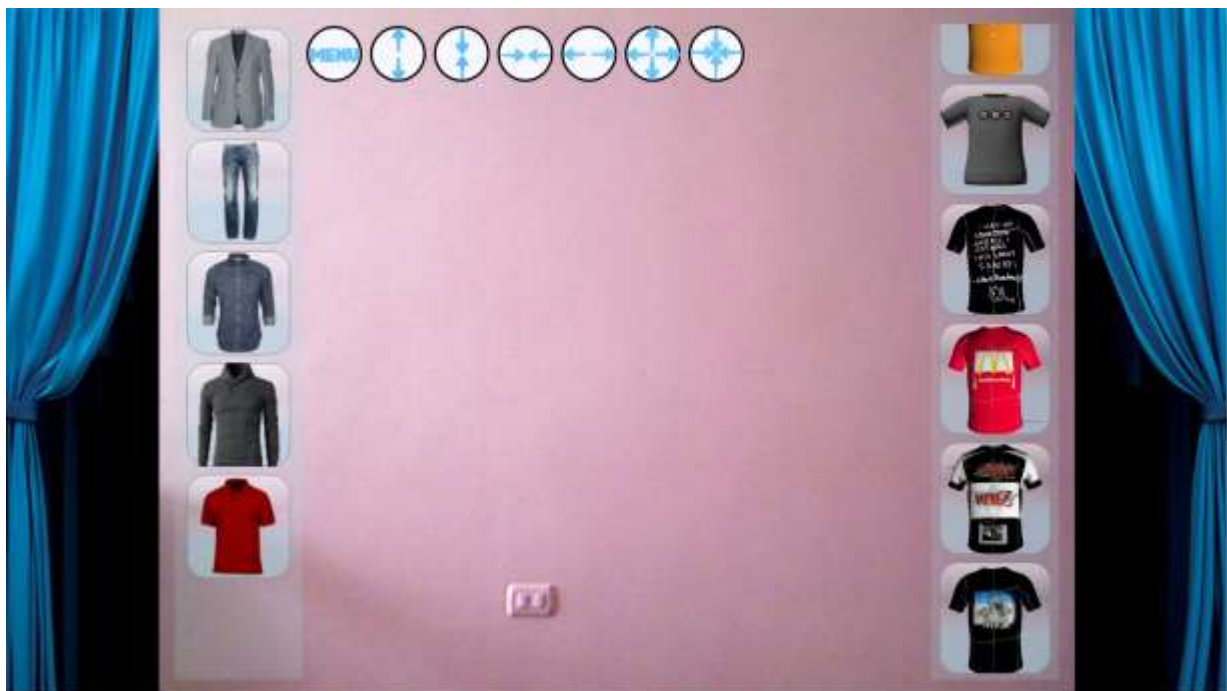
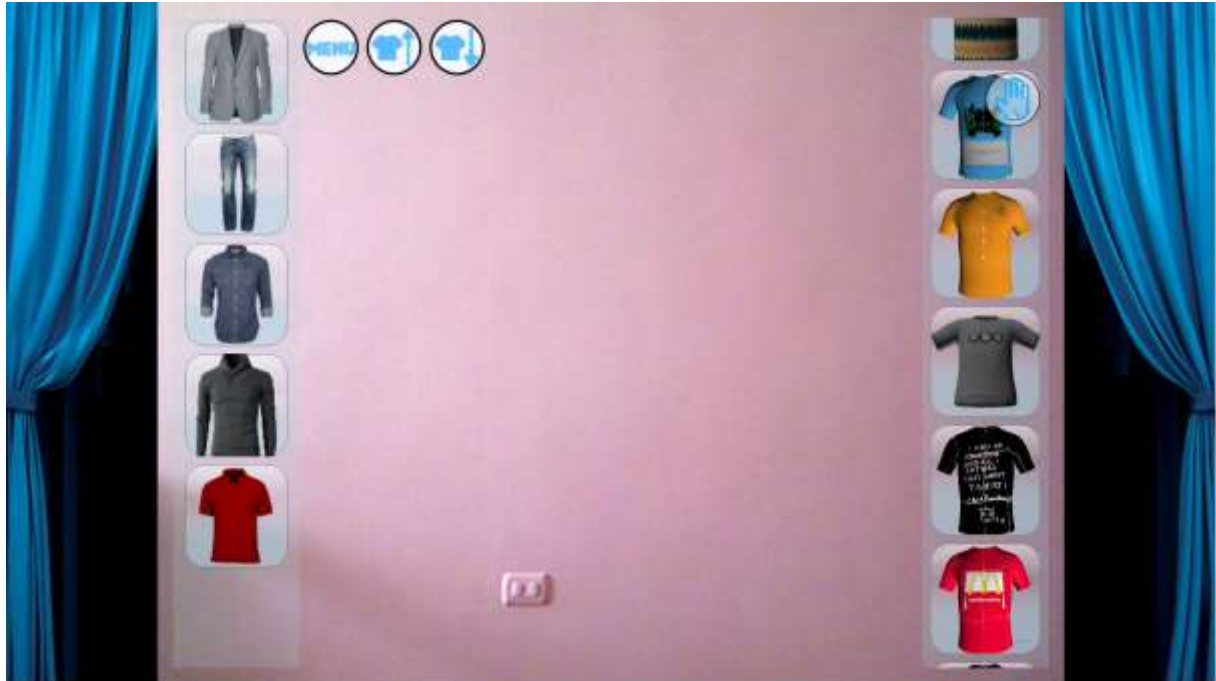


Figura 5. 23: Interfaz ajuste de prenda 2 Fuente: Elaboración propia



5.7.1.5. Etapa N° V: “Implementación del Sistema”

Objetivo:

Describir el código producido, los archivos generados (fuente, intermedios, ejecutables) y adquiridos (bibliotecas de funciones) y las herramientas y procedimientos requeridos para obtener, instalar y poner en funcionamiento los ejecutables del sistema.

A continuación mostramos los productos finales obtenidos los cuales son:

- **Modelo de Componentes.-** Describe los archivos que integran el sistema y las relaciones de dependencia que existen entre ellos.

DIAGRAMA DE COMPONENTES ESQUEMA GENERAL

Figura 5. 24: Diagrama de componentes de esquema general **Fuente:** Elaboración propia

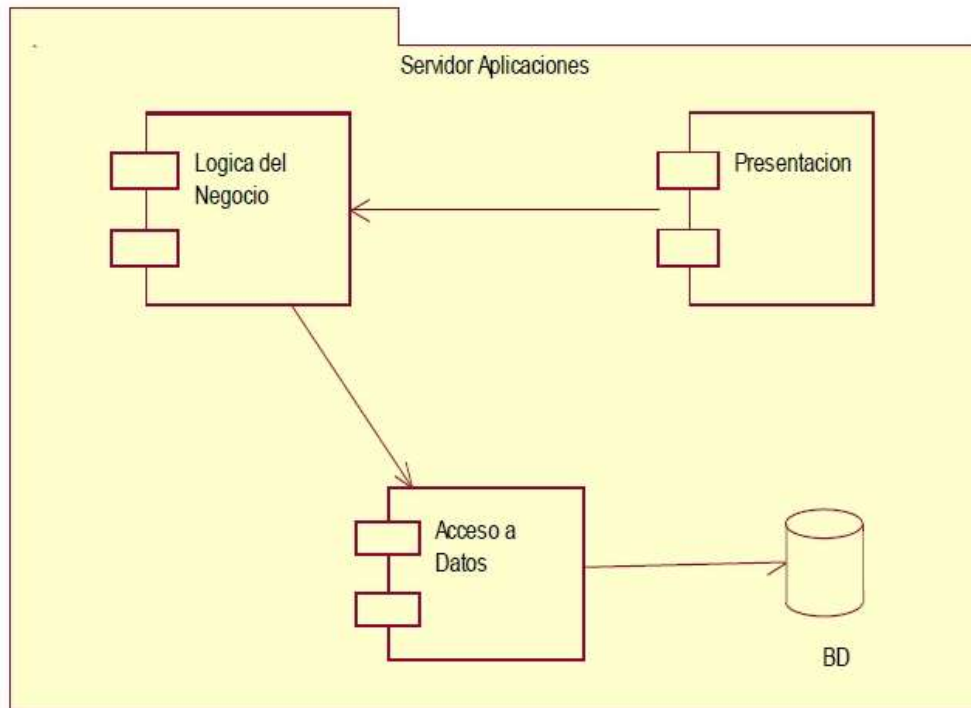


DIAGRAMA DE COMPONENTES ESQUEMA GENERAL

Figura 5. 25: Diagrama de componentes de esquema general **Fuente:** Elaboración propia

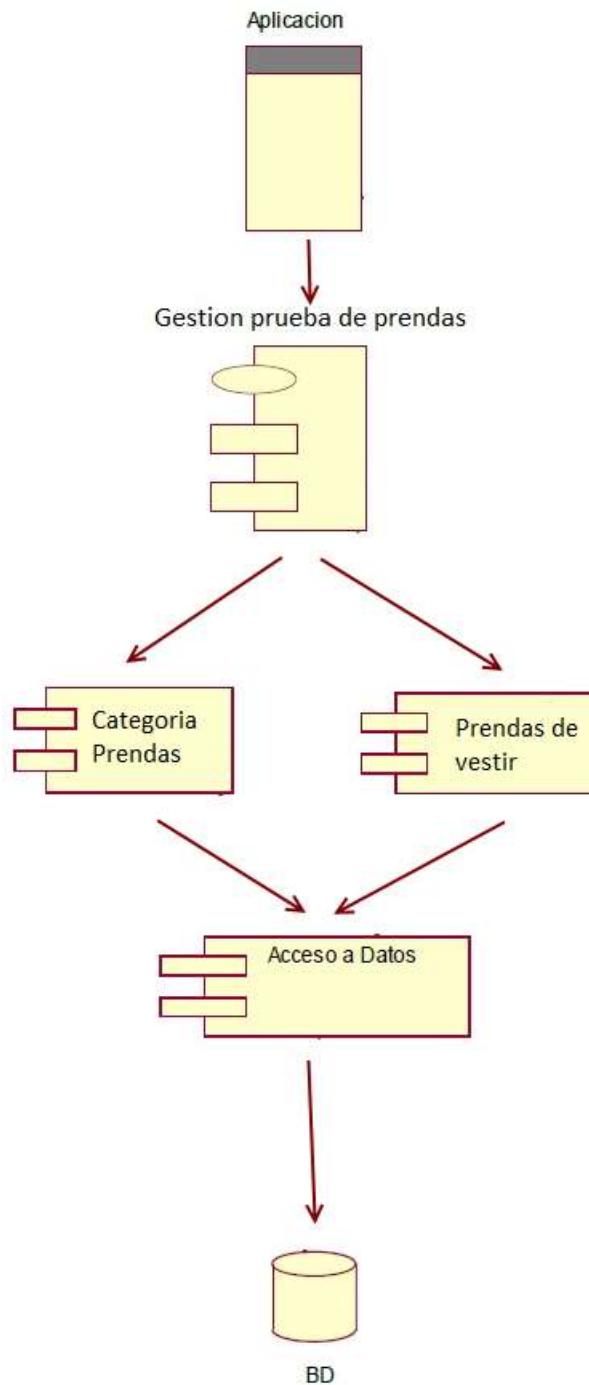


DIAGRAMA DE ARQUITECTURA DE COMPONENTES

Figura 5. 26: Diagrama de arquitectura de componentes Fuente: Elaboración propia

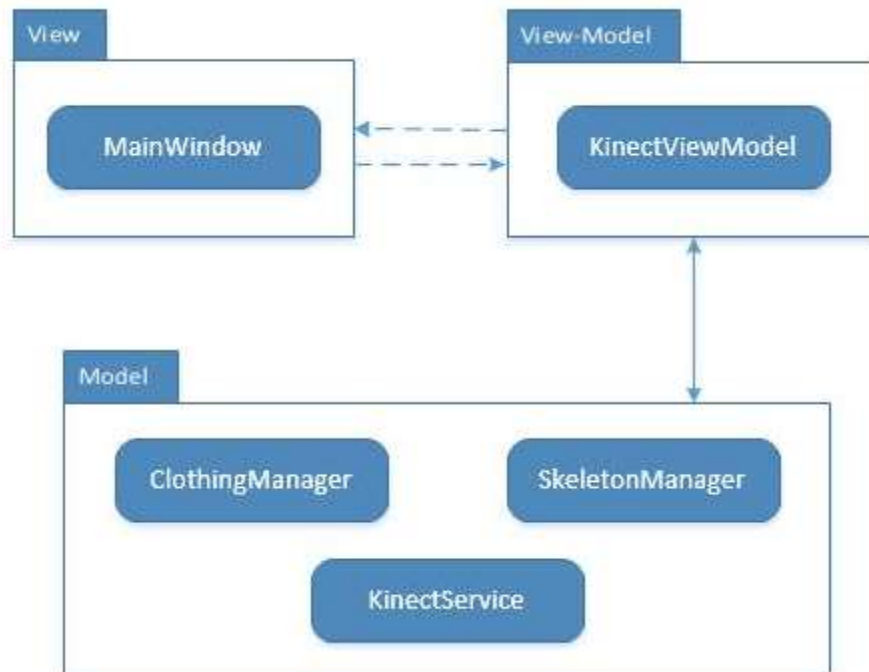
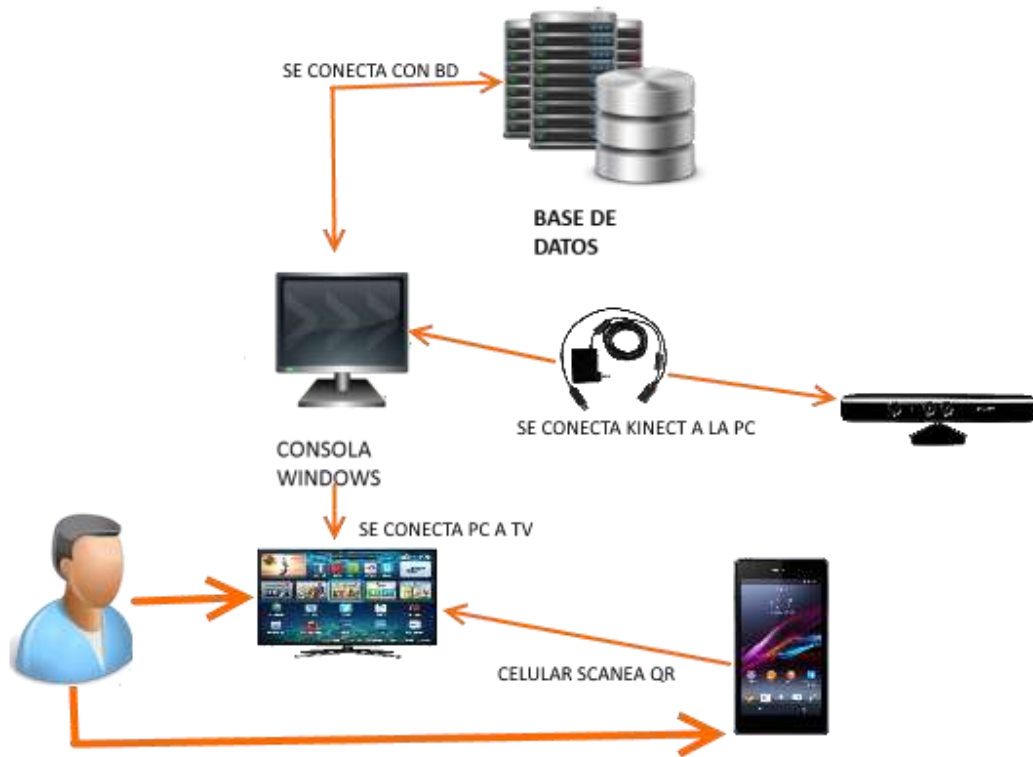


DIAGRAMA DE ARQUITECTURA DE HARDWARE

Figura 5. 27: Diagrama Componentes de Hardware Fuente: Elaboración propia



5.7.2. Aplicación de web

5.7.2.1. Etapa Nº I: “Modelado del Negocio”.

Objetivo:

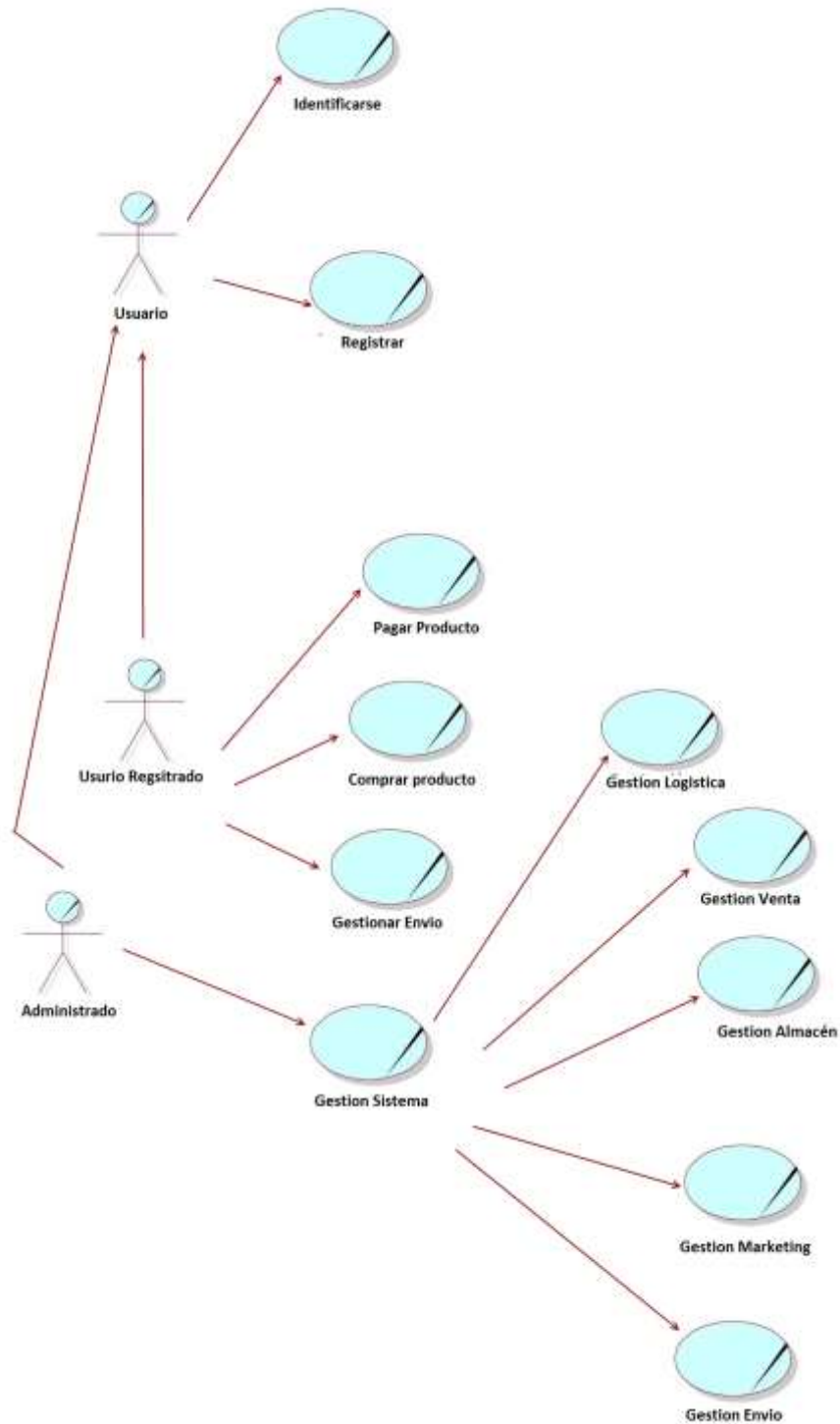
Comprender los procesos del Negocio de la Empresa, con el fin de obtener información relevante y así poder llevar a cabo la implementación del Sistema de Información.

A continuación, mostramos el desarrollo de las actividades que realizamos en esta etapa:

- **Modelo de Casos de Uso del Negocio.** - Representan los modelos genéricos de un sistema, es el primero en realizarse.
- **Modelo de Objetos del Negocio.** - Es aquella que va a plasmar el contexto de alguna de las actividades.
- **Modelo del Dominio del Problema.** - Representa la visión lógica del sistema, relaciona todas las entidades del modelo anterior.

Modelo de Caso de Uso de Negocio (MCUN):

Figura 5. 28 Diagrama de Caso de Uso de Negocio De Web Fuente: Elaboración propia



Especificaciones De Casos De Uso De Negocio:

Tabla 5. 3: Casos de uso de venta de prendas de vestir **Fuente:** Elaboración propia

Nombre	Realizar prueba
Descripción	En este caso de uso empieza con el comprador al realizar una selección de la prenda de vestir y termina en la entrega de la prenda
Actores del Negocio	Intervienen en este proceso el COMPRADOR
Entradas	selección de prenda
Entregables	Se entrega la prenda al comprador
Identifica	El usuario inicia sesión
Registrar	El usuario se registra
Pagar producto	El usuario paga el producto
Compra productos	El usuario compra producto
Envío	Usuario gestiona envío
Gestión logística	Administrador revisa producto
Gestión producto	Almacén verifica producto
Gestión usuario	Administrador gestiona usuario
Gestión marketing	Administra ofertas
Gestión envío	Almacén gestiona envío del producto al usuario
Gestión Venta	Vendedor gestiona comprobante de venta



Modelo de Objeto de Negocio (MON): Vender Productos

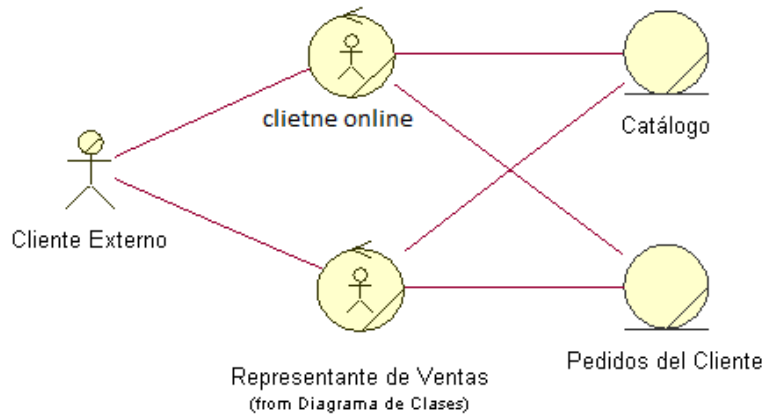


Figura 5. 29 Modelo de Objeto de Negocio vender producto **Fuente:** Elaboración propia

Modelo de Objeto de Negocio (MON): Reponer Stock

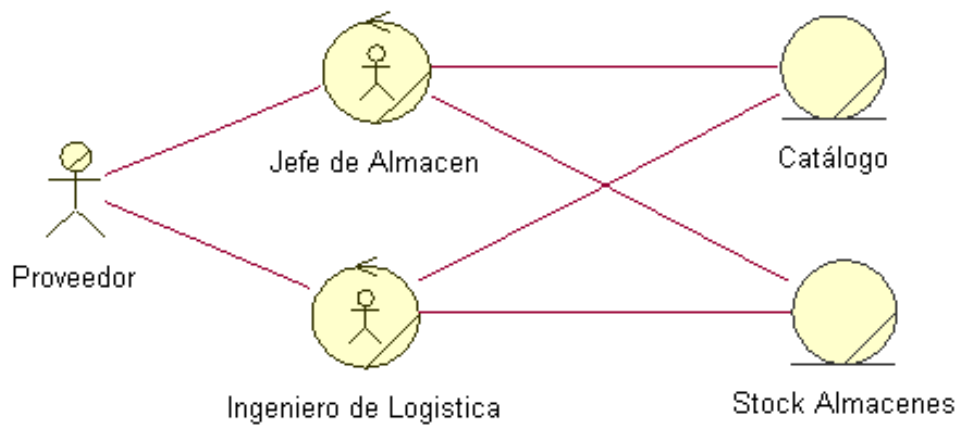


Figura 5. 30 Modelo de Objeto de Negocio reponer stock **Fuente:** Elaboración propia



Modelo de Objeto de Negocio (MON): Realizar Entrega

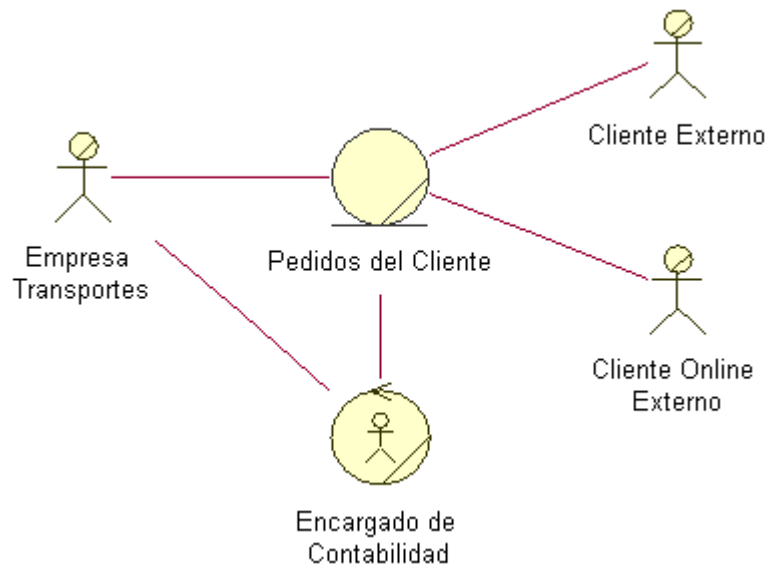


Figura 5. 31 Modelo de Objeto de Negocio realizar entrega **Fuente:** Elaboración propia

Modelo de caso de Negocio

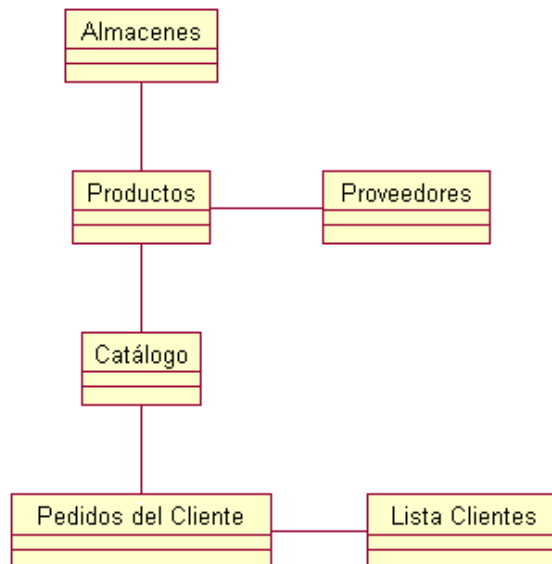


Figura 5. 32: Diagrama Del Modelo De Dominio del Problema ventas de prendas de vestir **Fuente:** Elaboración propi



5.7.2.2. Etapa N° II: “Determinación de los Requerimientos del Software.”

Objetivo:

Obtener las características que describen el comportamiento final del sistema que el usuario espera que realice, esto nos va a permitir obtener los requerimientos funcionales y no funcionales del sistema

A continuación mostramos los productos finales obtenidos los cuales son:

➤ **Modelo de Casos de Uso o Diagramas de Casos de**

Uso:

Captura todos los requerimientos funcionales y no funcionales del sistema.

Diagrama De Case Use: Gestión venta

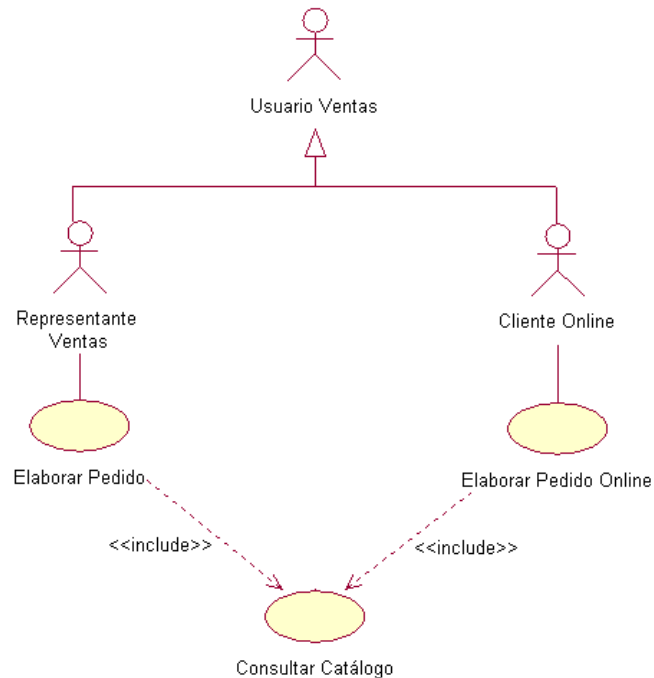


Figura 5. 33: Modelo De Caso De Uso de Gestionar venta **Fuente:** Elaboración propia

Diagrama De Case Use: Facturación

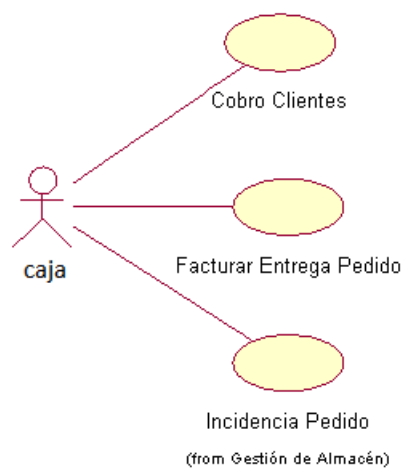


Figura 5. 34: Modelo De Caso De Uso facturación **Fuente:** Elaboración propia



Diagrama De Case Use: Gestión Almacén

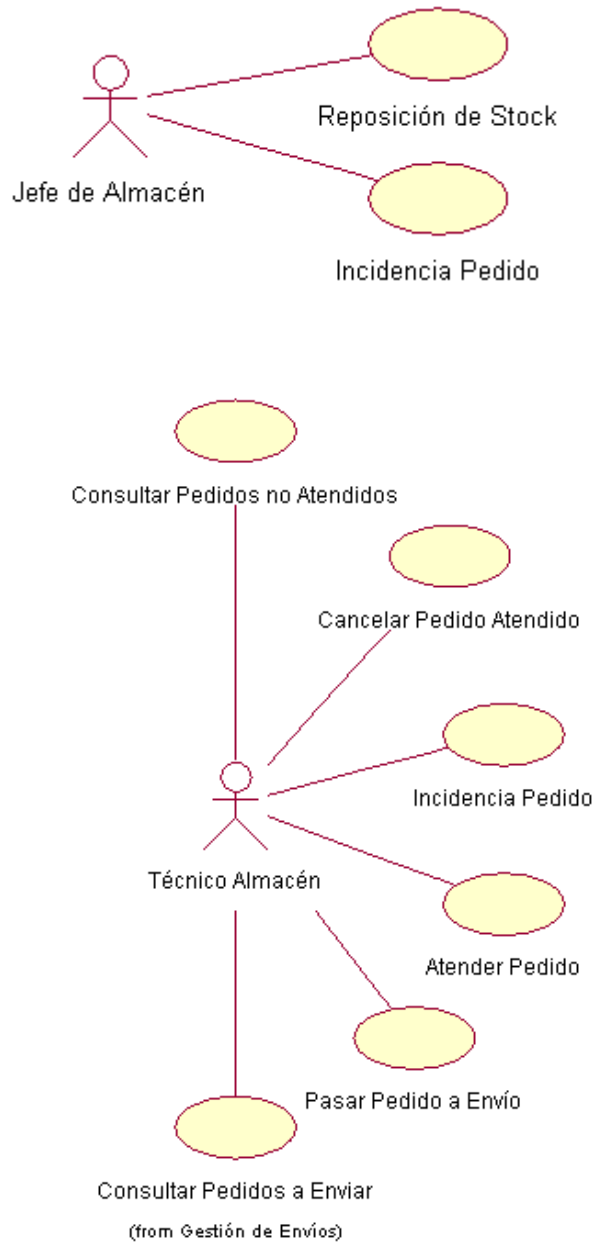


Figura 5. 35 Modelo De Caso De Uso de gestión almacén **Fuente:** Elaboración propia



Diagrama De Case Use: Gestión envío

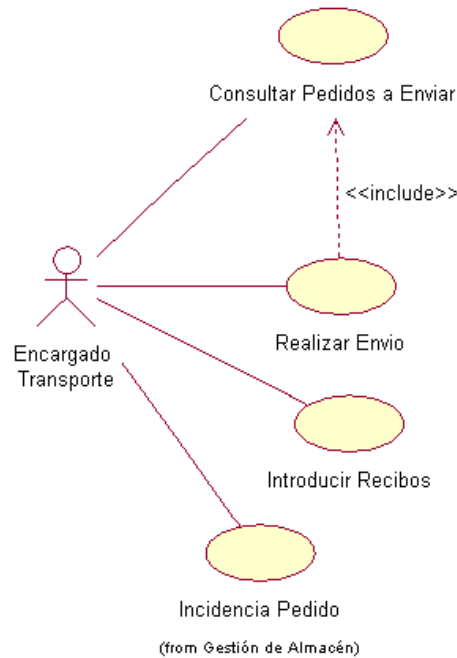


Figura 5. 36 : Modelo De Caso De Uso de gestión envío **Fuente:** Elaboración propia

Diagrama De Case Use: Gestión Logística

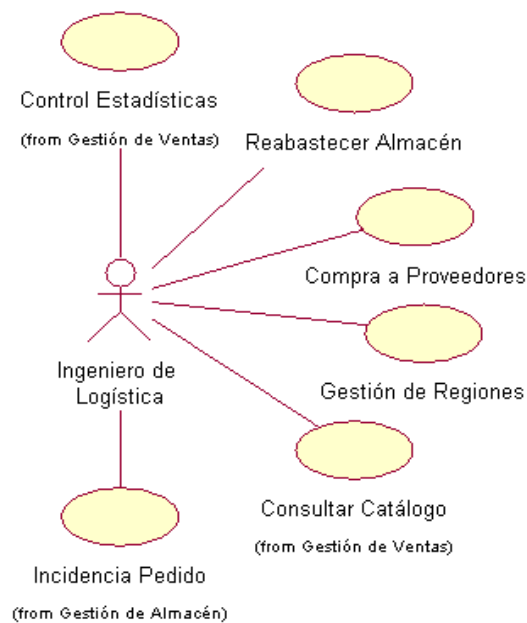


Figura 5. 37: Modelo De Caso De Uso de gestión logistica **Fuente:** Elaboración propia



Diagrama De Case Use: Gestión Marketing

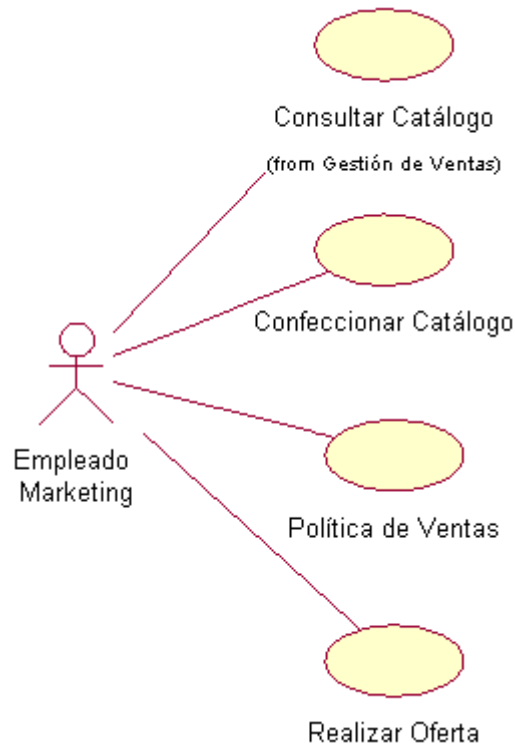


Figura 5. 38 Modelo De Caso De Uso de gestión marketing **Fuente:** Elaboración propia



5.7.2.3. Etapa N° III: “Análisis del Software”

Objetivo:

Representar el comportamiento del sistema con base en los conceptos del dominio del problema sin entrar en detalles de la solución.

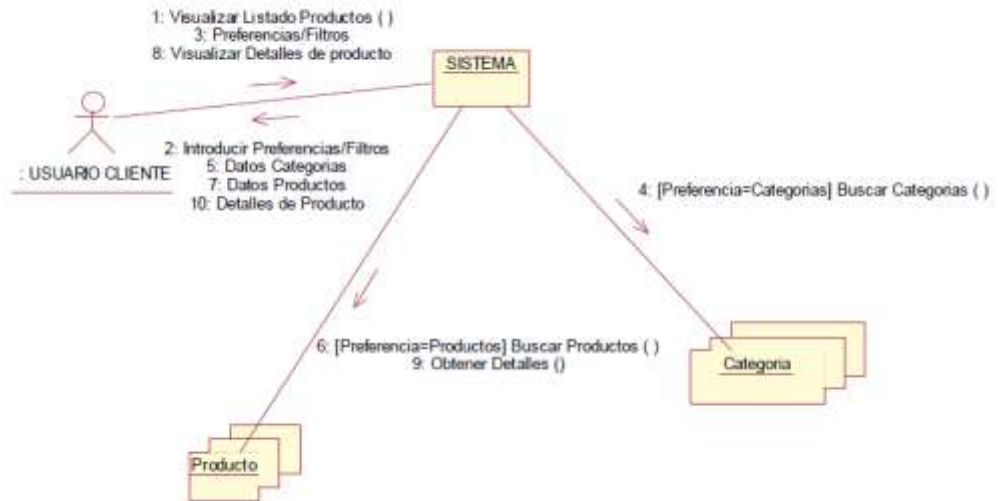
A continuación mostramos los productos finales obtenidos los cuales son:

- **Diagrama de Clases del Análisis.-** Rompe la relación de muchos a muchos del dominio del problema y se agrega atributos.
- **Diagrama de Secuencia.-** Igual que el diagrama de colaboraciones, pero diferente en cuanto a su estructura.
- **Diagrama de Colaboraciones.-** Modela los objetos y los enlaces implicados en la implementación de una interacción (involucra recibir información y responder a usuarios y sistemas externos.).

DIAGRAMA DE COLABORACION:

A. CONSULTAR PRODUCTO

Figura 5. 39 : Diagrama de Colaboración consultar producto **Fuente:** Elaboración propia



B. GESTIONAR DATOS DEL CLIENTE - VISUALIZAR

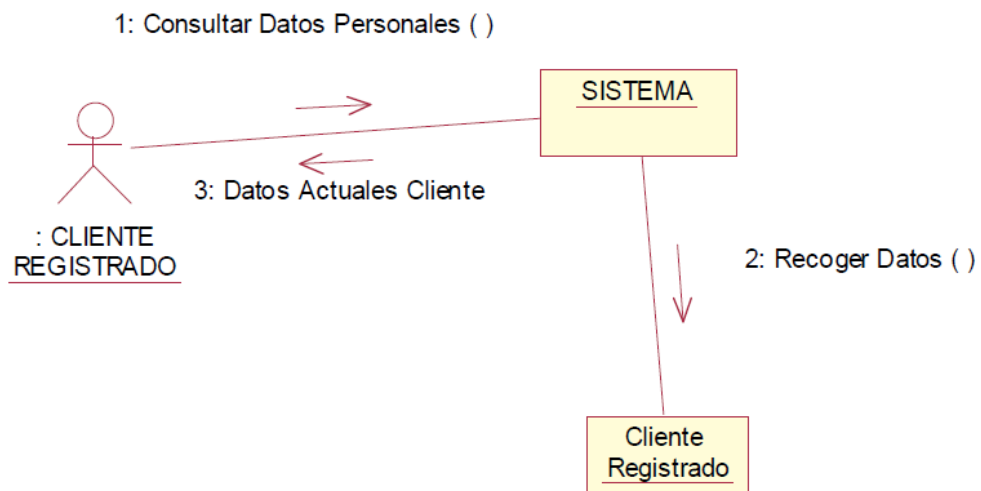


Figura 5. 40 : Diagrama de Colaboración datos del cliente - visualziar **Fuente:** Elaboración propia



C. GESTIONAR DATOS DEL CLIENTE - MODIFICAR

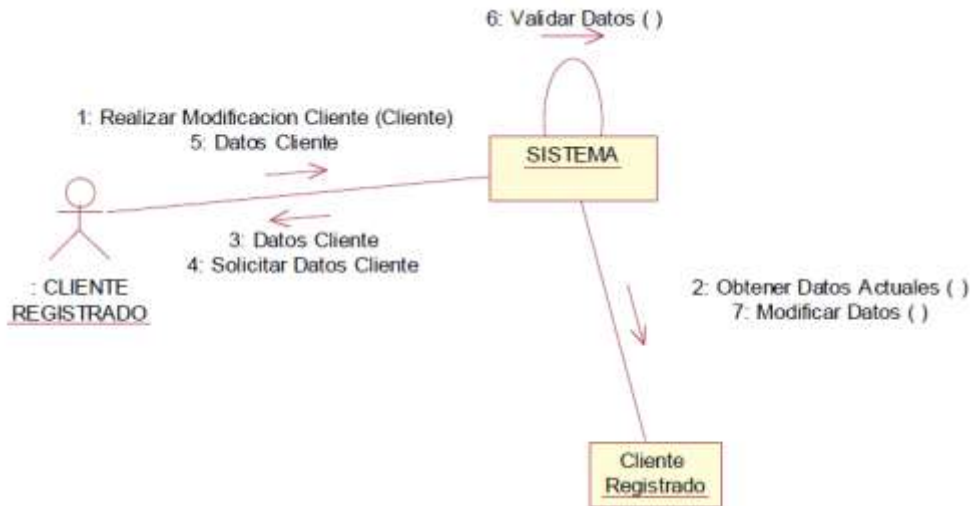


Figura 5. 41: Diagrama de Colaboración datos del cliente - modificar **Fuente:**
Elaboración propia

B. GESTIONAR DATOS DEL CLIENTE – REGISTRAR

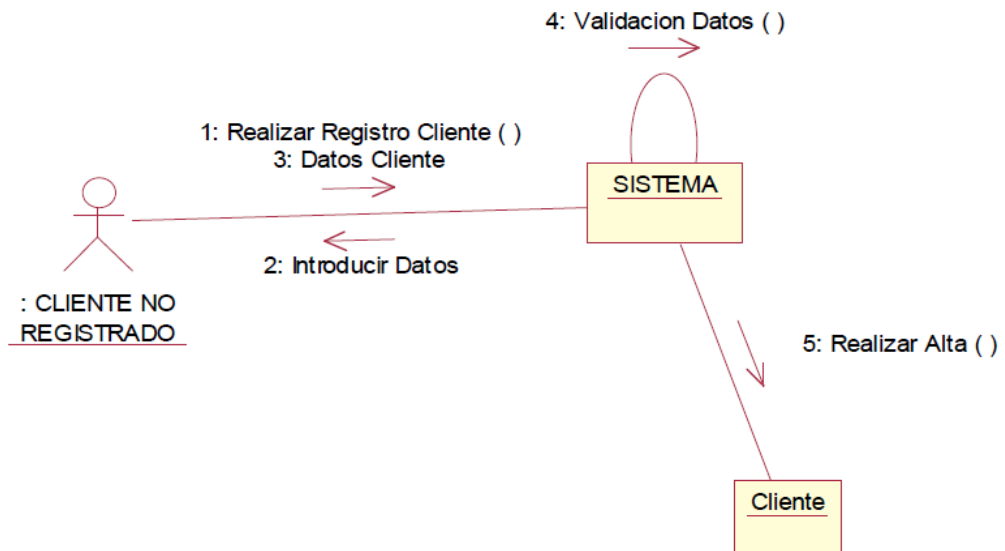


Figura 5. 42: Diagrama de Colaboración datos del cliente - registrar **Fuente:**
Elaboración propia



C. GESTIONAR COMPRA

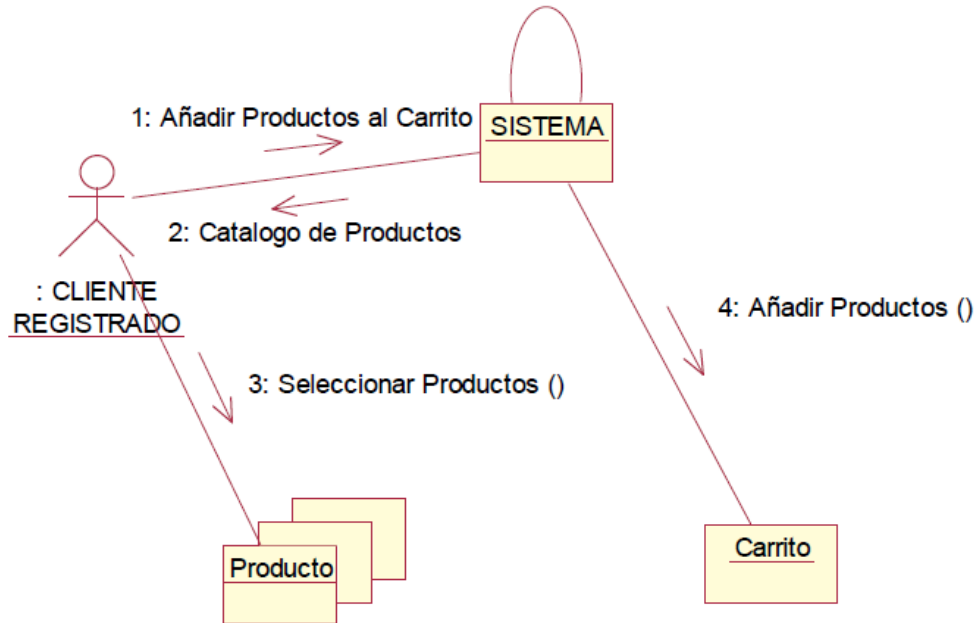
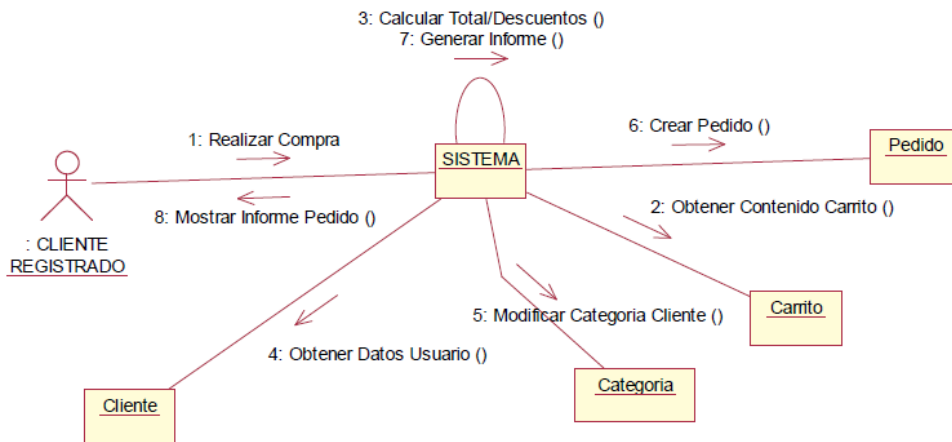


Figura 5. 43: Diagrama de Colaboración gestionar compra **Fuente:** Elaboración propia

D. GESTIONAR COMPRA – VISUALIZAR

Figura 5. 44: Diagrama de Colaboración compra - visualziar **Fuente:** Elaboración propia



E. GESTIONAR COMPRA - PEDIDO

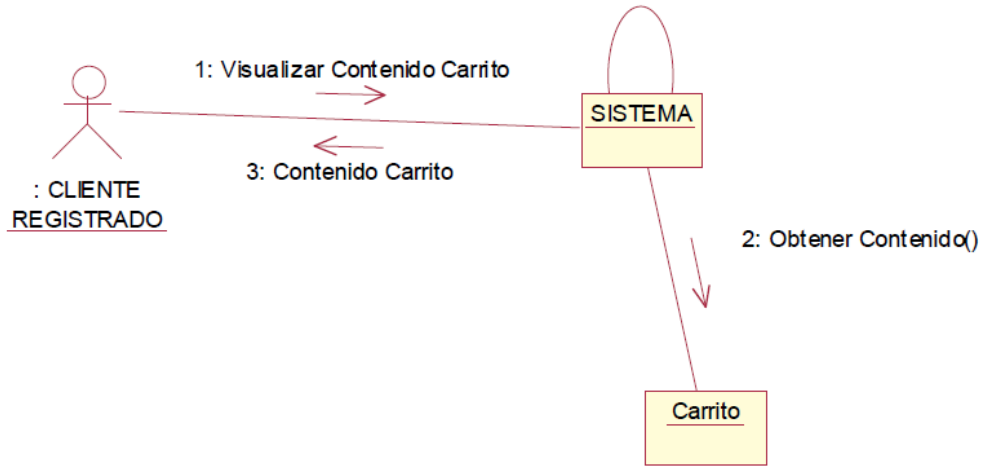


Figura 5. 45: Diagrama de Colaboración compra pedido **Fuente:** Elaboración propia

F. ESTABLECER FORMA DE PAGO

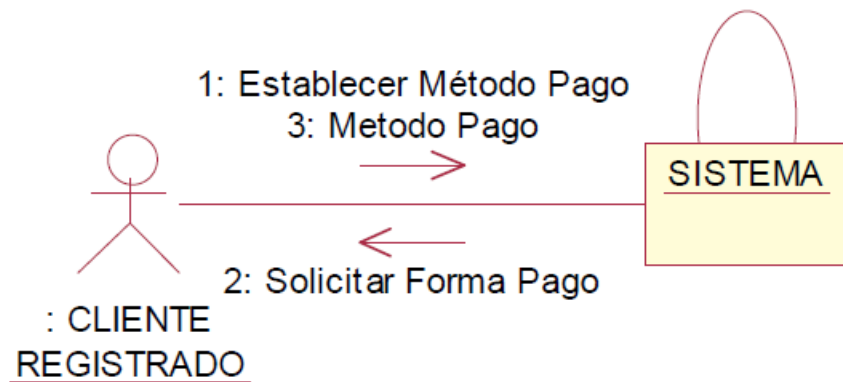


Figura 5. 46: Diagrama de Colaboración forma de pago **Fuente:** Elaboración propia



G. IDENTIFICACION DE USUARIO

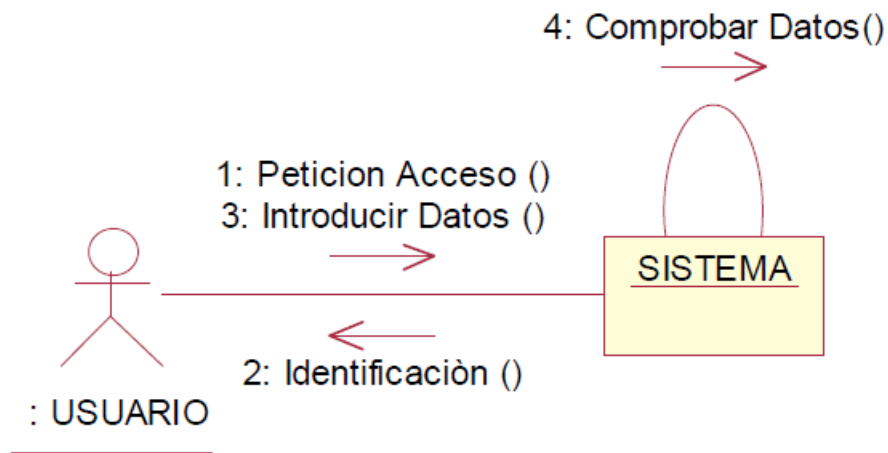


Figura 5. 47: Diagrama de Colaboración identificar usuario **Fuente:** Elaboración propia

H. GESTIONAR PRODUCTO

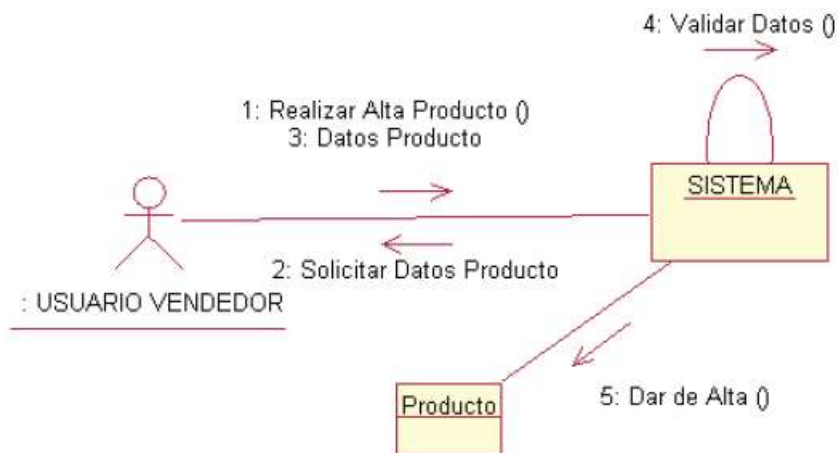


Figura 5. 48: Diagrama de Colaboración gestionar producto **Fuente:** Elaboración propia

I. GESTIONAR PRODUCTO - VISUALIZAR

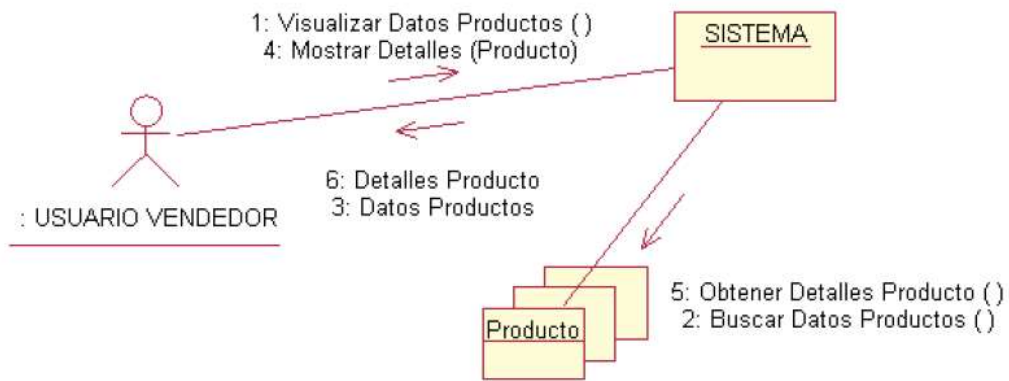


Figura 5. 49 : Diagrama de Colaboración producto - visualizar Fuente: Elaboración propia

J. GESTIONAR PRODUCTO - MODIFICAR

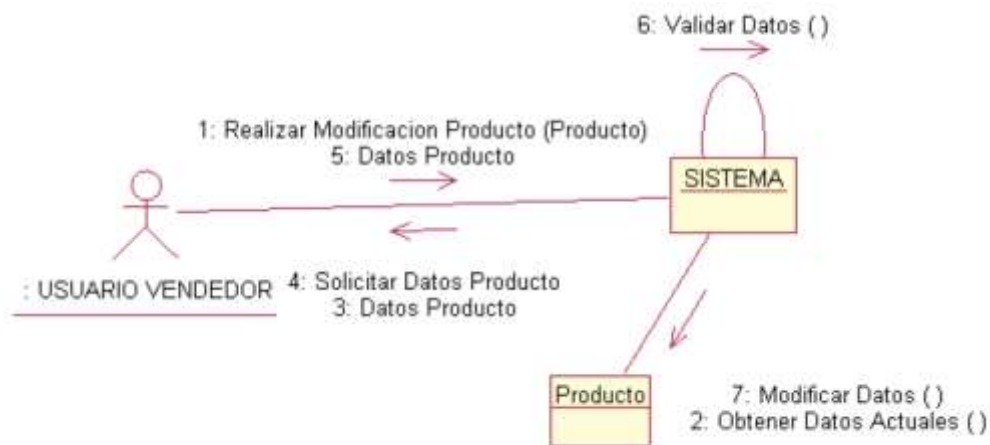


Figura 5. 50 Diagrama de Colaboración producto - modificar Fuente: Elaboración propia



K. GESTION CATEGORIA PRODUCTO

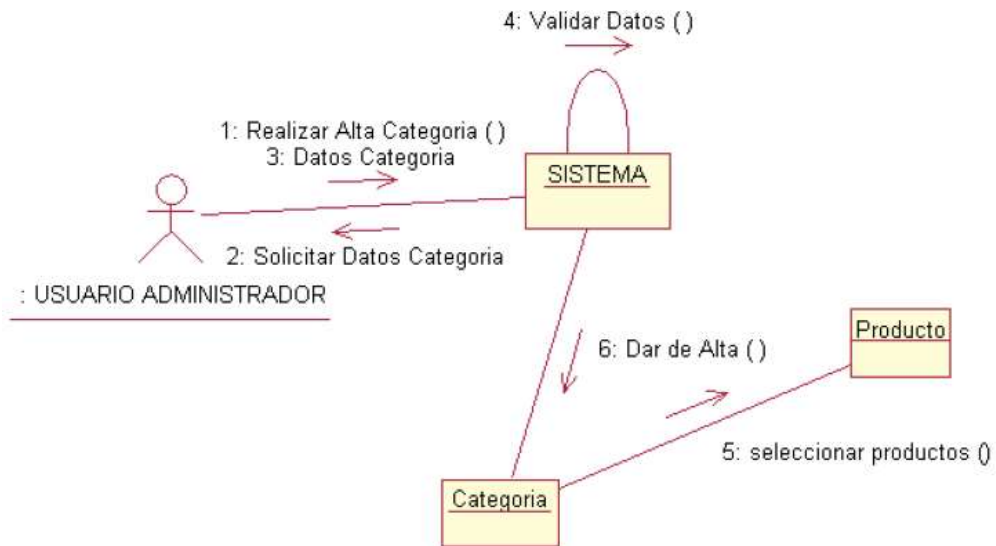


Figura 5. 51 : Diagrama de Colaboración categoría producto Fuente: Elaboración propia



A. CONSULTAR PRODUCTO:

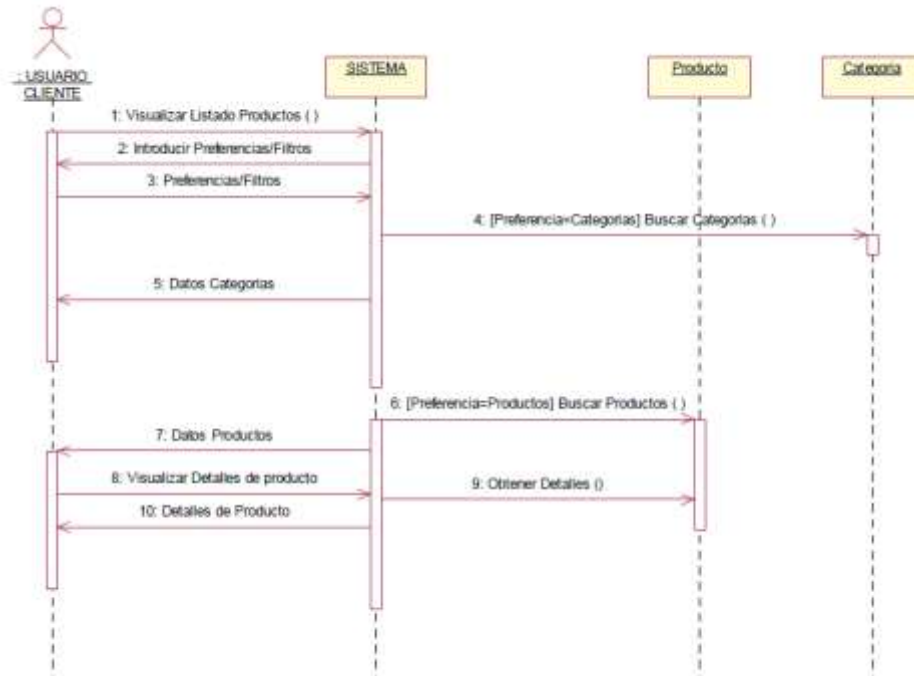


Figura 5. 52: Diagrama de secuencia consultar producto **Fuente:** Elaboración propia



A. GESTIONAR DATOS DEL CLIENTE - VISUALIZAR

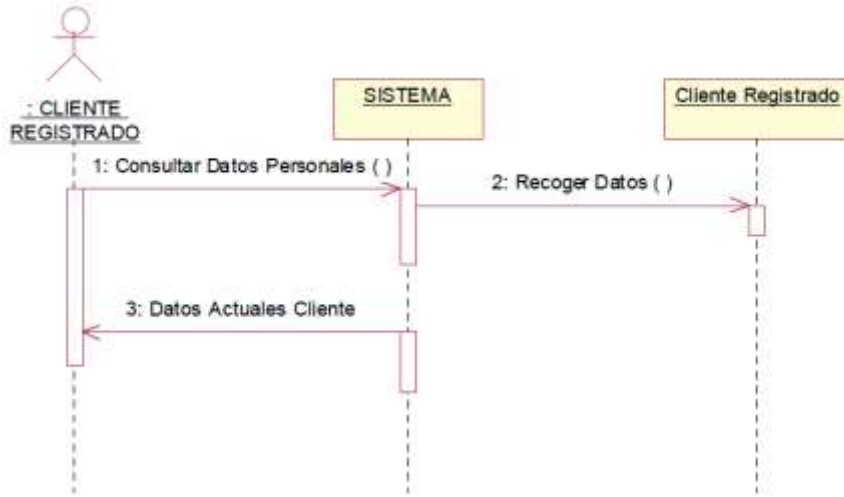


Figura 5. 53: Diagrama de secuencia datos cliente - visualizar Fuente: Elaboración propia

B. GESTIONAR DATOS DEL CLIENTE – MODIFICAR

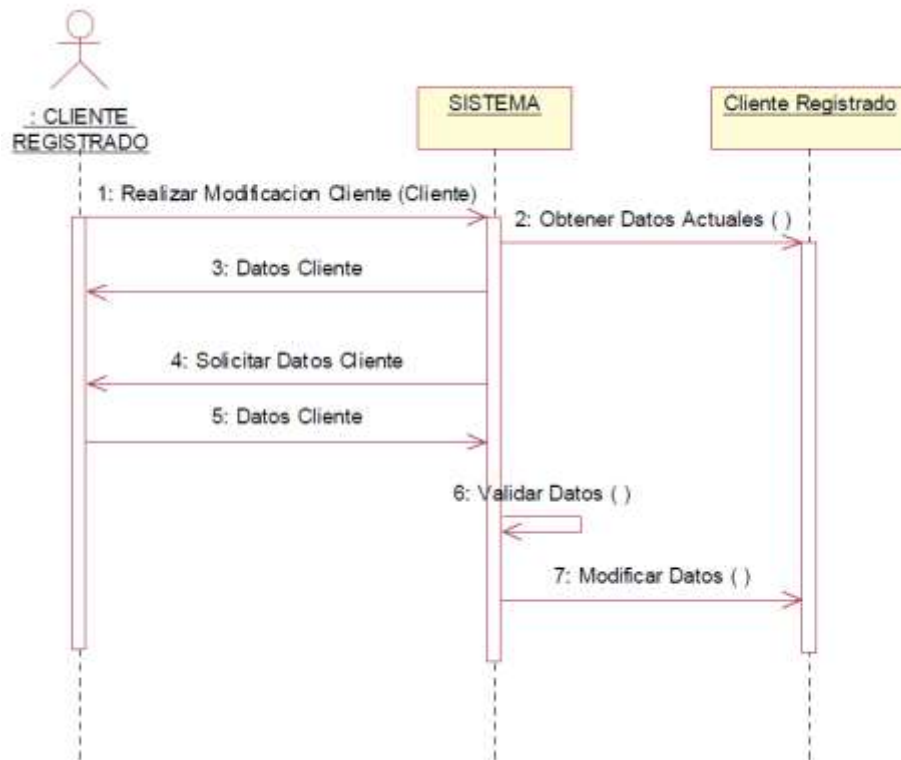


Figura 5. 54 Diagrama de secuencia datos cliente - modificar Fuente: Elaboración propia



C. GESTIONAR DATOS DEL CLIENTE - REGISTRAR

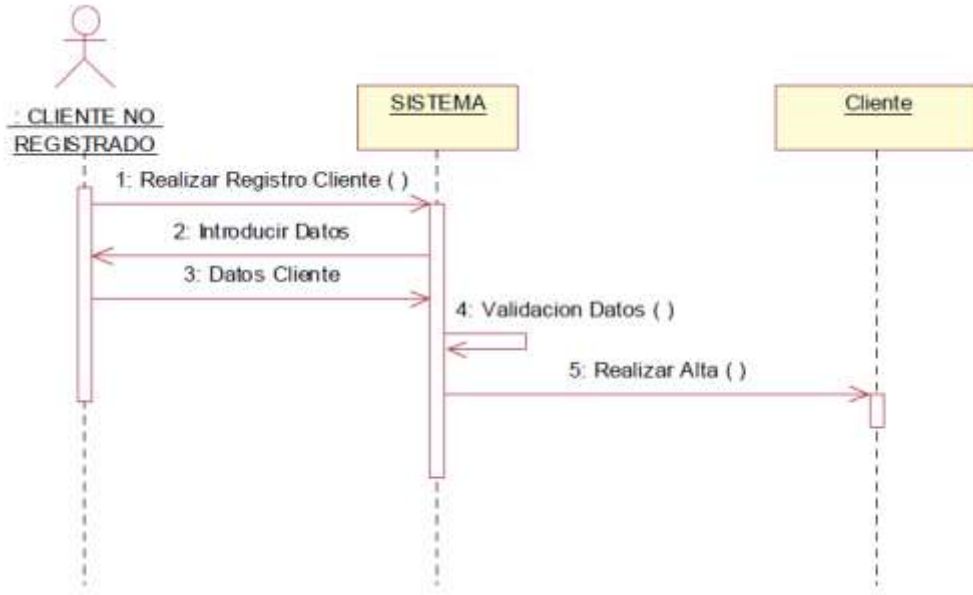


Figura 5. 55: Diagrama de secuencia **datos cliente registrar** Fuente: Elaboración propia

D. GESTIONAR COMPRA

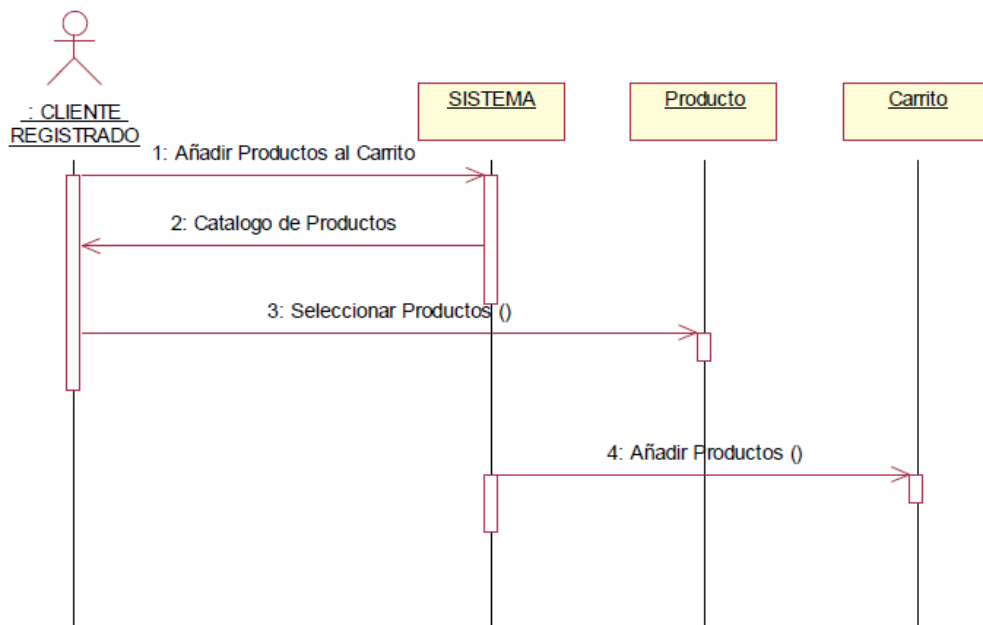


Figura 5. 56 Diagrama de secuencia compra Fuente: Elaboración propia



E. GESTIONAR COMPRA - VISUALIZAR

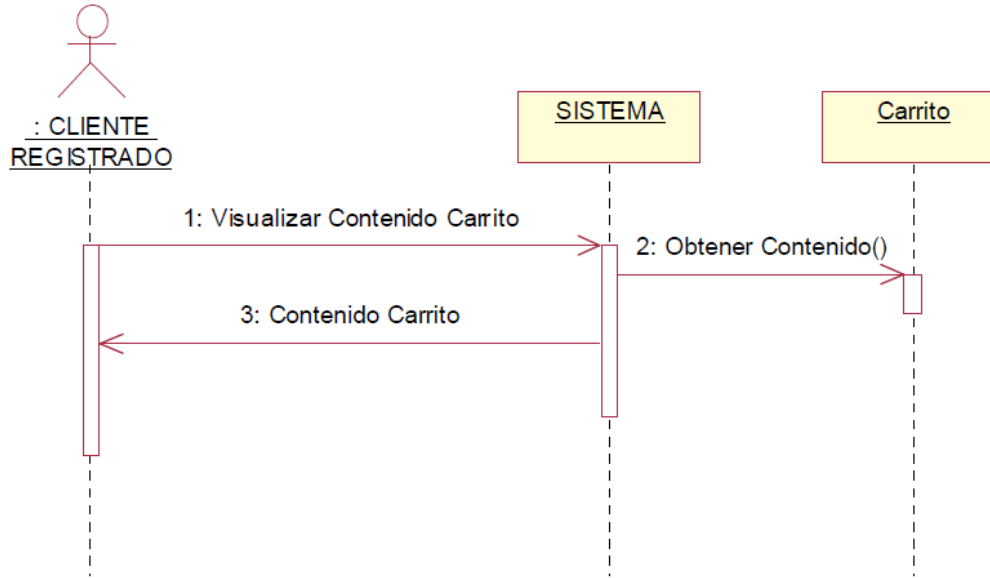


Figura 5. 57: Diagrama de secuencia compra- visualizar Fuente: Elaboración propia

F. GESTIONAR COMPRA PEDIDO

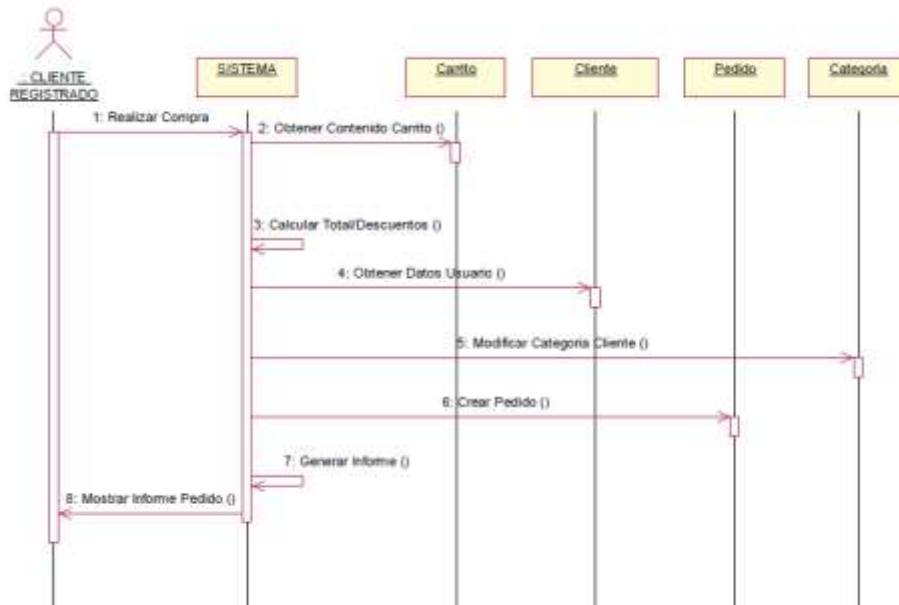


Figura 5. 58: Diagrama de secuencia compra pedido Fuente: Elaboración propia



G. ESTABLECER FORMA DE PAGO

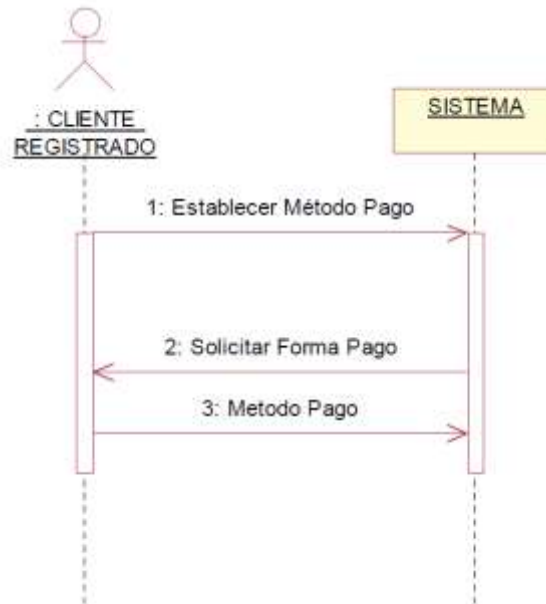


Figura 5. 59: Diagrama de secuencia forma de pago Fuente: Elaboración propia

H. IDENTIFICACION DE USUARIO

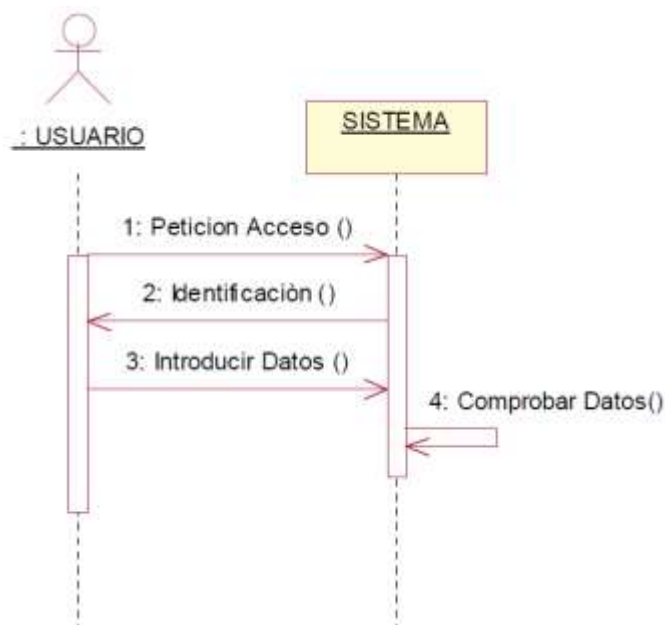


Figura 5. 60 Diagrama de secuencia identificación de usuario Fuente: Elaboración propia



I. GESTIONAR PRODUCTO

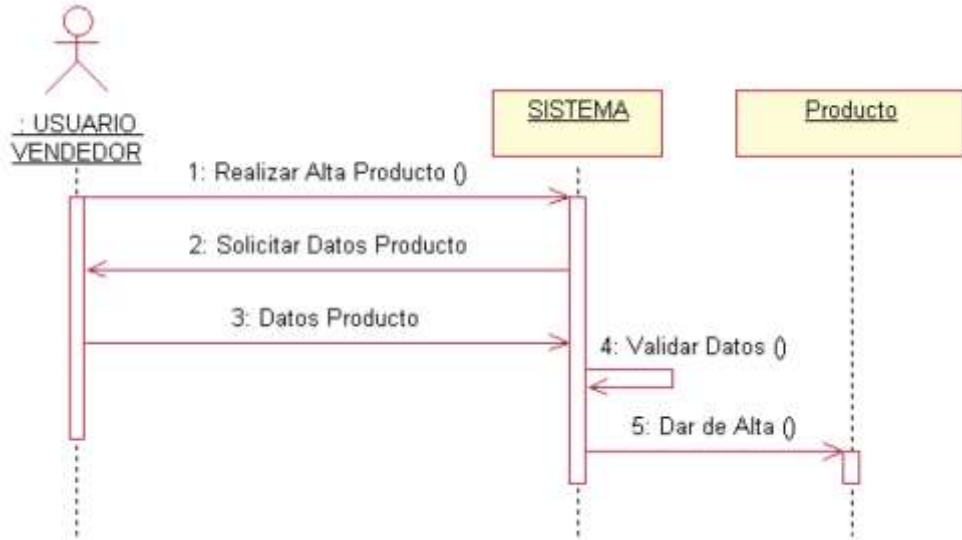


Figura 5. 61: Diagrama de secuencia producto Fuente: Elaboración propia

J. GESTIONAR PRODUCTO - VISUALIZAR

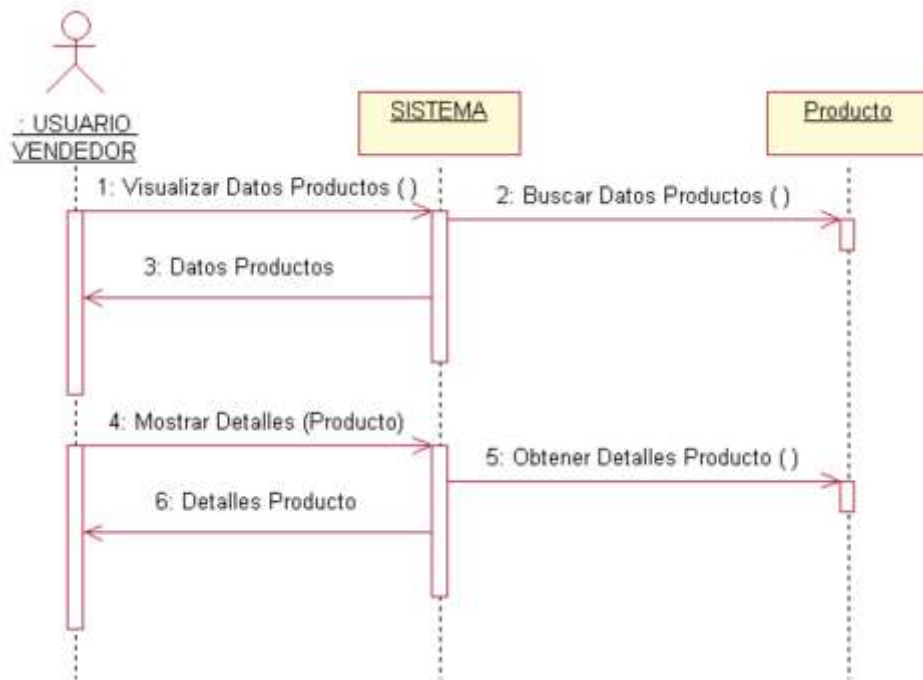


Figura 5. 62 Diagrama de secuencia producto visualizar Fuente: Elaboración propia



K. GESTIONAR PRODUCTO – MODIFICAR

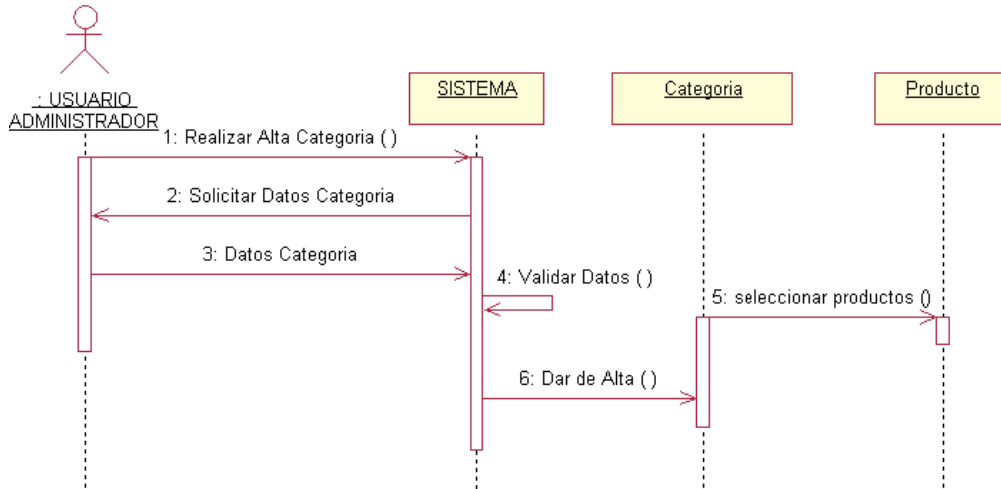


Figura 5. 63: Diagrama de secuencia producto - modificar Fuente: Elaboración propia

A. GESTION CATEGORIA PRODUCTO

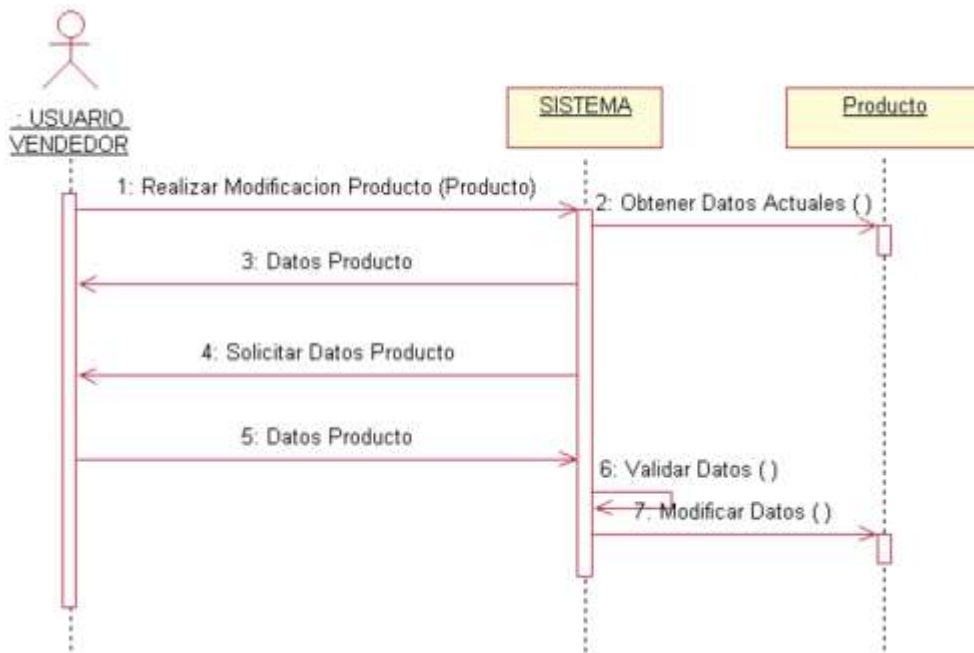


Figura 5. 64: Diagrama de secuencia categoría producto Fuente: Elaboración propia



5.7.2.4. Etapa N° IV: “Diseño de sistemas”

Objetivo:

Representar los componentes con los cuales se va a construir el software.

A continuación mostramos los productos finales obtenidos los cuales son:

- **Modelo de Datos (Modelo Relacional en SQL).-**
Modela la base de datos, diseñada en un Gestor de base de datos.
- **Interface Gráfica de Usuarios (GUI).-** Presenta las interfaces del sistema, diseñada en un Lenguaje de Programación.

INTERFACES DE LA PROPUESTA DE LA WEB

Figura 5. 67: de resumen de compra **Fuente:** Elaboración propia



Figura 5. 68: Interfaz registrar o ingresar usuario **Fuente:** Elaboración propia



Figura 5. 69: Interfaz de llenado de datos del usuario Fuente: Elaboración propia

Figura 5. 70: Interfaz de llenado de datos del usuario Fuente: Elaboración



Figura 5. 71: Interfaz de datos de dirección Fuente: Elaboración propia

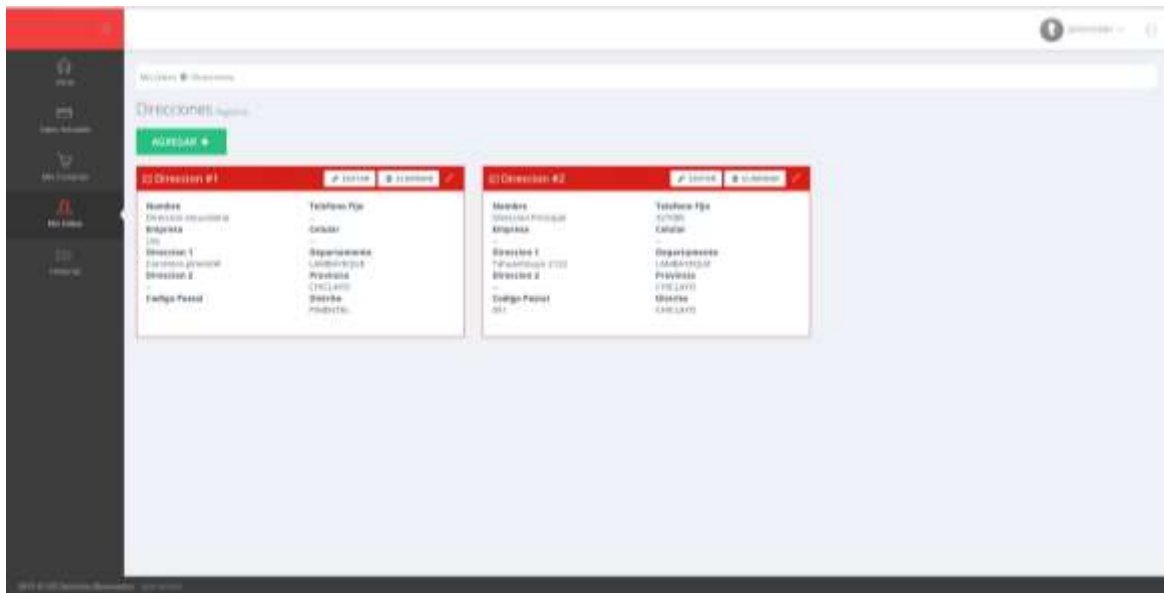


Figura 5. 72: Interfaz de resumen de compra Fuente: Elaboración propia

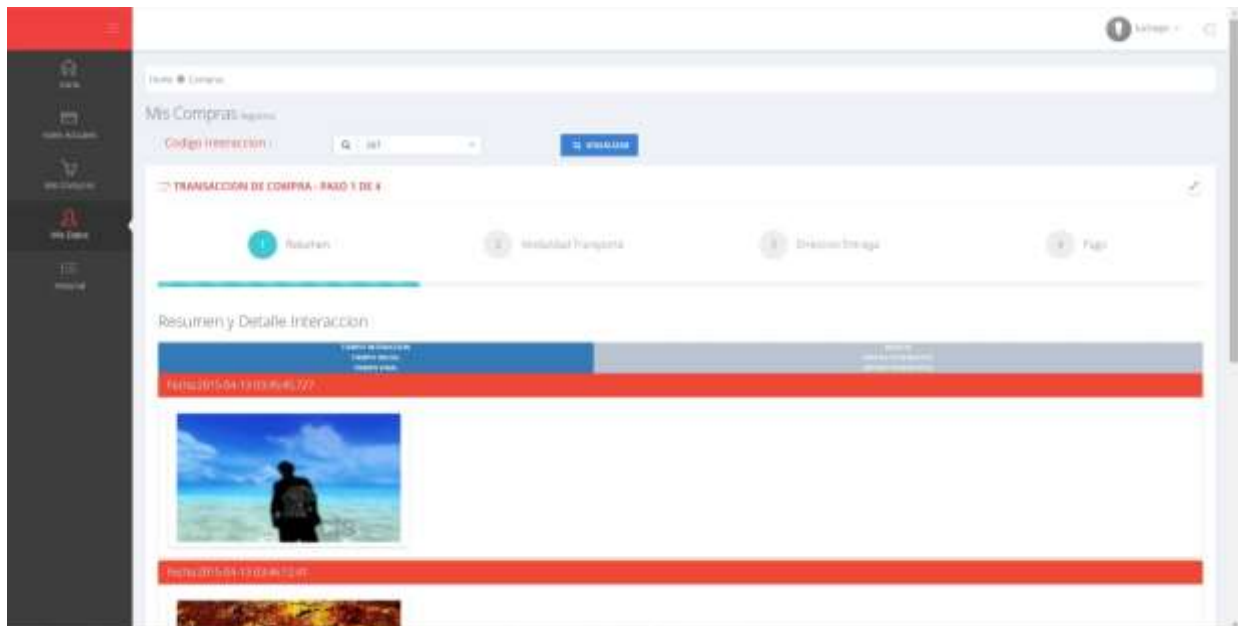


Figura 5. 73: Interfaz de cuenta de usuario Fuente: Elaboración propia

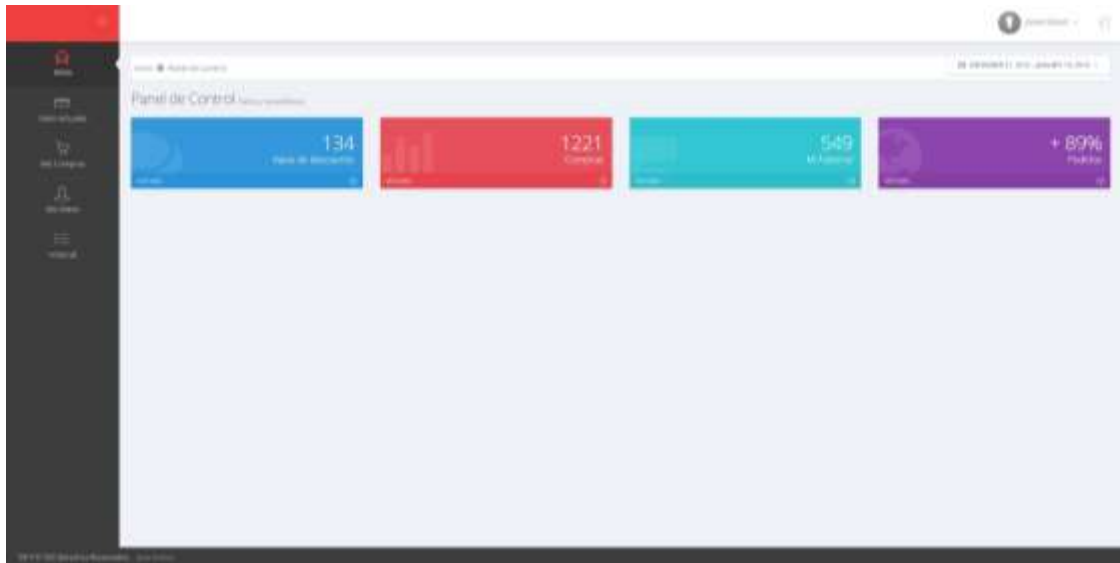


Figura 5. 74: Interfaz de agregar prendas en aplicación Fuente: Elaboración propia

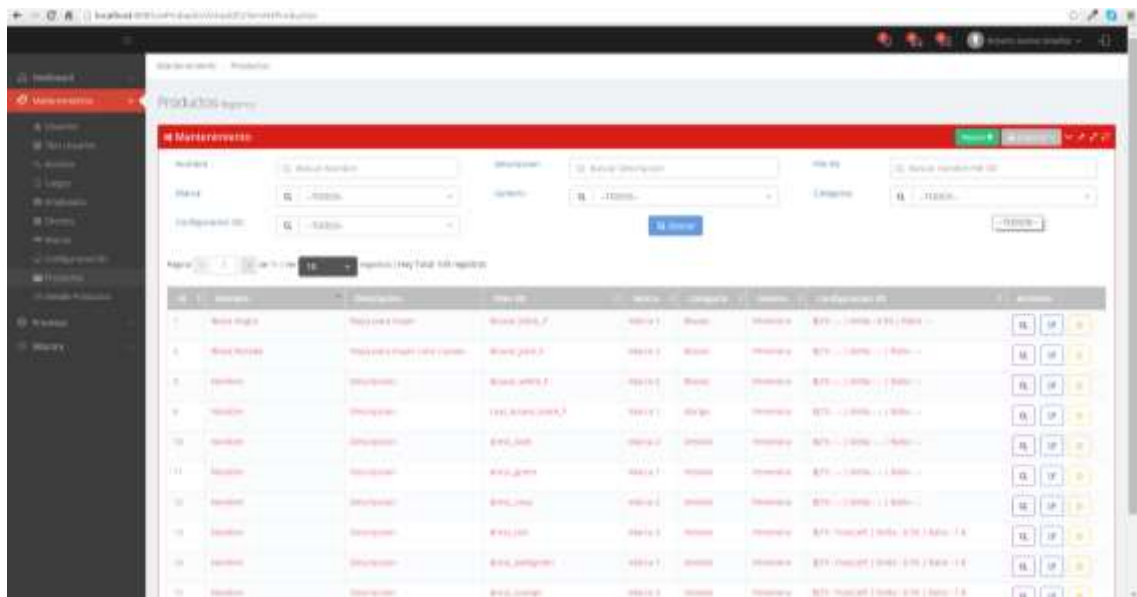
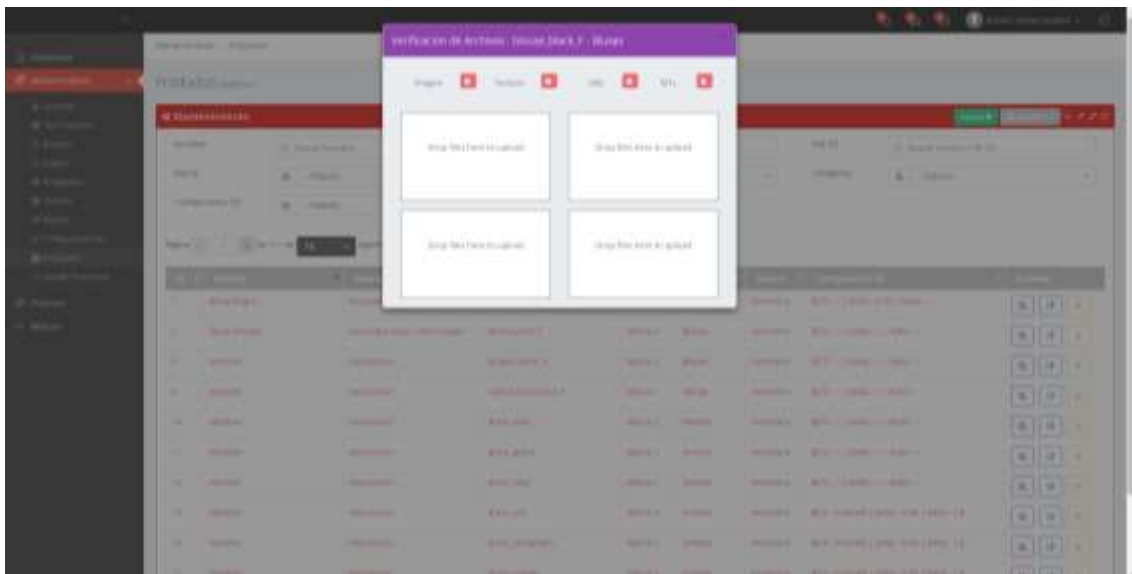


Figura 5. 75: Interfaz de selección de prendas a agregar **Fuente:** Elaboración propia



5.7.2.5. Etapa N° V: “Implementación del Sistema”

Objetivo:

Describir el código producido, los archivos generados (fuente, intermedios, ejecutables) y adquiridos (bibliotecas de funciones) y las herramientas y procedimientos requeridos para obtener, instalar y poner en funcionamiento los ejecutables del sistema.

A continuación mostramos los productos finales obtenidos los cuales son:

- **Modelo de Componentes.-** Describe los archivos que integran el sistema y las relaciones de dependencia que existen entre ellos.



DIAGRAMA DE COMPONENTES ESQUEMA GENERAL

Figura 5. 76: Diagrama de componentes de esquema general Fuente: Elaboración propia

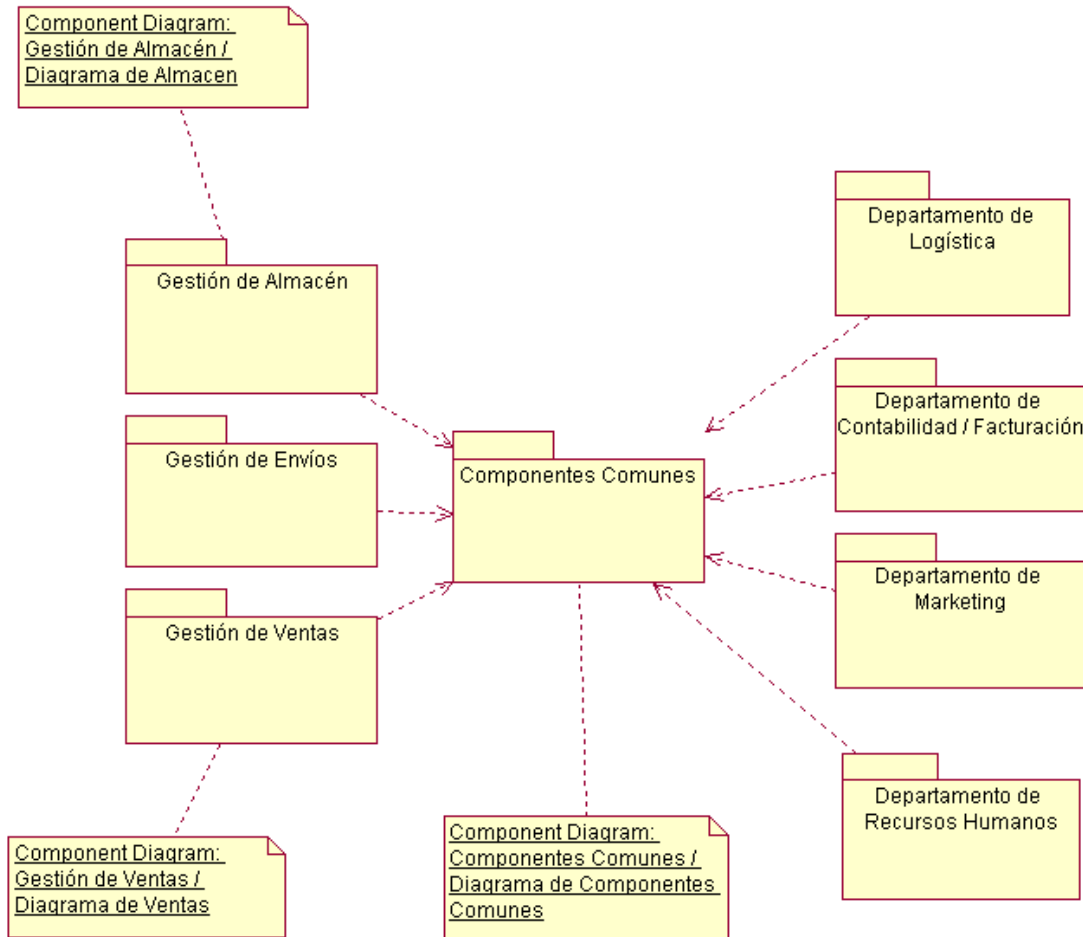


DIAGRAMA DE COMPONENTE ALMACÉN

Figura 5. 77: Diagrama de componentes de almacén Fuente: Elaboración propia

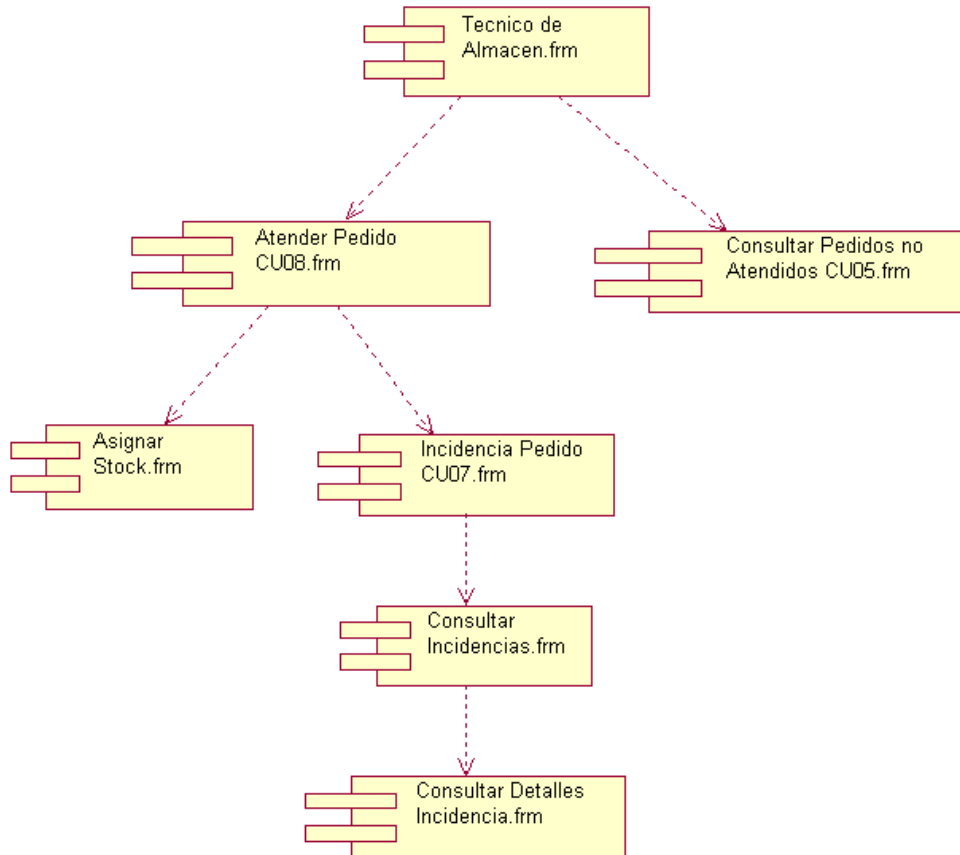


DIAGRAMA DE COMPONENTE VENTAS

Figura 5. 78: Diagrama de componentes de ventas Fuente: Elaboración propia

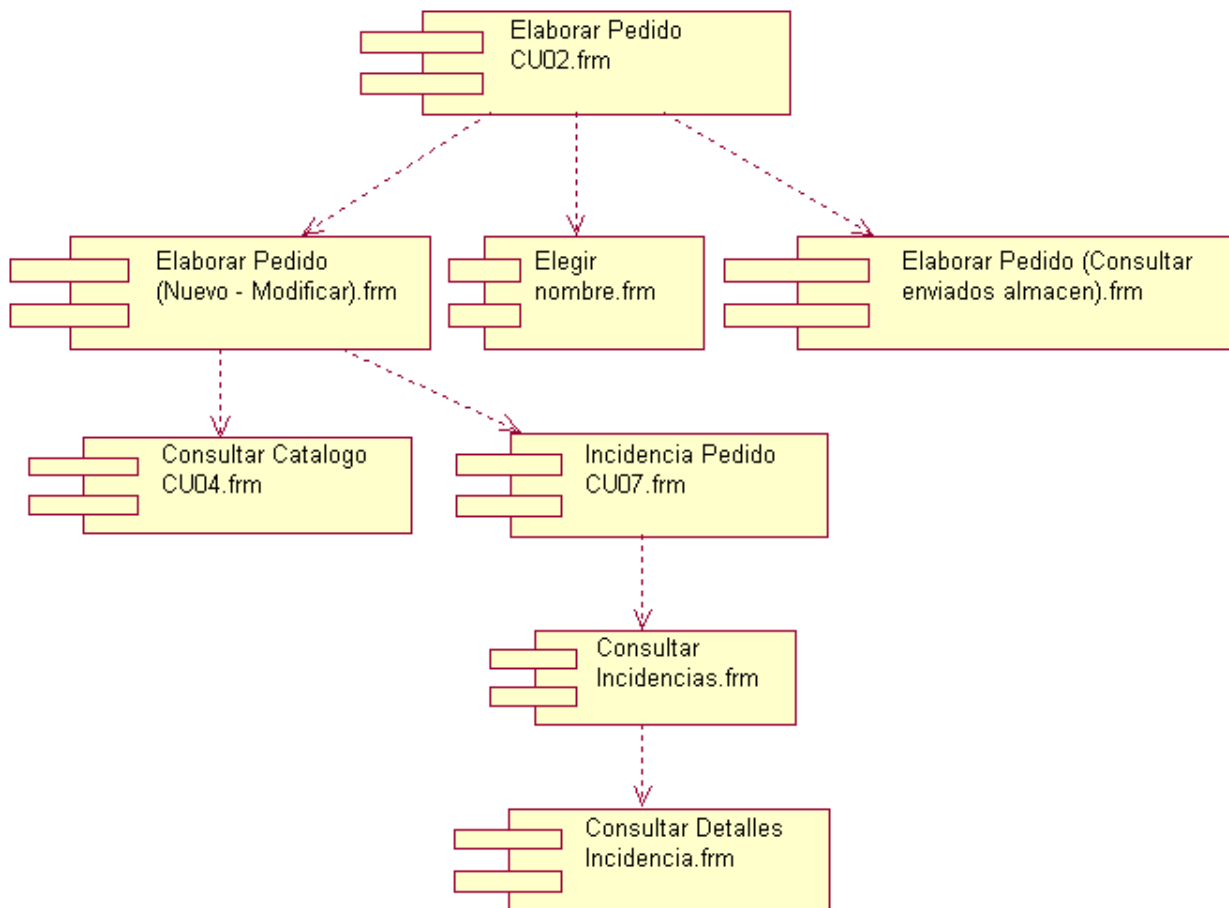
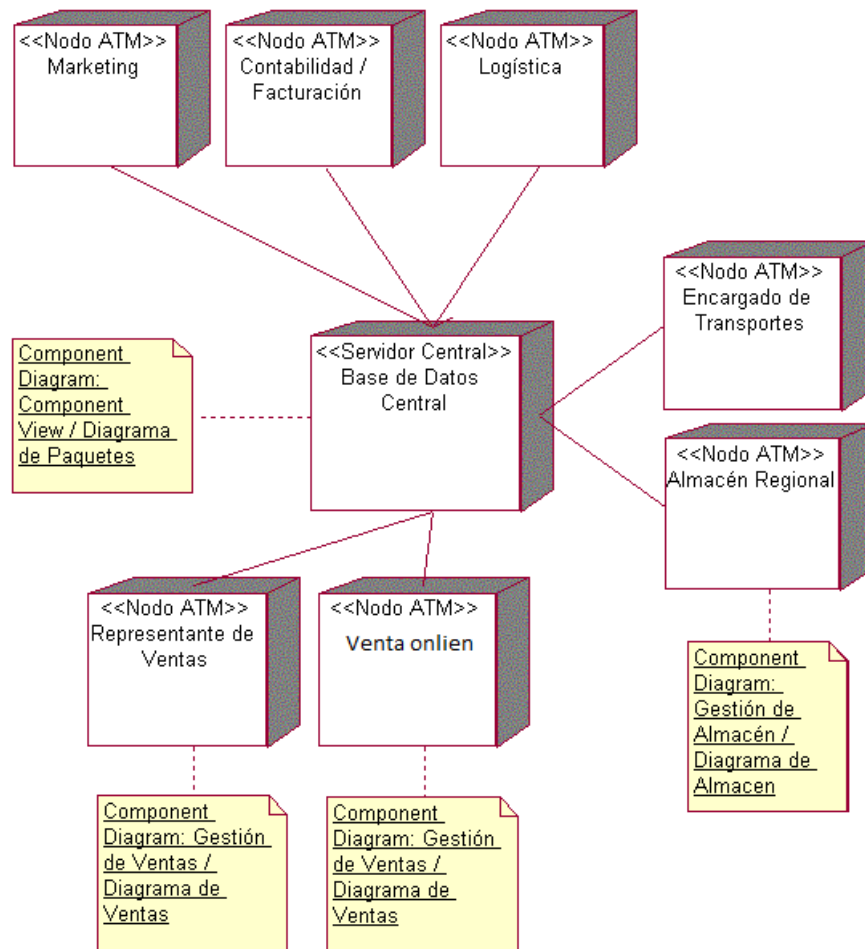


DIAGRAMA DE DESPLIEGUE

Figura 5. 79: Diagrama de desplique Fuente: Elaboración propia



5.8. Niveles De Seguridad Y Planes De Contingencia

5.8.1. CREACIÓN DE POLÍTICAS DE SEGURIDAD

La seguridad de un sitio electrónico tiene que ser confiable para que el mismo tenga éxito., este involucra las siguientes partes:

1. **Privacidad:** que las transacciones no sean visualizadas por nadie.
2. **Integridad:** que los datos o transacciones como números de tarjeta de créditos o pedidos no sean alterados.
3. **No Repudio:** posibilita que el que generó la transacción se haga responsable de ella, y brinda la posibilidad de que este no la niegue.
4. **Autenticación:** que los que intervienen en la transacción sean leales y válidas.
5. **Facilidad:** que las partes que intervienen en la transacción no encuentren dificultad al hacer la transacción.

Una vez planteada una política de seguridad, debemos de establecer las medidas para que cumpliendo con la política de seguridad las pérdidas sean las menores posibles y que esto se transforme en ganancias, ya sean materiales o de imagen. Las posibles medidas que se pueden establecer se pueden dividir según la siguiente tabla:

Tipos	Protección	Medidas	Medidas de Organización
	Física	Técnicas	
Preventivas	PF	PT	PO
Detectives	DF	DT	DO
Correctivas	CF	CT	CO

PF: Control en el acceso al software

DF: Monitor de vigilancia, detector de metales, detector de movimiento.

CF: Respaldo de fuente de poder.

PT: Firewalls, criptografía.

DT: Control de acceso lógico, sesión de autenticación.

CT: Programa antivirus.

PO: Cursos de actualización, organización de las claves.

DO: Monitoreo de auditoria.

CO: Respaldos automáticos, plan de incidentes (sanciones).

El ingreso al sitio está permitido a cualquier persona, la cual puede ejecutar búsquedas y ver información de productos, pero el acceso a los servicios de compra que ofrece el sitio estará destinado especialmente a los clientes que tengan una cuenta de usuario válida. Al igual que el administrador, el cual es el único que puede acceder a los formularios de mantenimiento y



bases de datos, pero a pesar de que puede navegar libremente en la misma, no puede hacer cambios en los datos del cliente, información que solamente con la cuenta de usuario_cliente puede acceder y modificar. Estas cuentas de usuarios con la que inician sesión, pueden ser una combinación de letras o números (nombre de usuario y contraseña).

El ingreso a la base de datos está definido por la seguridad que presenta SQL Server, la cual se empleará con el inicio de sesión del administrador del sistema, el cual se declara al momento de la instalación, donde se establece el nivel de auditoría que tendrá la base de datos.

Como es una aplicación Web esta se encuentra expuesta a cualquier ataque o violación de acceso, por lo tanto, otra medida de seguridad con que se cuenta es el uso de firewalls, programas antivirus, con los cuales se estaría evitando el ingreso de usuarios mal intencionados que puedan poner en riesgo la confidencialidad y el uso de los datos que se poseen.

El acceso al área de la empresa donde se encuentre el servidor con la base de datos y la aplicación estará restringido solo a personal autorizado, por lo tanto, se deberá tener un monitoreo y control del área para evitar una salida o manejo inadecuado de la información.

El sitio manejará el pago por medio de tarjetas de crédito, una seguridad que será controlada por la entidad financiera emisora de la misma y que este registrando el movimiento o transacción que realice el cliente a través de un sistema que controla al detalle cada transacción y que queda en responsabilidad de la empresa que lo adquiera. Cada transacción se realizará por una encriptación de datos o una configuración ASCII que es interpretada por cada financiera o banco.

5.7.2. Creación De Usuarios

Existen tres tipos de usuarios que harán uso de la aplicación, un usuario cliente, usuario vendedor y usuario administrador. La creación del usuario cliente se llevará a cabo por medio de la aplicación Web, donde el cliente que desee comprar y no se encuentre registrado como un usuario activo crea su propia cuenta para poder tener acceso a la misma, para esto el cliente introduce sus datos personales, especifica el tipo de cliente que es (consumidor final o crédito fiscal), elige un nombre de usuario con su respectiva contraseña (existe una verificación del nombre de usuario para evitar duplicidad); luego de haber ingresado todos los datos requeridos, se envía el formulario a la base de datos, donde se registran y se validan, la respuesta

a dicha validación se envía al correo electrónico, donde el cliente debe dar conocimiento del envío de la respuesta para que su cuenta de usuario sea activada, todo esto por medio de un link que se envía en el correo electrónico; una vez activada la cuenta, el usuario podrá hacer uso de la aplicación.

Cabe mencionar que el cliente puede cambiar los datos ingresados y contraseña, en el sitio personalizado para cada uno.

El usuario vendedor, al igual que el usuario cliente este podrá acceder a la aplicación desde cualquier parte con la diferencia que su página de acceso es distinta a la del cliente, la creación de esta cuenta se lleva a cabo en las páginas que contiene los formularios de ventas la cual permite la visualización, modificar, ingresar (para nuevas ventas), su cuenta de usuario y contraseña, también tiene acceso a los formularios de consultas u otros que no involucre formularios del administrador.

El otro tipo de usuario es el administrador, al igual que el usuario cliente este podrá acceder a la aplicación desde cualquier parte con la diferencia que su página de acceso es distinta a la del cliente, la creación de esta cuenta se lleva a cabo en la páginas que contiene los formularios de mantenimiento, donde existe la forma de empleados, la cual

permite la visualización, modificar, ingresar (para nuevos usuarios administrativos –opcional) datos de empleado, su cuenta de usuario y contraseña, e igual que el usuario cliente tiene a posibilidad de hacer cambios en los mismos en una apartado destinado para esto.

El usuario administrador llevará un control de los usuarios clientes, usuario empleados, por lo tanto, este puede eliminarlos si es necesario, pero no modificar o agregarlos.

5.7.3. Evaluación De Riesgos Y Amenazas

En la actualidad, la constante evolución de la tecnología informática impacta directamente en el procesamiento de la información y la comunicación de una organización, pero todo este cambio implica a su vez nuevos riesgos & amenazas que no sólo hay que evitar, sino también prevenir.

El negocio de las organizaciones modernas depende en gran medida de la seguridad de sus sistemas/aplicaciones, hasta un punto tal que si los sistemas de la organización se detienen, se puede detener el negocio.

Esto es algo que ninguna organización competitiva de primer nivel debe permitir, dado que una interrupción o desastre informático (por mínimo que resulte) es un costo demasiado alto a pagar y lo

que es peor, se podía haber prevenido a un costo mucho mas bajo.

A continuación, podemos citar algunos de los peligros actuales que se pueden presentar:

- ✓ Intrusiones internas/externas (cambio o destrucción de información, robo o sabotaje de sistemas, denegación de servicio, etc.).
- ✓ Virus & gusanos informáticos.
- ✓ Problemas de generación o gestión de la información.
- ✓ Interrupción de suministros informáticos (electricidad, comunicaciones, etc.).
- ✓ Violación de la privacidad.
- ✓ Cyberterrorismo.
- ✓ Repudio de transacciones e-commerce.
- ✓ Hacking individual.

Es recomendable que las medidas de seguridad a implementar siempre sean proactivas, debiendo instalar sistemas de seguridad proactivos y no reactivos, siendo estas medidas más rentables desde el punto de vista de los costos.

5.7.4. Desarrollo De Planes Contingenciales

Para reducir al mínimo el impacto de las amenazas en potencia, las organizaciones deben desarrollar planes de

contingencia como parte de su proceso para evaluar estrategias. Los *planes de contingencia* se pueden definir como planes alternativos que se pueden poner en práctica cuando ciertos hechos clave no ocurren como se esperaba. Sólo las áreas que tienen verdadera prioridad requieren la seguridad de planes de contingencia. Los estrategias no pueden ni deben tratar de cubrir todas las bases, haciendo planes para todas las contingencias posibles.

Cuando las actividades para evaluar estrategias revelan rápidamente la necesidad de un cambio mayor, el plan de contingencia adecuado se puede ejecutar en forma oportuna. Los planes de contingencia pueden mejorar la capacidad del estrategia para responder velozmente a los cambios clave operados en las bases internas y externas de la estrategia presente de la organización.

Los datos de los sistemas informáticos están en constante peligro por varias causas: errores de los usuarios o ataques intencionados o fortuitos. Pueden producirse accidentes y ciertas personas con intención de atacar el sistema pueden obtener acceso al mismo e interrumpir los servicios, inutilizar los sistemas o alterar, suprimir o robar información.

Los sistemas informáticos pueden necesitar protección en algunos de los siguientes aspectos de la información:

- ❖ **Confidencialidad.** El sistema contiene información que requiere protección contra la divulgación no autorizada. Por ejemplo, datos que se van a difundir en un momento determinado (como, información parcial de informes), información personal e información comercial patentada.
- ❖ **Integridad.** El sistema contiene información que debe protegerse de modificaciones no autorizadas, imprevistas o accidentales. Por ejemplo, información de censos, indicadores económicos o sistemas de transacciones financieras.
- ❖ **Disponibilidad.** El sistema contiene información o proporciona servicios que deben estar disponibles puntualmente para satisfacer requisitos o evitar pérdidas importantes. Por ejemplo, sistemas esenciales de seguridad, protección de la vida y predicción de huracanes.

Los administradores de seguridad tienen que decidir el tiempo, dinero y esfuerzo que hay que invertir para desarrollar las directivas y controles de seguridad apropiados. Cada organización debe analizar sus necesidades específicas y determinar sus requisitos y limitaciones en cuanto a recursos y programación. Cada sistema informático, entorno y directiva organizativa es distinta, lo que hace que cada servicio y cada estrategia de seguridad sean únicos. Sin embargo, los

fundamentos de una buena seguridad siguen siendo los mismos y este documento se centra en dichos principios.

Aunque una estrategia de seguridad puede ahorrar mucho tiempo a la organización y proporcionar importantes recomendaciones de lo que se debe hacer, la seguridad no es una actividad puntual. Es una parte integrante del ciclo vital de los sistemas. El objetivo de esta tesis no es exponer planes contingenciales, esto dependerá de los intereses para los administradores de recursos de información, los directores de seguridad informática y los administradores.

5.9. Componentes hardware y software

Componentes Hardware.

En cuanto a hardware se estima los siguientes aspectos como requerimientos mínimos para la implementación de la aplicación. Como equipo de cómputo tenemos lo siguiente (solo en caso de requerirse).

A continuación, se detallan los equipos propuestos.

1. COMPUTADOR CORE I3
2. COMPUTADOR DUAL CORE
3. KINECT
4. ADAPTADOR
5. TV

1. Computador de escritorio CORE I3

Características:

Modelo: ADVANCE V05535

Procesador: INTEL PENTIUM G620 (2.60GHz, 3MB CACHE L3)

Memoria: 2GB

Disco Duro: 500 GB SATA 6.0 bbps/7200 RPM

Monitor: de 15" Dell (15" visible)

Tarjeta de Sonido: Realtek Alc887

Tarjeta de Red: Realtek 811F 10/1000 MB/S

Conector USB: 4 USB

Conecto de video: VIDEO DB-15

2. Computador de escritorio DUAL CORE

Características:

Modelo: ADVANCE VS5456

Procesador: INTEL CORE I3-2120 (3.30 GHZ, 3MB CACHE L3)

Memoria: 4GB DDR3

Disco Duro: 500 GB SATA 6.0 bbps/7200 RPM

Monitor: de 15" Dell (15" visible)

Tarjeta de video: 1GB GEGORCE GT520 1GB DDR3

Tarjeta de Sonido: Realtek Alc887

Tarjeta de Red: Realtek 811F 10/1000 MB/S

Conector USB: 4 USB

Conecto HDMI: 1 HDMI

3. Sensor Kinect Xbox one

Características

1 cámara RGB de 1080p a 30 FPS

1 cámara de infrarrojos

3 infrarrojos de control remoto

1 procesador

1 memoria

4 fases

1 LED de encendido.

4. Adaptador Kinect Xbox one

Características

Marca KMD

Adaptador de Corriente AC para Kinect

CA: 100V-240V, CC: 12V

Compatible con Kinect para Xbox one

5. Televisor

Características

LG 42L85500

Led Full HD 1080p

42"

HDMI 2

USB 1

Sintonizador Digital

Triple XD Engine



Componentes Software.

En cuanto al software la institución debería de contar como mínimo con la versión del sistema operativo que se requiere para el buen funcionamiento del sistema propuesto

A continuación, se detallan los Softwares propuestos.

1. Windows 8
2. Visual studio 2013
3. Sql server 2012
4. Sdk Kinect
5. Kinect Speech Language Pack 11.0
6. Autodesk 3D max
7. NETBEANS

5.10. Análisis costo beneficio.

El análisis Costo Beneficio permitirá demostrar la rentabilidad del proyecto de implementación, el cual será evaluado no sólo desde el punto de vista monetario, sino también considerando los beneficios intangibles que se obtendrán.

5.10.1. Inversión Inicial:

a) Costos de Servicios y Materiales:

Tabla 5. 4: Costos de servicios **Fuente:** Elaboración Propia

Descripción	Promedio Monto	Días por Mes(S/.)	Meses	Total(S/.)
Transporte	4,00	20	5	480,00
Servicio de Internet	1,00	20	5	100,00
Servicio de Luz(*)	0.86	20	5	86,00
Monto Total(S/.)				666.19
(*)Tomando en cuenta que el consumo de una computadora por hora es de 0.300Kw, Y el costo de energía tiene un valor de S/.0.3583 x Kwh , entonces el monto promedio diario se calculó de la siguiente forma: $0.3Kw/hr*8hr/día*S/.0.3583=S/.0.86$ soles diarios.				

Los servicios descritos en el cuadro anterior, corresponden a los gastos Realizados concurrentemente durante los 5 meses de desarrollo del proyecto.

Tabla 5. 5: Costos de materiales **Fuente:** Elaboración Propia

Descripción	Cantidad	Unidad	Precio Unitario(S/.)	Total(S/.)
Impresiones	720	Hojas	0,20	144,00
Fotocopias	100	Hojas	0,05	5,00
Papel BondA4	1	Millar	13,00	13,00
CD's	4	Unidad es	1,00	4,00
Insumos de oficina	1	Juego	20,00	20,00
Empastados	3	Juegos	15,00	45,00
Monto Total(S/.)				231,00



Estos gastos están relacionados en su mayoría a la documentación del proyecto.

b) Costos de Personal:

Tabla 5. 6: Costos de Personal **Fuente:** Elaboración Propia

Descripción	Duración			Costo por Hora (S/.)	Total(S/.)
	Meses	Días por mes	Horas por Día		
Análisis y Diseño (*)	3	20	4	10,00	2100. ⁰⁰
Implementación(**)	3	20	5	15,00	3300. ⁰⁰
Instalación y Pruebas	4	20	4	10,00	2500. ⁰⁰
Monto Total(S/.)					7900.⁰⁰
(*)El modelado del negocio y requerimientos corresponden a las tareas de análisis del desarrollador. (**)La implementación está referida a la construcción del software por parte del programador, considera los artefactos utilizados en las etapas anteriores de la metodología RUP.					

El tiempo de análisis, diseño e implementación; corresponde a los 5 meses empleados en el desarrollo completo del proyecto. Esto considerando solo los meses programados.



c) Costos de software

Tabla 5. 7: Licencias de Software de Desarrollo e Implementación **Fuente:**

Elaboración Propia

Descripción	Total(S./.)
Software	
Sql server 2014	1116. ⁰⁰
Visual Studio 2013	926. ⁹⁰
Windows 8	620. ⁰⁰
SDK KINECT	00. ⁰⁰
Autodesk 3D Max 2014 (3años)	8000. ⁰⁰
netbeans-windows	00. ⁹⁰
KinectSpeechLanguagePack_es-MX	00. ⁰⁰
Total	10663. ⁸⁰

d) Costos de hardware

Tabla 5. 8: Costo de hardware para la Implementación **Fuente:** Elaboración Propia

Descripción	Total(S./.)
Hardware	
Adaptador Kinect	150. ⁰⁰
Kinect	500. ⁰⁰
Tv	1500. ⁰⁰
PC 1	500. ⁰⁰
PC 2	1300. ⁰⁰
Total	3950. ⁰⁰



e) Gastos operativos

Tabla 5. 9: Gastos operativos I **Fuente:** Elaboración Propia

Descripción	Duración			Costo por Hora (S/.)	Total(S/.)
	Meses	Días por mes	Horas por Día		
Administración de la aplicación	10	15	6	10,00	9000,00
Total de Gastos Operativos					9,000.00

Tabla 5. 10: Gastos operativos II **Fuente:** Elaboración Propia

Descripción	N° Equipo	Veces x año	Costo por Hora (S/.)	Total(S/.)
Mantenimiento de la PC's	2	3	50.00	300.00
Total de Gastos Operativos				300.00

5.10.2. Beneficios:

Los beneficios serán estimados en base a algunos indicadores y otros aspectos que se lograrían al implantar el sistema, además se describirán los beneficios intangibles los cuales no serán calculados por ser beneficios sociales.



Beneficios tangibles

Tabla 5. 11: Ahorro de personal Fuente: Elaboración Propia

Sin implementación del sistema	
Cantidad actual de personal de vestir	4
Sueldo actual (S/)	850
Promedio anual(meses)	12
Total por año	40,800.00

Con la implementación del sistema	
Cantidad actual de personal(U)	1
Sueldo actual (S/)	850
Promedio anual(meses)	12
Total por año	10,200.00

Ahorro Actual 30,600.00

Ahorro en personal, ya que hay efectivos que están realizando el trabajo ayudar a escoger la prenda de vestir



Tabla 5. 12: Ahorro de probadores **Fuente:** Elaboración Propia

Sin implementación del sistema	
Cantidad actual de probadores de vestir	5
Costo actual (S/)	350.00
Total	1,750.00

Con la implementación del sistema	
Cantidad actual de probadores de vestir	1
Costo actual (S/)	350.00
Total	350.00

Ahorro Actual 1400.00

Ahorro en probadores de vestir, ya que hay efectivos que están realizando el trabajo Prueba de prendas



Beneficios intangibles

- Reducción del tiempo promedio en la prueba de alguna prenda de vestir
- Reducción del tiempo promedio de compra de alguna prenda de vestir.
- Aumento en la seguridad de prueba de prenda de vestir.
- Reducción del circuito para la obtención de la prenda de vestir.
- Incremento de la productividad de ventas.
- Incremento en la Imagen institucional.

5.11. Flujo de Caja

Tabla 5. 13: Flujo de caja **Fuente:** Elaboración Propia

Descripción	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5	Total
INVERSION INICIAL	(S/)	(S/)	(S/)	(S/)	(S/)	(S/)	(S/)
Costo de Servicios	666.00	-	-	-	-	-	666.00
Costo de materiales	231.00	-	-	-	-	-	231.00
Costo de personal	7,900.00	-	-	-	-	-	7900.00
Costo de software	10663.80	-	-	-	-	-	10663.80
Costo de hardware	3950.00	-	-	-	-	-	3950.00
Total de Inversión inicial	23410.80	-	-	-	-	-	23410.80
GASTOS OPERATIVOS	(S/)	(S/)	(S/)	(S/)	(S/)	(S/)	(S/)
Administración de la aplicación	9,000.00	9,000.00	9,000.00	9,000.00	9,000.00	9,000.00	54,000.00
Mantenimiento de PCS	300.00	300.00	300.00	300.00	300.00	300.00	1800.00
Total de Gastos Operativos	9,300.00	9,300.00	9,300.00	9,300.00	9,300.00	9,300.00	55,800.00
Total de Gastos por año	23410.80	9,150.00	9,150.00	9,150.00	9,150.00	9,150.00	78,310.80
Total de Gastos acumulados	23410.80	32,560.80	41,710.80	50,860.80	60,010.80	69,160.80	
BENEFICIOS	(S/)	(S/)	(S/)	(S/)	(S/)	(S/)	(S/)
Ahorro en Personal	-	30,600.00	30,600.00	30,600.00	30,600.00	30,600.00	153,000.00
Ahorro en probadores	-	1,400.00	0	0	0	0	1,400.00
Total Bruto de beneficios por año	-	32,000.00	30,600.00	30,600.00	30,600.00	30,600.00	154,400.00
Total Bruto de beneficios acumulados	-	32,000.00	62,600.00	93,200.00	123,800.00	154,400.00	

Total neto del beneficio por año		32,000.00	30,600.00	30,600.00	30,600.00	30,600.00	154,400.00
Total Neo de beneficio acumulado		32,000.00	62,600.00	93,200.00	123,800.00	154,400.00	
FLUJO DE CAJA NETO	(S/)	(S/)	(S/)	(S/)	(S/)	(S/)	(S/)
Flujo de caja neto anual	-15410.80	32,000.00	62,600.00	93,200.00	123,800.00	154,400.00	450,589.20
Flujo de caja neto acumulado	-15410.80	16,589.2	79,189.2	172,389.2	296,189.2	450,589.2	
TIR	27.75%						
VAN	S/.244,734.912						

VAN

El VAN es un procedimiento que permitirá calcular el valor presente de un determinado número de flujos de caja futuros.

El cálculo del VAN se ha realizado utilizando la función VNA (tasa; periodo_1: periodo_5), del programa Microsoft Excel. Donde la tasa de interés es del 18% y los periodos son extraídos del Flujo de Caja Neto Anual presentado en tabla que se mostró anteriormente, obteniendo como resultado S/244,734.912; lo que indica el óptimo nivel de rentabilidad del proyecto propuesto, y que se recupera al tercer año.

TIR

El TIR es la tasa de interés con la cual el valor actual neto o valor presente neto (VAN o VPN) es igual a cero.

Al igual que el VAN, el cálculo del TIR se ha realizado utilizando la función de Microsoft Excel: TIR (periodo_0: periodo_5).

El resultado obtenido es de 27.75%.



CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones.

1. Referente al objetivo “Caracterizar los gestos que se usaran para la comunicación”. Se concluye que los gestos más comunes para la interacción entre usuario y aplicación son ineficientes generando ciertas limitaciones, por lo que es necesario la implementación de una aplicación que permita optimizar la comunicación entre usuario y aplicación.
2. Referente al objetivo “Desarrollar una aplicación bajo la tecnología Kinect, intuitiva y rápida en la NUI”. Podemos rescatar que como se ha utilizado un lenguaje de programación C# con la tecnología Kinect estos han permitido el desarrollo de la aplicación para la interacción entre usuario y aplicación.
3. Referente al objetivo “Desarrollar un sistema web de venta que interactúe con la aplicación.” Se expone que se ha utilizado un lenguaje de programación HTML, JSP y JAVASCRIPT, estos han permitido el desarrollo del sistema web para el despliegue de la API y así comprar la prenda de vestir.
4. Referente al objetivo “Analizar resultados” se constata que al analizar los resultados demuestra que la aplicación tiene una interacción natural.

6.2 Recomendaciones.

1. Que para que exista una mejor aplicación de realidad aumentada usando el Kinect, se debe de tener conocimientos previos para el desarrollo y modificación del código fuente; además de tener en cuenta la versión de los SDK KINECT que se usan.
2. Por tratarse de una aplicación nueva, se sugiere implementar tareas automatizadas (plan de mantenimiento) con el fin de prevenir un siniestro de la totalidad de la aplicación.
3. Utilizar el mejor equipamiento para la implementación de la aplicación y así obtener un mejor funcionamiento del mismo.
4. Se recomienda realizar investigaciones sobre otras tecnologías de Garment Transfers, que conlleven a mejorar la transferencia de prendas en este tipo de aplicación.
5. Como toda evolución tecnológica, por lo frecuente, conlleva a una significativa disminución de los recursos humanos dentro de las organizaciones, ya que los procesos se automatizan, los tiempos disminuyen y los recursos se ahorran, en el corto plazo. Es por ello que se propone a la institución que la reducción del personal planteada en el análisis costo beneficio, no se realice de forma estricta, sino más bien se haga una redistribución del mismo en áreas que lo necesiten

Referencias Bibliográficas

- ABREGO, M. (23 de 10 de 2012). *Mejores prácticas para el diseño de interacción Kinect – Usuario*. Obtenido de <https://malenyabrego.wordpress.com/2012/10/23/mejores-practicas-para-el-diseno-de-interaccion-kinect-usuario/>
- Aden Hepburn, G. (2012). *the Augmented Reality Shoe Store*. Obtenido de <http://www.digitalbuzzblog.com/goertz-augmented-reality-virtual-shoe-fitting-store-installation/>
- Alegsa. (30 de 09 de 2013). *QUE ES LIBRERIA*. Obtenido de <http://www.alegsa.com.ar/Dic/biblioteca.php>
- Alegsa. (30 de 09 de 2013). *QUE ES UNA API?* Obtenido de <http://www.alegsa.com.ar/Dic/api.php>
- Alegsa.Screenshoot. (30 de 09 de 2013). *QUE ES SDK?* Obtenido de <http://www.alegsa.com.ar/Dic/sdk.php>
- Álvarez, M. A. (01 de 01 de 2001). *Qué es HTML*. Obtenido de <http://www.desarrolloweb.com/articulos/que-es-html.html>
- Asbanc, A. d. (24 de 03 de 2015). *E-commerce en Perú: Compras online se disparan*. Obtenido de <http://blog.webtilia.com/e-commerce-en-peru-compras-online/>
- Aspiazu, G. C. (25 de 01 de 2010). *Modelos desarrollo de software*. Obtenido de <http://menteerrabunda.blogspot.com/2010/03/modelos-desarrollo-de-software.html>
- AZUMA, R. e. (2011). *Recent advances in augmented reality*. . Obtenido de Disponible en <http://www.intechopen.com/books/show/title/augmented-reality>
- BABEL. (2014). *BABEL*. Obtenido de <http://es.babbel.com/>
- Barfield, W. &. (2011). *Fundamentos de Informática usable y Realidad Aumentada*. Mahwah, NJ: Lawrence Erlbaum.
- Bautista, E. P. (08 de 12 de 2011). *Java Server Pages*. Obtenido de <http://aplicaciones-web-lenguajes-programaci.blogspot.com/2011/12/jsp.html>
- Bennett, K. H. (2010). *Software maintenance and evolution: a roadmap*. In Proceedings of the Conference on The Future of Software Engineering. Limerick, Ireland. ACM., España.
- BIMBER, O. y. (s.f.). *Spatial Augmented Reality –* . Obtenido de <http://www.intechopen.com/books/show/title/augmented-reality>
- Binstock, A. (12 de 11 de 2014). *C# and .NET's Sudden Ubiquity*. Obtenido de <http://www.drdoobs.com/windows/c-and-nets-sudden-ubiquity/240169282>
- Blázquez, S. (02 de 09 de 2014). *El mercado de realidad aumentada se multiplicará por 750 en cinco años*. Obtenido de <http://www.media-tics.com/noticia/721/dircom-2.0/el-mercado-de-realidad-aumentada-se-multiplicara-por-750-en-cinco-anos.html>
- Blender. (2011). *Blender*. Obtenido de <https://www.blender.org/download/>
- BOLADO, J. S. (2011). *INFORME DE REALIDAD AUMENTADA Y EDUCACIÓN 3.0*. Lima - Peru: La republica.
- Bong, S. (2012). *ArtoolWork*. Obtenido de www.artoolworks.com/products/desk-top/nyartoolkit

- Caudell, T. P. (1992). Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. En *International Conference on Systems Sciences* (págs. 659-669). Hawaii.
- Chavesta Velásquez, W. (2013). *Implementación de una aplicación móvil con Realidad Aumentada basada en plataforma Android como apoyo a la promoción del Turismo en el Departamento de Lambayeque*. Chiclayo: Universidad Señor de Sipán .
- COUNCIL, B. (2014). *BRITISH COUNCIL*. Obtenido de <http://www.britishcouncil.org/>
- Crio, P. (2010). Obtenido de Proceso: <http://www.monografias.com/trabajos14/administ-procesos/administ-procesos.shtml>
- criscobo. (12 de 01 de 2011). *progpracticasc*. Obtenido de http://progpracticasc.blogspot.com/2011_01_01_archive.html
- Cruz, W. F. (2014). *ISO 27005*. Obtenido de <http://seguriedades.com/index.php/metodologias-de-riesgos/iso-27005>
- Dept, H. E. (2012). *Kinect*. Obtenido de <https://es.scribd.com/doc/86437713/kinectdoc>
- Duncan, G. (28 de 06 de 2012). *Jnect for Eclipse - The Java to Kinect for Windows SDK bridge*. Obtenido de <http://channel9.msdn.com/coding4fun/kinect/Jnect-for-Eclipse-The-Java-to-Kinect-for-Windows-SDK-bridge>
- Eduardo, M. (07 de 02 de 2003). *alzado.org*. Obtenido de http://www.alzado.org/articulo.php?id_art=40
- Estanteriasmoro. (2014). *Detalle del Producto*. Obtenido de http://www.estanteriasmoron.com.ar/desarrollo_producto.php?pro_id=375
- Esteban, M. d. (30 de 10 de 2013). *roggerocorp*. Obtenido de <http://www.roggerocorp.com/impacto-de-metodologias-agiles-scrum-kanban-en-las-organizaciones/>
- Estévez, Á. R. (27 de 7 de 2013). *Sistema de entrenamiento con Kinect*. Barcelona - España. Obtenido de UN ESPEJO DE MAQUILLAJE VIRTUAL GRACIAS A LA REALIDAD AUMENTADA: http://www.brainstorm9.com.br/30910/advertising/natura-espelho-de-maquagem-virtual/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+brainstorm9+%28Brainstorm9%29
- Etronika, C. I. (23 de 03 de 2012). *Kinect Banking App*. Obtenido de <http://www.psfk.com/2011/10/new-lego-game-uses-ios-app-to-combine-digital-and-physical-play.html>
- fergarciac*. (25 de enero de 2013). Obtenido de Obtenido de: <https://fergarciac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>
- Fernández Sánchez, L. &. (2007). *Gestión del riesgo en la fase de ingeniería de requisitos de un proyecto software*. Argentina: El Cid Editor - Informática.
- FIGUEROA, J. C. (2013). "EFECTO DEL USO DEL SENSOR KINECT PARA MEJORAR LA ATENCIÓN Y CONCENTRACIÓN EN LOS NIÑOS DEL TERCER GRADO "A" DE LA INSTITUCIÓN DIVINO MAESTRO –

- MOLLEPAMPA, CAJAMARCA, 2013". CAJAMARCA:UNIVERSIDAD PRIVADA DEL NORTE.
- FOMBONA, J., & GOULAO, M. d. (2012). *Using Augmented Reality and m-learning to optimize students performance in Higher*. Obtenido de <http://www.sciencedirect.com>
- Garcia, A. (2012). *netwnews.blogspot*. Obtenido de <http://netwnews.blogspot.com/2012/05/la-realidad-aumentada-con-kinect-y-una.html>
- geneura. (2008). *Qué es Apache*. Obtenido de <http://geneura.ugr.es/~gustavo/apache/>
- Geoffrey POON, Y. Y.-M. (2014). *A Consumer Level 3D Object Scanning Device using Kinect for Web-based C2C Business*. Hong Kong: Caritas Institute of Higher Education.
- Gonzalo. (31 de marzo de 2010). *¿Que es 3D Studio Max?* Obtenido de <http://blogs.ua.es/gonzalo/2010/03/31/%C2%BFque-es-3d-studio-max/>
- Gracia, L. M. (9 de Octubre de 2013). *OPEN CV*. Obtenido de unpocodejava.wordpress.com/2013/10/09/que-es-opencv/
- Guevara, Z. Q. (09 de 09 de 2014). *Comercio Electronico / e-commerce*. Obtenido de <http://es.slideshare.net/ZQG1987/comercio-electrnico-e-commerce>
- Herrera, G. (2013). *QUÉ ES GLASSFISH*. Obtenido de <https://es.scribd.com/doc/148219808/QUE-ES-GLASSFISH>
- HUELVA, S. (6 de 6 de 2014). *Premio en Silicon Valley para la realidad aumentada de Seabery*. Obtenido de <http://www.huelvainformacion.es/article/huelva/1788527/premio/silicon/valley/para/la/realidad/aumentada/seabery.html>
- Inc, G. (2013). *Google Glass*. Obtenido de [Google Glass](http://www.google.com/glass/start/). Obtenido de <http://www.google.com/glass/start/>
- ISSI, G. (2003). *Ingeniería del Software y Sistemas de Información Colaboradore*. Valencia: Universidad politécnica de Valencia. Taller realizado en el marco de las VIII jornadas de Ingeniería del Software y Bases de Datos.
- Ivar, J., Grady, B., & James, R. (1999). *The Unified Software Development Process*.
- Jesús, B. (15 de Noviembre de 2010). *Imagen Kinect for Xbox 360, partes que conforman el Kinect*. Obtenido de <http://noticias.softonic.com/noticia-ya-hay-mas-de-un-millon-de-kinect-vendidos-en-el-mundo-kinect-xbox360>
- JOHANNA, L. (24 de octubre de 2012). *Realidad Aumentada*. Obtenido de <http://realidadaumentadauniminuto8.blogspot.com/2012/10/tecnicas-de-visualizacion.html>
- José Luis Sierra, A. F.-V. (15 de Febrero de 2005). *From Chasqui to Chasqui II: an Evolution in the Conceptualization of Virtual Objects*. España. Obtenido de <http://www.ordenadores-y-portatiles.com/objetos-virtuales.html>
- Julius. (28 de 09 de 2008). *DIFERENCIAS PHP, JSP Y ASP*. Obtenido de <http://informetics.blogspot.com/2008/09/diferencias-php-jsp-y-asp.html>

- Karina Melendez, A. D. (2009). *Normas de la Calidad del Producto Software*. Lima.
- Kato, H. &. (1999). Marker tracking and HMD calibration for a video-based augmented reality conferencing system.
- Kinect. (29 de 08 de 2014). *Xbox 360 + Kinect*. Obtenido de <http://paraqueleasunpoco.crearblog.com/?p=1283>
- KIRKPATRICK. (6 de 2009). *elogos.es*. Obtenido de www.elogos.es/.../2009-06-Medida-del-impacto-de-la-formacion.pdf
- Klein, G. a. (2007). *Parallel Tracking and Mapping for Small AR, 6th IEEE and ACM international Symposium on Mixed and Augmented Reality. Symposium on Mixed and Augmented Reality. IEEE Computer Society*,. Washington, DC, 1-10.
- Lab, A. E. (01 de 04 de 2010). *Augmented Environments Lab*. Obtenido de <http://ael.gatech.edu/lab/>
- Landerpfc. (2012). *Freenect, openkinect, openNI*. Obtenido de <https://landerpfc.wordpress.com/2011/02/18/kinect-freenect-openkinect-openni/>
- LARIOSZH, A. (2011). *TECNOLOGÍA DE LA INFORMACIÓN Y LA COMUNICACIÓN EN EL PROCESO ENSEÑANZA APRENDIZAJE*. Obtenido de <http://aleelarioszh.blogspot.com/2011/02/tecnologia-de-la-informacion-y-la.html>
- Leder Rivas Andrés, F. S. (2011). *eCommerce*.
- Lee San Hyuk, O. S. (2013). *Kinect sensor based Pc control interface for handicapped users*. Corea del Sur: Universidad de Dongguk.
- LOOSER, J. G. (2009). Obtenido de www.artoolworks.com/community/osgart/
- López, A. E. (15 de 5 de 2013). *DESARROLLO DE UN PROTOTIPO DE UN API PARA LA INTERACCIÓN DE UN USUARIO CON APLICACIONES CON CONTENIDO 3D UTILIZANDO KINECT*. Guayaquil - Ecuador. Obtenido de Topshop hace realidad los probadores de realidad aumentada de la mano de Kinect: <http://www.marketingdirecto.com/actualidad/anunciantes/topshop-hace-realidad-los-probadores-de-realidad-aumentada-de-la-mano-de-kinect/>
- López, D. (2014). *Katherine Lisset*. Chiclayo: Universidad Católica Santo Toribio de Mogrovejo.
- López, D. K. (2014). *Realidad aumentada con kinect para contribuir en el proceso de terapoas de los pacientes con parálisis cerebral infantil*. Chiclayo:: Universidad Católica Santo Toribio de Mogrovejo.
- Luis, Á. J. (24 de 10 de 2014). *Fantasmas digitales*. Obtenido de <http://www.elmundo.es/blogs/elmundo/el-gadgetoblog/2011/10/24/fantasmas-digitales.html>
- Mark Otto, J. T. (2014). *Bootstrap 3*. Obtenido de https://librosweb.es/libro/bootstrap_3/
- Martinez, L. R. (20 de 09 de 2012). *VITRINAS INTERACTIVAS KINECT Y PROBADOR VIRTUAL*. Obtenido de <https://emedesing.wordpress.com/category/airkinect-2/page/2/>
- Martinez, R. (02 de 10 de 2010). *PostgreSQL*. Obtenido de http://www.postgresql.org.es/sobre_postgresql

- Mentoring, G. (26 de 01 de 2012). *Qué es un IDE*. Obtenido de <http://globalmentoring.com.mx/cursos-java/java-fundamentos/que-es-un-ide/>
- Microsoft. (08 de 10 de 2012). *Kinect for Windows Language Packs v11.0*. Obtenido de <http://www.microsoft.com/en-us/download/details.aspx?id=34809>
- Microsoft. (2012). *Visual Studio*. Obtenido de <http://msdn.microsoft.com/es-es/library/dd831853.aspx>
- Microsoft, A. (2007). *Qué es un control ActiveX*. Obtenido de <https://www.microsoft.com/es-es/security/resources/activex-what-is.aspx>
- Microsoft. (08 de 08 de 2007). *Cómo utilizar VBScript de cliente con secuencias de comandos remotas*. Obtenido de <https://support.microsoft.com/es-es/kb/229661/es>
- MIKOLUK, K. (12 de 12 de 2013). *JQuery vs. JavaScript: ¿Cuál es la Diferencia En Definitiva?* Obtenido de <https://blog.udemy.com/jquery-vs-javascript-2-cual-es-la-diferencia-en-definitiva/>
- Minayo, J. A. (04 de 04 de 2011). *Base De Datos*. Obtenido de http://jesicaandreaminayo.blogspot.com/2011_04_01_archive.html
- Mocholi, A. (15 de 10 de 2014). *Desarrollar aplicaciones móviles de Realidad Aumentada*. Obtenido de <http://www.yeeply.com/blog/desarrollar-aplicaciones-moviles-de-realidad-aumentada/>
- Möhring, M. L. (2004). Video See-Through AR on Consumer Cell Phones. En *international Symposium on Mixed and Augmented Reality* (págs. 252-253).
- Mohta, A. (28 de 02 de 2012). *Kinect Enabled Shopping Cart*. Obtenido de <http://www.wpxbox.com/kinect-enabled-shopping-cart/>
- msdn.microsoft. (2012). *Kinect for Windows*. Obtenido de <http://msdn.microsoft.com/en-us/library/hh855347.aspx>
- Mundo, B. (23 de 01 de 2015). *Holo Lens, el dispositivo que permite crear y tocar hologramas en tu casa*. Obtenido de http://www.bbc.co.uk/mundo/noticias/2015/01/150123_tecnologia_holo_lens_microsoft_interactivo_amv
- Neria, G. M. (2014). *DESARROLLAR UNA APLICACION BASADA EN REALIDAD AUMENTADA COMO MATERIAL DE APRENDIZAJE DEL CUERPO HUMANO CON TECNOLOGIA MOVIL ANDROID*. Chiclayo - Peru.
- NetBeans. (10 de 12 de 2014). *NetBeans IDE 8.0.2 Information*. Obtenido de <https://netbeans.org/community/releases/80/>
- OLMEDO, H. y. (2012). *Augmented Reality To Mixed Reality: Technological Options*.
- Ortega, S. H. (26 de MARZO de 2013). *Características Microsoft SQL Server*. Obtenido de <http://es.slideshare.net/CrypticHernandezOrtega/caracteristicas-microsoft-sql-server>
- Parrado, N. (12 de Noviemnre de 2012). *La importancia del probador en la tienda*. Obtenido de <http://jacintollorca.com/blog/blog/la-importancia-del-probador-en-la-tienda/>

- Pressman, R. S. (2002). *Ingeniería de Software un enfoque practico*. España: McGraw-Hill - ISBN: 8448132149.
- Rancan, C. (2003). *Gestión de Configuración de productos software en etapa de desarrollo. Trabajo final de la Especialidad en Control y Gestión de Software*. ITBA.
- Research.microsoft. (2011). *KINECT SDK*. Obtenido de <http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/about.aspx>
- Rincón Mahecha, O. F. (19 de Agosto de 2011). *Slideshare*. Obtenido de Obtenido de: http://es.slideshare.net/FALKO890317/que-son-ambientesyobjetosvirtualesdeaprendizaje-omar-ricon?qid=c5d1e1c5-00c4-4a03-8b12-78a7b0b13971&v=default&b=&from_search=1
- Rodríguez, C. A. (2013). *ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE COMERCIO ELECTRÓNICO INTEGRADO CON UNA APLICACIÓN MÓVIL PARA LA RESERVA Y VENTA DE PASAJES DE UNA EMPRESA DE TRANSPORTE INTERPROVINCIAL*. LIMA: PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ.
- RODRÍGUEZ, J. P. (2011). *Realidad Aumentada para el Aprendizaje de Ciencias en niños de Educación General Básica. Tesis de Grado Ingeniero Civil en Computación*. Santiago de Chile Universidad de Chile. : Facultad de Ciencias Físicas y Matemáticas. Departamento de Ciencias de la Computación.
- Rosales, R. J. (5 de 10 de 2010). *rolandojaldin*. Obtenido de <http://rolandojaldin.blogspot.com/2010/10/introduccion-la-metodologia-rup-proceso.html>
- Sabia. (2011). *Imagen Kinect, partes internas y externas del kinect*. Obtenido de <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/VisionArtificial/iteracion.html>
- Schach, S. R. (29 de 08 de 2009). *Ingeniería de Software Clásica y Orientada a Objetos* Sexta Edición. Mexico: MCGRAW-HILL.
- Schwaber, K., & Beedle, M. (2006). *Agile Software Development with SCRUM*. Conchango: ISBN: 0130676349.
- Serrano Mamolar, A. (2012). *Herramientas de desarrollo libres para aplicaciones de Realidad Aumentada con Android. Análisis comparativo entre ellas. Trabajo fin de Máster Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital*. Universidad Politécnica de Valencia.
- SHARP, J. (2009). *VISUAL C# 2008 (PASO A PASO)*. ANAYA MULTIMEDIA, ISBN: 9788441524491.
- Solis, J. (26 de 09 de 2014). *¿QUÉ ES BOOTSTRAP Y CÓMO FUNCIONA EN EL DISEÑO WEB?* Obtenido de <http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>
- Solorio, M. (16 de 04 de 2013). *Project-IS*. Obtenido de <http://metodologiaencascada.blogspot.com/>
- subgurim.net. (07 de Febrero de 2006). *Por qué C#*. Obtenido de <http://www.subgurim.net/Articulos/csharp/5/por-que-csharp.aspx>

- Takeuchi, H., & Nonaka, I. (1986). The New New Product Development Game. *Harvard Business Review*. 137 - 146. Obtenido de http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADAs_SC_RUM_y_XP
- TANZ, J. (2011). *Composición de hardware de Kinect*. Obtenido de http://www.wired.com/2011/06/mf_kinect/2/
- Tapia, C. (7 de 10 de 2010). *QUE ES EL KINECT*. Obtenido de <http://tecnologiapor1000.blogspot.com/2010/10/que-es-el-kinect.html>
- Technology, N. R. (2000). *BARS: Battlefield Augmented Reality System*. telecos.cnmc.es. (2011). *Informe e-commerce*.
- TELEFÓNICA. (2011). *RA: Una nueva lente para ver el mundo*. Barcelona-España: Ariel, S.A.
- Thonburi, K. M. (2012). *INTERNATIONAL CONFERENCE ON COMPUTER AND COMMUNICATION TECHNOLOGIES*. Thailand.
- TILATAM. (2010). *lenguajes de programación: PHP, HTML, XHTML, SQL, XML, JavaScript, AJAX, CSS*. Obtenido de <http://joomla-chile.cl/diplomado-u-de-chile-joomla-chile/unidad-i/135-joomla-y-sus-lenguajes-de-programacion-php-html-xhtml-sql-xml-javascript-ajax-css.html>
- Valdes, A. S. (24 de 11 de 2013). *Visión por computadora*. Obtenido de <http://graficacionito.blogspot.com/2013/11/52-vision-por-computadora.html>
- Valdés, D. P. (03 de 07 de 2007). *¿Qué es Javascript?* Obtenido de <http://www.maestrosdelweb.com/que-es-javascript/>
- Valencia, U. P. (04 de 2006). *PROGRAMACION EXTREMA XP*. Obtenido de <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Vaquero, M. (2014). *¿Qué es un Applet Java?* Obtenido de <http://www.deciencias.net/simulaciones/paginas/appletjava.htm>
- villalobos, R. m. (09 de 07 de 2014). *Fundamentos de programación C++*. Obtenido de http://es.slideshare.net/Avallejos_/ricardo-marcelo-villalobos-c
- windows, x. (07 de 10 de 2013). *La evolución de Kinect y la importancia real de Microsoft Research*. Obtenido de <http://www.xatakawindows.com/xbox/la-evolucion-de-kinect-y-la-importancia-de-microsoft-research>
- WION, N. (15 de Diciembre de 2013). *Realidad Aumentada*. Obtenido de <http://www.realidad-aumentada.eu/author/wionadmin02/>
- Wouter. (27 de 06 de 2012). *AIRKinect*. Obtenido de <http://www.as3nui.com/airkinect-2-2-now-with-open-source-native-code/>
- X. Basogain, M. O. (2013). *Realidad Aumentada en la Educación: una tecnología emergente*. Bilbao, España.: Escuela Superior de Ingeniería de Bilbao, EHU.
- xbox. (20 de 07 de 2012). *Kinect sensor for Xbox 360 components*. Obtenido de <http://support.xbox.com/en-US/xbox-360/kinect/kinect-sensor-components>

ANEXO 1: MATRIZ DE CONSISTENCIA

TÍTULO	PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES/ INDICADORES	INSTRUMENTOS DE MEDICIÓN	POBLACIÓN Y MUESTRA
<p>PROPUESTA DE UNA SOLUCIÓN DE REALIDAD AUMENTADA BASADA EN TECNOLOGIA KINECT PARA LA INTERACCIÓN EN EL PROCESO DE PRUEBA DE PRENDAS DE VESTIR</p>	<p>¿Cómo desarrollar una solución que permita apoyar el proceso de interacción de prendas de prueba de vestir?</p>	<p>GENERAL.- Desarrollar una solución de Probador Virtual basada en tecnología Kinect para apoyar al proceso de interacción de prueba de prendas de vestir.</p> <p>ESPECIFICOS</p> <p>1. Caracterizar los gestos que se usaran para la comunicación.</p>	<p>Mediante la Solución de un probador virtual basada en tecnología Kinect apoyara en la interacción de interfaz natural a la prueba de prendas de vestir.</p>	<p>INDEPENDIENTE.- Solución de un probador virtual basada en tecnología Kinect.</p> <p>INDICADORES -Usabilidad -Grado de Confiabilidad del Producto</p>	<p>- Plantilla de evaluación de Uso</p> <p>- Plantilla de evaluación de software</p>	<p>Población: La población está determinada por el número total de prendas de vestir que se incluirá en el sistema que hacen un total de 200 imágenes</p> <p>Muestra: La muestra es poblacional ya que se realizará las pruebas con la totalidad de las imágenes que se encuentran en el</p>

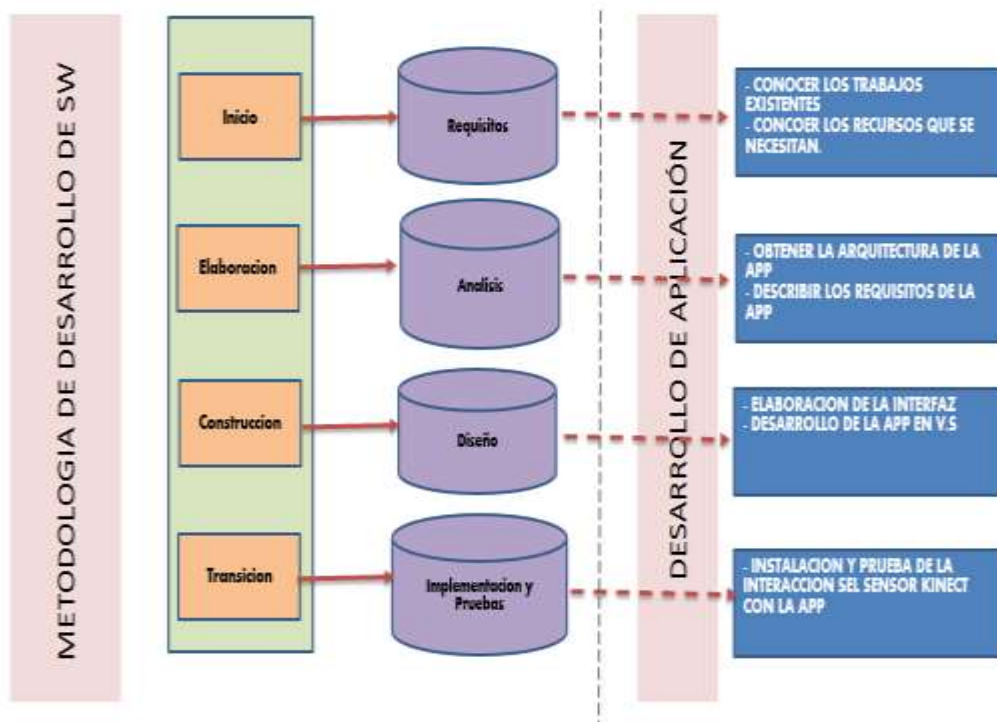


		<ol style="list-style-type: none"> 2. Desarrollar una API bajo la tecnología Kinect, intuitiva y rápida en la NUI. 3. Desplegar la API en un sistema web. 4. Analizar resultados. 		<p>DEPENDIENTE</p> <p>Interacción de interfaz natural</p> <p>INDICADORES</p> <ul style="list-style-type: none"> - Precisión - Usabilidad 	<p>- Ficha de evaluación de uso</p>	<p>sistema.</p>
--	--	--	--	--	-------------------------------------	-----------------



ANEXO 2: ESTRUCTURA DE LA PROPUESTA DE SOLUCIÓN.

Esta aproximación de metodología se traduce en un formato por 4 fases que explica explícitamente el proceso, Cada proceso recibe en tiempo real, esto consigue la información que necesita la aplicación.



ANEXO 3 FICHA DE USABILIDAD - POST TEST

1.- Es una interfaz limpia, sin ruido visual

- a) Si
- b) No

Respuesta:

2.- Uso correcto del espacio visual de la API

- a) Si
- b) No

Respuesta:

3.- Uso consistente de los controles de la interfaz

- a) Si
- b) No

Respuesta:

4.- Se emplea un gestos claros y concisos

- a) Si
- b) No

Respuesta:

5.- Interacción amigable, familiar y cercano

- a) Si
- b) No

Respuesta:

6.- Interacción de gestos comprensibles por cualquier usuario

- a) Si
- b) No

Respuesta:

7.- El usuario tiene todo el control sobre la interfaz

- a) Si
- b) No

Respuesta:

8.- Se ha controlado el tiempo de respuesta

- a) Si
- b) No

Respuesta:

ANEXO 4 PLANTILLA DE EVALUACION DE USO

Tabla 4. 6: Resultado en tablas y gráficos del Software Variable dependiente. Fuente: (Pressman, 2002)

Plantilla de evaluación de Uso

Dimensiones	Indicadores	Deficiente		Regular		Bien		Excelente	
		F	%	F	%	F	%	F	%
Eficacia	Al utilizar varios objetos la API fue:								

Dimensiones	Indicadores	Deficiente		Regular		Bien		Excelente	
		F	%	F	%	F	%	F	%
Productividad	El tiempo que se usó para completar la tarea fue:								
	El esfuerzo del usuario para completar la tareas fue:								

Dimensiones	Indicadores	Bajo		Alta		Bien		Excelente	
		F	%	F	%	F	%	F	%
Satisfacción	La satisfacción que el usuario siente al usar la API fue:								



**ANEXO 5 VALIDACIÓN DEL INSTRUMENTO DE RECOLECCIÓN
DE DATOS - FICHA DE USABILIDAD - POST TEST**

**Tema de Tesis: PROPUESTA DE UNA SOLUCIÓN DE REALIDAD
AUMENTADA BASADA EN TECNOLOGIA KINECT PARA LA
INTERACCIÓN EN EL PROCESO DE PRUEBA DE PRENDAS DE
VESTIR.**

Autor: José C. Inoñan Morales

ITEM (En función de cuántas preguntas tenga el instrumentos)	A) Correspondencia de las preguntas con los objetivos y la problemática planteada P= Pertinente NP = No pertinente		B) Calidad técnica y representativa O= Óptima B= Buena R= Regular D= Deficiente				C) Coherencia en el Lenguaje A= Adecuado I = Inadecuado		OBSERVACIONES
	P	NP	O	B	R	D	A	I	
1									
2									
3									
4									
5									
6									
7									
8									
DATOS DEL EVALUADOR			Nombre:						
			Profesión						
			Fecha						



ANEXO 6 VALIDACIÓN DEL INSTRUMENTO DE RECOLECCIÓN DE DATOS - PLANTILLA DE EVALUACION DE USO

Tema de Tesis: PROPUESTA DE UNA SOLUCIÓN DE REALIDAD AUMENTADA BASADA EN TECNOLOGIA KINECT PARA LA INTERACCIÓN EN EL PROCESO DE PRUEBA DE PRENDAS DE VESTIR.

Autor: José C. Inoñan Morales

INDICADOR	CRITERIOS	Deficiente 1	Baja 2	Regular 3	Buena 4	Muy buena 5
1 CLARIDAD	Esta formulada con lenguaje apropiado y comprensible					
2 OBJECTIVIDAD	Permite medir hechos reales					
3 ORGANIZACIÓN	Adecuado al avance a la ciencia y tecnología					
4 SUFICIENCIA	Comprende aspectos de las variables en cantidad y calidad suficiente					
6 PERTIENENCIA	Permite conseguir datos acuerdo a los objetivos y la problemática planteada.					
7 CONSISTENCIA	Pretende conseguir datos basados en teorías o modelos teóricos					
8 COHERENCIA	Entre variables, indicadores y los ítems					
9 METODOLOGIA	La estrategia responde al propósito de la investigación					
10 APLICACIÓN	Los datos permiten un tratamiento estadístico pertinente					
DATOS DEL EVALUADOR		Nombre:				
		Profesión				
		Fecha				

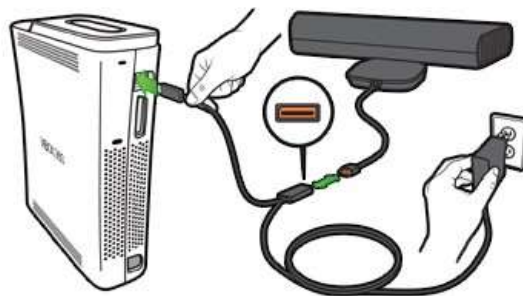


ANEXO 7 MANUAL DE USUARIO

1.1. Manual de Usuario

Implementación del Software

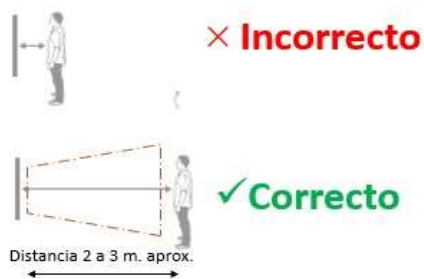
- a) Requerimientos de hardware:
 - Computadora
 - 4gb de RAM
 - 500 GB HDD
 - Puerto USB 3.0 Dedicado
 - Sensor *Kinect* 2.0
 - Cable USB de alimentación
- b) Requerimientos de Software:
 - Direct SDK 9.0



- *Kinect* SDK 2.9 Figura . 1 Conexión del Kinect a PC
- Microsoft XNA Game Studio 4.0

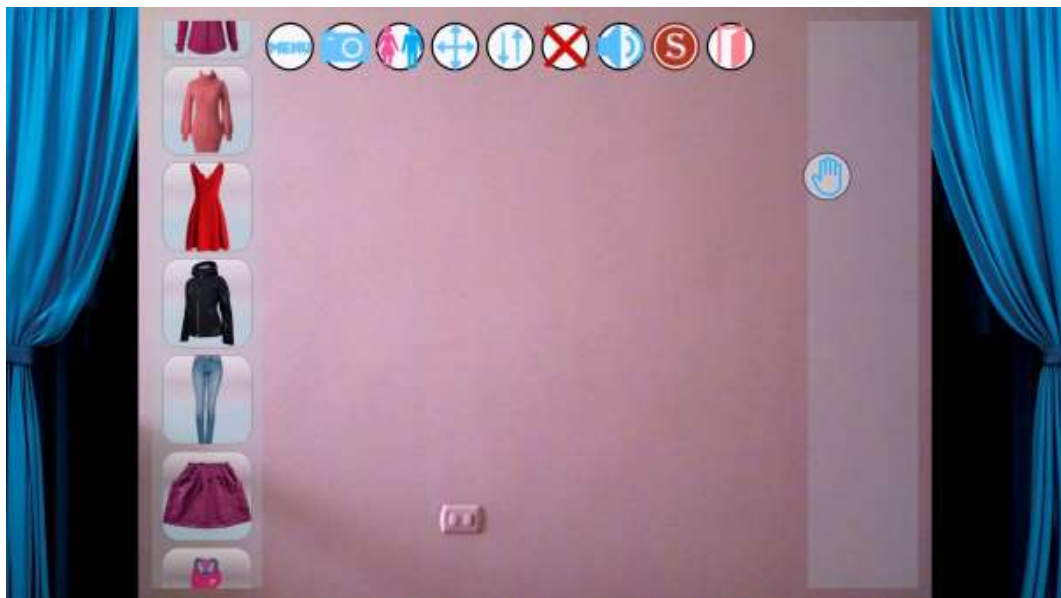
Acceso al Sistema

- a) Ejecute la aplicación.
- b) Conecte el sensor *Kinect* a la PC
- c) Colóquese a una distancia de tal forma que pueda verse completamente en la pantalla (2 a 3 m aprox. del sensor *Kinect*).



Iniciar la aplicación

- d) Enseguida usted verá una pantalla similar con la cual podrá interactuar con las prendas de vestir (Si no se visualiza esta pantalla, revise la sección de error de *Kinect de este manual*).



Visualización de la aplicación

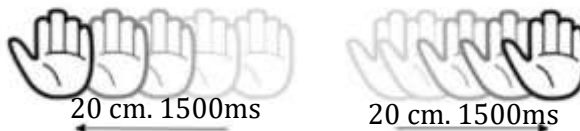
Deslizar la prenda de vestir

- a) En la parte de dercha o izquierda usted puede ver un menú de prendas de vestir disponibles, de las cuales usted puede elegir la de su preferencia.



- b) Usted debe mover su mano derecha en línea recta:

- Hacia la izquierda para elegir la pieza a la izquierda.
- Hacia la derecha para elegir pieza a la derecha.



a) Deslizar arriba – abajo b) Deslizar abajo - arriba

Figura 8 Deslizar prenda

Selección de la prenda

El gesto de seleccion cuando usted tiene alineadas la mano en el inoco de la prenda o menú y este se selecciona.



Manos para seleccionar la prenda

ANEXO 8 FOTOS DE PRUEBA DE PRENDAS DE VESTIR



ANEXO 9 INFORME TÉCNICO APLICACIÓN PROBADOR VIRTUAL

I. Algoritmo de Gestos con Realidad Aumentada

Seudocódigo:

- *Seguimiento de la Mano.*
 - ✓ Iniciar la aplicación de Probador Virtual
 - ✓ Ejecuta e inicializa los servicios de Kinect además de diseño de la aplicación.
 - ✓ Crea la Instancia de HandTracking (Seguimiento de Mano, la cual contiene todas las funciones a utilizar).
 - ✓ Encontrar los sensores de Kinect conectados.
 - ✓ Inicializa el sensor Kinect.
 - ✓ Activar las múltiples fuentes de datos.
 - ✓ Una vez activado las múltiples fuentes de datos, uno de ellos es referencia al marco del cuerpo (BodyFrameReference), esta fuente de dato captura los puntos del cuerpo.
 - ✓ La operación de usar siempre se utilizara siempre y cuando está activo el Kinect.
 - ✓ Usar (fuente de referencia del cuerpo y adquirirlo)
 - Si la fuente es diferente a nulo entonces
 - Captura la data del cuerpo con los puntos
 - Refresha los nuevos puntos del cuerpo
 - Si no hay data del cuerpo entonces
 - Mostrar mensaje que no se ha detectado cuerpo
 - Fin Si
 - Actualizar la posición de la mano con los parámetros de la data de Kinect
 - Actualiza el diseño de la imagen a mostrar de la mano en el Kinect.
 - Fin Si
 - Fin de Usar.
 - ✓ Una vez capturado los la data del cuerpo para capturar los puntos que nos ofrece el Kinect (Joints), estos puntos de los 25, escogeremos el de la mano derecha y mano izquierda, para seguir la mano especifica pasaremos los parámetros mencionados y también la variable sensor del Kinect y la posición de alto y ancho .(Este método se encuentra en un objeto Hand Tracking)
 - ✓ Una vez capturado los puntos de la mano derecha y mano izquierda con la variable del sensor con las variables de altura y

ancho. Se visualiza en este seudocódigo de seguimiento de la mano.

Global Punto puntolzquierdo, puntoDerecho

Procedimiento SeguimientoMano (Punto Manolzquierda, Punto ManoDerecha, SensorKinect sensor, numero ancho, numero altura)

```

Si      Manoizquierda.EstadoSeguimiento      es      igual
EstadoSeguimiento.NoSeguido Y
      ManoDerecha.EstadoSeguimiento          es      igual
EstadoSeguimiento.NoSeguido
      Salir del procedimiento y no seguir en las instrucciones.

```

Fin Sí.

PuntoEspacioCamara puntol = Manolzquierda.Posicion

PuntoEspacioProfundidad

izquierdaPunto=sensor.MapeoDeCoordenadas.MapaCamaraPuntoDesdeEspacioProfundidad (puntol)

Entero izquierdaY=Entero (izquierdaPunto.Y*altura/sensor.FuenteProfundidad.DescripcionFuente.Altura)

Entero izquierdaX=Entero (izquierdaPunto.X*ancho/sensor.FuenteProfundidad.DescripcionFuente.Ancho)

PuntoEspacioCamara puntoD = ManoDerecha.Posicion

PuntoEspacioProfundidad

derechaPunto=sensor.MapeoDeCoordenadas.MapaCamaraPuntoDesdeEspacioProfundidad (puntoD)

Entero derechaY=Entero (derechaPunto.Y*altura/sensor.FuenteProfundidad.DescripcionFuente.Altura)

Entero derechaX=Entero (derechaPunto.X*ancho/sensor.FuenteProfundidad.DescripcionFuente.Ancho)

puntolzquierdo=crear Punto (izquierdaX, izquierdaY)

puntoDerecho=crear Punto (derechaX, derechaY)

Fin Procedimiento

- ✓ Este seudocódigo nos indica que para seguir la mano tiene que preguntar si el punto de Kinect que este caso es de izquierda y derecha está en estado de no seguido si es así entonces no hace nada y sale del procedimientos.
- ✓ Luego el Kinect nos ofrece un hardware para capturar la profundidad y con el API del Kinect capturamos el punto, la posición del punto de la mano izquierda y así tenemos la posición en el espacio X,Y,Z de la cámara
- ✓ Así se lograra captar el punto de la cámara con el punto de la profundidad mediante el mapeo de coordenadas desde el espacio XYZ y alinearlo con la profundidad para así obtener el verdadero punto X e Y en la aplicación (en la ventana).
- ✓ La misma lógica tiene en la mano derecha



- ✓ Se asigna los valores verdaderos de los puntos de la mano izquierda y derecha.
- ✓ Cada acción modificada o activada la propiedad de cambio se envía los puntos verdaderos a la ventana de la aplicación que en este caso es tecnología (XAML)
- *Selección de una mano con un botón.*
 - ✓ Click en el Boton o colocar la mano sobre el botón indica primero ejecuta un evento de HandCursorClick pero esto nos indica que ya esta seguido la posición de la mano izquierda o derecha y estas posiciones esta de forma global del diseño .
 - ✓ Comienza el tiempo de sonido para dar el toque de click que ejerce sobre el botón.
 - ✓ Cuando Se ejerce da el efecto de levantamiento.
 - ✓ Se activa el evento de entrar en ejecucion al botón (Enter)
 - ✓ Se activa el evento de pasar de lado del Boton(Move)
 - ✓ Dejar de soltar el botón (Leave)
 - ✓ Una vez que sono el click entonces se ejecuta la acción que sea programado para funcionalidad por ejemplo en la aplicación de seleccionar la categoría de Ropa.
- *Scroll Horizontal y Vertical con la Mano.*
 - ✓ Al arrastrar el Scroll de ítem de prendas de vestir o su categorías consiste en mover la mano hacia la arriba o hacia abajo.
 - ✓ Una vez que se cargó las posiciones de la mano derecha o izquierda gracias al seguimiento de la mano en el Scroll o áreas donde contiene botones solo debe aceptar una mano al interactuar.
 - ✓ Cargar las posiciones de Derecha o izquierda pero para interacción de los paneles de los botones, al momento de ejecutar la aplicación.
 - ✓ Activar los eventos y registrarlos como dependencias, cual indica que está en constante cambio.(Lo mismo en el ítem 2)
 - ✓ Modificación automático de las posiciones de las manos al momento de interactuar del panel. .(Lo mismo en el ítem 2)
 - ✓ Una vez que la posición de la mano este en el panel de botones se muestra el icono de la mano para así indicándonos que se puede seleccionar o interactuar con los botones. .(Lo mismo en el ítem 2).
 - ✓ Según este pseudocódigo que nos muestra la activación del icono de la mano y la única selección de una mano (sea derecha o izquierda).

```

Procedimiento ControladorManoEnMovimiento(Punto
manolzquierda,Punto manoDerecha)
    ManoCursor.Visible=Visible.Colapsado
    Variable elemento
=PanelBotonesCanvas.ActivacionEntradaPrueba(manolzquierda)
    Si grillaCerrar es visible entonces

        Elemento=grillaCerrar.ActivacionEntradaPrueba(manolzquierda
)
        Sino if (grillaPreguntarTomarFoto es visible) entonces
            Elemento=
grillaPreguntarTomarFoto.ActivacionEntradaPrueba(manolzquierda)
            Sino if (grillaRedesSociales es visible) entonces
                Elemento=
grillaRedesSociales.ActivacionEntradaPrueba(manolzquierda)
                Sino if (grillaTipoProducto es visible) entonces
                    Elemento=
grillaTipoProducto.ActivacionEntradaPrueba(manolzquierda)
                FinSi
                Variable mano=manolzquierda
                Si elemento no es un elementoUI entonces
                    Si grillaCerrar es visible entonces

                        Elemento=grillaCerrar.ActivacionEntradaPrueba(manolzquierda
)
                        Sino if (grillaPreguntarTomarFoto es visible) entonces
                            Elemento=
grillaPreguntarTomarFoto.ActivacionEntradaPrueba(manolzquierda)
                            Sino if (grillaRedesSociales es visible) entonces
                                Elemento=
grillaRedesSociales.ActivacionEntradaPrueba(manolzquierda)
                                Sino if (grillaTipoProducto es visible) entonces
                                    Elemento=
grillaTipoProducto.ActivacionEntradaPrueba(manolzquierda)
                                Sino
                                    elemento
=PanelBotonesCanvas.ActivacionEntradaPrueba(manolzquierda)
                                FinSi
                                Mano =ManoDerecha
                                Si elemento no es un elementoUI entonces
                                    Ejecutar evento de soltar /salir elemento
                                    Salir del procedimiento

                                Fin si
                                Fin Si
                                ManoCursor.Visible =Visible.Visible
                                Se asigna con canvas tamaño real en la pantallas
                                (ManoCursor,Mano.X-actualAncho/2);
                                Se asigna con canvas tamaño real en la pantallas
                                (ManoCursor,Mano.Y-actualAltura/2);
    
```


Se ejecuta evento del botón con la mano y el elemento seleccionado
FinProcedimiento.

- ✓ Una vez q aparece el icono de la mano entonces como el Scroll vertical que son de la categoría de la ropa y de las prendas de esa categoría, y el Scroll horizontal que es en la parte inferior para las fotos, se ejecuta los eventos que se activa al mover hacia arriba o abajo con animaciones de la misma tecnología
- *Interacción con voz*
 - ✓ Inicializar el Kinect
 - ✓ Abrir el Kinect ,activarlo
 - ✓ Cargar la ingeniería del habla pasándolo el sensor kinect
 - ✓ Cargar la gramatica del diccionario del habla ,y en el diccionario se reconoce el lenguaje correcto , en este caso que es español y que sea el driver correcto en este caso del kinect.
 - ✓ En la gramatica se asigna las opciones que aceptara la aplicación por ejemplo salir , se cierra la aplicación, polo se hace click en la categoría polos y te muestra en el panel de todos los polos.
 - ✓ Se activa el evento de reconocimiento de las palabras asignadas x ejmplo decir blusa se ejecuta la blusa como fuera darse click.
 - ✓ Se asigna la información del audio correcto.
 - ✓ Se llama prepara el sensor con su funcion AudioSource.
 - ✓ Se convierte el código de la fuente del audio del speech del mismo Microsoft y se asigna a la ingeniería del habla (speechengine) con un modo de reconocimiento multiple.

II. Código Fuente de los Gestos

Código Fuente:

- *Seguimiento de la Mano.*
 - ✓ Clase de KinectViewModelLoader

```
using System.Windows.Media.Imaging;
namespace appProbadorVirtual3D.ViewModel
{
    /// <summary>
    /// Loads Kinect view model
    /// </summary>
    public class KinectViewModelLoader
    {
        #region Private Fields
        /// <summary>
        /// Kinect view model
        /// </summary>
        static KinectViewModel _kinectViewModel;
        /// <summary>
        /// Kinect service
        /// </summary>
        static KinectService _kinectService;
        #endregion Private Fields
        #region Public Properties
        /// <summary>
        /// Gets the view model.
        /// </summary>
        public KinectViewModel KinectViewModel
        {
            get { return _kinectViewModel ??
                (_kinectViewModel = new KinectViewModel(_kinectService)); }
        }
        #endregion Public Properties
        #region .ctor
        /// <summary>
        /// Initializes a new instance of the <see
        cref="KinectViewModelLoader"/> class.
        /// </summary>
        public KinectViewModelLoader()
        {
            _kinectService = new KinectService();
            _kinectService.Initialize();
        }
        #endregion .ctor
        #region Public Methods
        /// <summary>
        /// Cleans up this instance.
        /// </summary>
        public static void Cleanup()
        {
            _kinectService.Cleanup();
        }
        public static void InitializeCameraColor()
        {
            _kinectService.KinectCameraImage = null;
        }
    }
}
```

```

        _kinectService._mode = CameraMode.Color;
    }
    public static void InitializeBackgroundRemoved()
    {
        _kinectService.KinectCameraImage=null;
        _kinectService._mode =
CameraMode.BackgroundRemoved;
    }
    public static WriteableBitmap Camera()
    {
        return _kinectService.KinectCameraImage;
    }
    #endregion Public Methods
}
}

```

✓ Clase KinectService

```

    public void Initialize()
    {
        Hand = new HandTracking();

        #if DEBUG
        SkeletonManager = new SkeletonManager();
        #endif

        ErrorGridVisibility = Visibility.Hidden;
        ClothesAreaVisibility = Visibility.Visible;
        DiscoverKinectSensors();
        _mode= CameraMode.Color;
    }

```

✓ Inicializa las fuentes de datos de la cual para el seguimiento de la mano es de Body.

```

    private void InitializeKinectSensor(KinectSensor
sensor)
    {
        if (sensor != null)
        {
            try
            {
                sensor.Open();
                LoadSpeechEngine(sensor);
                _reader =
sensor.OpenMultiSourceFrameReader(FrameSourceTypes.Color |
FrameSourceTypes.Depth |
FrameSourceTypes.Body |
FrameSourceTypes.BodyIndex);
                _reader.MultiSourceFrameArrived +=
Reader_MultiSourceFrameArrived;

                //Se asigna el mapping del sensor en
caso de background
                _backgroundRemovalTool = new
BackgroundRemovalTool(sensor.CoordinateMapper);
            }
            catch (Exception)

```



```

        {
            UninitializeKinectSensor(sensor);
            Kinect = null;
            ErrorGridVisibility =
Visibility.Visible;
            ErrorGridMessage = "Se encontro
problemas con el Kinect." + Environment.NewLine +
                "Trate de desconectar y volver a
conectar el dispositivo al ordenador." +
Environment.NewLine +
                "Asegúrese de la conexiones de el
Kinect o la fuente de alimentacion.";
        }
    }
}

```

- ✓ En el método Reader_MultiSourceFrameArrived que es el lector de los multiples fuentes ejecutados esta del tipo Body, y se visualiza el llamado del método de modificar el cursor de la mano (posiciones.)

```

using (var frame =
reference.BodyFrameReference.AcquireFrame())
{
    if (frame != null)
    {
        _bodies = new
Body[frame.BodyFrameSource.BodyCount];

        frame.GetAndRefreshBodyData(_bodies);

        var body = GetPrimaryBody(_bodies);

        if (body== null)
        {
            ErrorGridVisibility =
Visibility.Visible;
            ErrorGridMessage = "No detectado
esqueleto o perdido su posición." + Environment.NewLine +
                "Espera un minuto y compruebe
si se pone de pie a una distancia apropiada desde el
dispositivo.";
            ClothesAreaVisibility =
Visibility.Hidden;
            return;
        }
        if (ClothesAreaVisibility ==
Visibility.Hidden)
        {
            ErrorGridVisibility =
Visibility.Collapsed;
            ClothesAreaVisibility =
Visibility.Visible;
        }
        Hand.UpdateHandCursor(body, Kinect,
Width, Height);
    }
}

```

```

        ClothingManager.Instance.UpdateItemPosition(body, Kinect,
            Width, Height);
    
```

- ✓ En la clase HandTraking.

```

    public void UpdateHandCursor(Body body, KinectSensor sensor, double
width, double height)
    
```

```

    {
        if (body == null) return;
    
```

```

        TrackHand(body.Joints[JointType.HandLeft],
body.Joints[JointType.HandRight], sensor, width, height);
    }
    
```

- ✓ El método de seguimiento de mano

```

    private void TrackHand(Joint leftHand, Joint rightHand,
KinectSensor sensor, double width, double height)
    
```

```

    {
        if (leftHand.TrackingState ==
TrackingState.NotTracked && rightHand.TrackingState ==
TrackingState.NotTracked)
            return;
    
```

```

        CameraSpacePoint
jointPositionLH=leftHand.Position;
        // DepthImagePoint leftPoint =
sensor.CoordinateMapper.MapSkeletonPointToDepthPoint(leftHa
nd.Position, sensor.DepthStream.Format);
        DepthSpacePoint leftPoint =
sensor.CoordinateMapper.MapCameraPointToDepthSpace(jointPos
itionLH);
    
```

```

        int lx = (int)((leftPoint.X * width /
sensor.DepthFrameSource.FrameDescription.Width));
        int ly = (int)((leftPoint.Y * height /
sensor.DepthFrameSource.FrameDescription.Height));
    
```

```

        CameraSpacePoint jointPositionRH =
rightHand.Position;
        // DepthImagePoint rightPoint =
sensor.CoordinateMapper.MapSkeletonPointToDepthPoint(rightH
and.Position, sensor.DepthStream.Format);
        DepthSpacePoint rightPoint =
sensor.CoordinateMapper.MapCameraPointToDepthSpace(jointPos
itionRH);
    
```

```

        int rx = (int)((rightPoint.X * width /
sensor.DepthFrameSource.FrameDescription.Width));
        int ry = (int)((rightPoint.Y * height /
sensor.DepthFrameSource.FrameDescription.Height));
    
```

```

        LeftPosition = new Point(lx, ly);
        RightPosition = new Point(rx, ry);
    
```

```

    }
    #endregion Methods
    
```

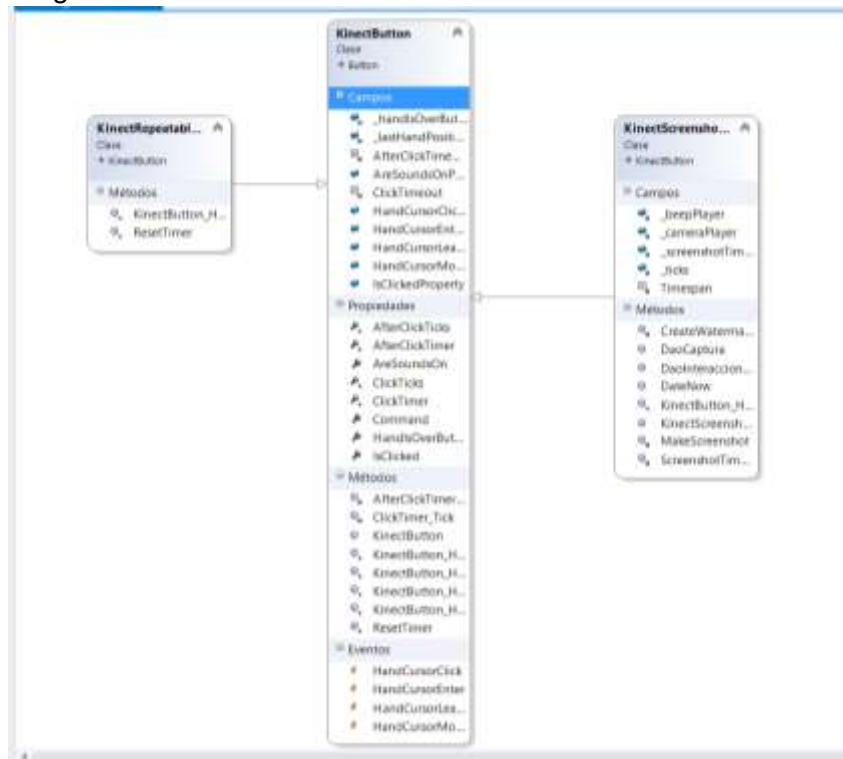
```

}
    
```

- ✓ Paso de valor a la ventana de la aplicación.

```
#region Public Properties
/// <summary>
/// Gets or sets the Position of the left hand.
/// </summary>
/// <value>
/// The Position of the left hand.
/// </value>
public Point LeftPosition
{
    get { return _leftPosition; }
    set
    {
        if (_leftPosition == value)
            return;
        _leftPosition = value;
        OnPropertyChanged("LeftPosition");
    }
}
/// <summary>
/// Gets or sets the Position of the right hand.
/// </summary>
/// <value>
/// The Position of the right hand.
/// </value>
public Point RightPosition
{
    get { return _rightPosition; }
    set
    {
        if (_rightPosition == value)
            return;
        _rightPosition = value;
        OnPropertyChanged("RightPosition");
    }
}
#endregion Public Properties
```


- Selección de una mano con un botón.
 ✓ Diagrama de clases de Boton.



- ✓ En el XAML (diseño se llama a la clase).

```

<buttons:KinectButton Width="{Binding Width,
ConverterParameter=30, Converter={StaticResource
ReducedValueConverter}, ElementName=LeftPanel}"

    Height="{Binding Width, ConverterParameter=30,
Converter={StaticResource ReducedValueConverter},
ElementName=LeftPanel}"

    Margin="5" >
    <i:Interaction.Triggers>
    <i:EventTrigger
EventName="HandCursorClick">
<i:InvokeCommandAction Command="{Binding ClickCommand}"
/>
    </i:EventTrigger>
</i:Interaction.Triggers>
<Image Source="{Binding
Image, Converter={StaticResource
BitmapToImageSourceConverter}}"

    Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

    Height="{Binding Height,

```



```
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}]"
```

```
Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}]"
```

```
Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}]"
```

```
Margin="-5"/>
```

```
</buttons:KinectButton>
```

- ✓ A la hora de click con la mano , ya se creo la instancia inicializando lo eventos de enter del botón,click, dejar de hacer click.

```
using appProbadorVirtual3D.View.Buttons.Events;
using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using System.Windows.Threading;
```

```
namespace appProbadorVirtual3D.View.Buttons
{
```

```
/// <summary>
```

```
/// Button class that responds to Kinect events
```

```
/// </summary>
```

```
public class KinectButton : Button
```

```
{
```

```
    #region Constants
```

```
    /// <summary>
```

```
    /// Number of seconds that Click event occurs
```

```
    /// </summary>
```

```
    private const int ClickTimeout = 40;
```

```
    /// <summary>
```

```
    /// Number of seconds after Click event
```

```
    /// </summary>
```

```
    private const int AfterClickTimeout = 10;
```

```
    #endregion Constants
```

```
    #region Private Fields
```

```
    /// <summary>
```

```
    /// Determines if hand is over button
```

```
    /// </summary>
```

```
    private bool _handIsOverButton;
```

```
    /// <summary>
```

```
    /// The last hand position
```

```
    /// </summary>
```

```
    private Point _lastHandPosition;
```

```
    #endregion Private Fields
```



```

#region Events
/// <summary>
/// Hand cursor enter event
/// </summary>
public static readonly RoutedEvent HandCursorEnterEvent
=
KinectEvents.HandCursorEnterEvent.AddOwner(typeof(KinectButt
on));
/// <summary>
/// Hand cursor move event
/// </summary>
public static readonly RoutedEvent HandCursorMoveEvent
=
KinectEvents.HandCursorMoveEvent.AddOwner(typeof(KinectButt
on));
/// <summary>
/// Hand cursor leave event
/// </summary>
public static readonly RoutedEvent HandCursorLeaveEvent
=
KinectEvents.HandCursorLeaveEvent.AddOwner(typeof(KinectButt
on));
/// <summary>
/// Hand cursor click event
/// </summary>
public static readonly RoutedEvent HandCursorClickEvent
=
KinectEvents.HandCursorClickEvent.AddOwner(typeof(KinectButt
on));
#endregion Events
#region Event handlers
/// <summary>
/// Hand cursor enter event handler
/// </summary>
public event HandCursorEventHandler HandCursorEnter
{
    add { AddHandler(HandCursorEnterEvent, value); }
    remove { RemoveHandler(HandCursorEnterEvent, value); }
}
/// <summary>
/// Hand cursor move event handler
/// </summary>
public event HandCursorEventHandler HandCursorMove
{
    add { AddHandler(HandCursorMoveEvent, value); }
    remove { RemoveHandler(HandCursorMoveEvent, value); }
}
}
/// <summary>

```



```

/// Hand cursor leave event handler
/// </summary>
public event HandCursorEventHandler HandCursorLeave
{
    add { AddHandler(HandCursorLeaveEvent, value); }
    remove { RemoveHandler(HandCursorLeaveEvent, value); }
}
}
/// <summary>
/// Hand cursor click event handler
/// </summary>
public event HandCursorEventHandler HandCursorClick
{
    add { AddHandler(HandCursorClickEvent, value); }
    remove { RemoveHandler(HandCursorClickEvent, value); }
}
#endregion Event handlers
#region Properties
/// <summary>
/// Get information if hand is over button
/// </summary>
public bool HandIsOverButton
{
    get { return _handIsOverButton; }
}
/// <summary>
/// Has Click event occurred
/// </summary>
public bool IsClicked
{
    get { return (bool)GetValue(IsClickedProperty); }
    set { SetValue(IsClickedProperty, value); }
}
/// <summary>
/// Gets or sets information about sounds.
/// </summary>
/// <value>
/// Information about sounds.
/// </value>
public bool AreSoundsOn
{
    get { return (bool)GetValue(AreSoundsOnProperty); }
    set { SetValue(AreSoundsOnProperty, value); }
}
/// <summary>
/// Gets or sets the command to invoke when this button is
pressed.
/// </summary>
public new ICommand Command

```

```

{
    get { return (ICommand)GetValue(CommandProperty); }
    set { SetValue(CommandProperty, value); }
}
/// <summary>
/// Number of elapsed ticks for _clickTimer
/// </summary>
protected int ClickTicks { get; set; }
/// <summary>
/// Number of elapsed ticks for _afterClickTimer
/// </summary>
protected int AfterClickTicks { get; set; }
/// <summary>
/// Determines how much time elapsed since
HandCursorEnterEvent occurred
/// </summary>
protected DispatcherTimer ClickTimer { get; private set; }
/// <summary>
/// Determines how much time elapsed since
HandCursorClickEvent occurred
/// </summary>
protected DispatcherTimer AfterClickTimer { get; private set; }
#endregion Properties
#region Dependency Properties
/// <summary>
/// IsClicked dependency property
/// </summary>
public static readonly DependencyProperty IsClickedProperty
= DependencyProperty.Register(
    "IsClicked", typeof(bool), typeof(KinectButton), new
PropertyMetadata(default(bool)));
/// <summary>
/// AreSoundsOn dependency property
/// </summary>
public static readonly DependencyProperty
AreSoundsOnProperty = DependencyProperty.Register(
    "AreSoundsOn", typeof(bool), typeof(KinectButton), new
PropertyMetadata(default(bool)));
#endregion Dependency Properties
#region .ctor
/// <summary>
/// Initializes a new instance of the <see cref="KinectButton"/>
class.
/// </summary>
public KinectButton()
{
    SetValue(IsClickedProperty, false);
    _handIsOverButton = false;
}

```

```

        ClickTimer = new DispatcherTimer { Interval = new
TimeSpan(0, 0, 0, 0, 1) };
        ClickTicks = 0;
        ClickTimer.Tick += ClickTimer_Tick;
        AfterClickTimer = new DispatcherTimer { Interval = new
TimeSpan(0, 0, 0, 0, 1) };
        AfterClickTicks = 0;
        AfterClickTimer.Tick += AfterClickTimer_Tick;

        HandCursorEnter += KinectButton_HandCursorEnter;
        HandCursorMove += KinectButton_HandCursorMove;
        HandCursorLeave += KinectButton_HandCursorLeave;
        HandCursorClick += KinectButton_HandCursorClick;
    }
#endregion .ctor
#region Methods
/// <summary>
/// Handles HandCursorEnter event
/// </summary>
protected void KinectButton_HandCursorEnter(object sender,
HandCursorEventArgs args)
{
    _handlsOverButton = true;
    ClickTimer.Start();
}
/// <summary>
/// Handles HandCursorMove event
/// </summary>
protected void KinectButton_HandCursorMove(object sender,
HandCursorEventArgs args)
{
    _lastHandPosition = new Point(args.X, args.Y);
}
/// <summary>
/// Handles HandCursorLeave event
/// </summary>
protected void KinectButton_HandCursorLeave(object
sender, HandCursorEventArgs args)
{
    _handlsOverButton = false;
    if (IsClicked)
        SetValue(IsClickedProperty, false);
    ResetTimer(ClickTimer);
}
/// <summary>
/// Counts the number of timer ticks of_clickTimer
/// </summary>
private void ClickTimer_Tick(object sender, EventArgs e)
{

```



```

ClickTicks++;

if (ClickTicks <= ClickTimeout)
    return;

ResetTimer(ClickTimer);
RaiseEvent(new
HandCursorEventArgs(HandCursorClickEvent,
_lastHandPosition));
}
/// <summary>
/// Counts the number of timer ticks of _afterClickTimer
/// </summary>
private void AfterClickTimer_Tick(object sender, EventArgs e)
{
    AfterClickTicks++;

    if (AfterClickTicks <= AfterClickTimeout)
        return;

    ResetTimer(AfterClickTimer);
    SetValue(IsClickedProperty, false);
}
/// <summary>
/// Imitates the click event
/// </summary>
protected virtual void KinectButton_HandCursorClick(object
sender, HandCursorEventArgs args)
{
    SetValue(IsClickedProperty, true);
    AfterClickTimer.Start();
}
/// <summary>
/// Resets the timer
/// </summary>
protected virtual void ResetTimer(DispatcherTimer timer)
{
    timer.Stop();
    if (timer == ClickTimer)
        ClickTicks = 0;
    else
        AfterClickTicks = 0;
}
#endregion Methods
}
}

```

- *Scroll Horizontal y Vertical con la Mano.*
- ✓ En la clase MainWindow.

```
private void HandleHandMoved(Point leftHand, Point rightHand)
{
    HandCursor.Visibility = Visibility.Collapsed;
    var element = ButtonPanelsCanvas.InputHitTest(leftHand);

    if (CloseAppGrid.Visibility == Visibility.Visible)
    {
        element = CloseAppGrid.InputHitTest(leftHand);
    }
    else if (QuestionScreenshotAppGrid.Visibility ==
    Visibility.Visible)
    {
        element = QuestionScreenshotAppGrid.InputHitTest(leftHand);
    }
    else if (SocialMediasAppGrid.Visibility == Visibility.Visible)
    {
        element = SocialMediasAppGrid.InputHitTest(leftHand);
    }
    else if (TypeProductAppGrid.Visibility == Visibility.Visible)
    {
        element = TypeProductAppGrid.InputHitTest(leftHand);
    }

    var hand = leftHand;

    if (!(element is UIElement))
    {
        if (CloseAppGrid.Visibility == Visibility.Visible)
        {
            element = CloseAppGrid.InputHitTest(rightHand);
        }
        else if (QuestionScreenshotAppGrid.Visibility ==
        Visibility.Visible)
        {
            element = QuestionScreenshotAppGrid.InputHitTest(rightHand);
        }
        else if (SocialMediasAppGrid.Visibility == Visibility.Visible)
        {
            element = SocialMediasAppGrid.InputHitTest(rightHand);
        }
        else if (TypeProductAppGrid.Visibility == Visibility.Visible)
        {
            element = TypeProductAppGrid.InputHitTest(rightHand);
        }
        else
        {
            element = ButtonPanelsCanvas.InputHitTest(rightHand);
        }
        hand = rightHand;
    }
    if (!(element is UIElement))
    {

```

```
ButtonsManager.Instance.RaiseCursorLeaveEvent(leftHand);
return;
}
}

HandCursor.Visibility = Visibility.Visible;
Canvas.SetLeft(HandCursor, hand.X - HandCursor.ActualWidth /
2.0);
Canvas.SetTop(HandCursor, hand.Y - HandCursor.ActualHeight /
2.0);
ButtonsManager.Instance.RaiseCursorEvents(element, hand);

}
```

```
<helpers:ScrollableCanvasHorizontal.ItemTemplate>
    <DataTemplate>
        <buttons:KinectButton Width="{Binding Width,
ConverterParameter=30, Converter={StaticResource ReducedValueConverter},
ElementName=LeftPanel}"
                                Height="{Binding Width,
ConverterParameter=30, Converter={StaticResource ReducedValueConverter},
ElementName=LeftPanel}"
                                Margin="5" >
            <i:Interaction.Triggers>
                <i:EventTrigger
EventName="HandCursorClick">
                    <i:InvokeCommandAction
Command="{Binding ClickCommand}" />
                </i:EventTrigger>
            </i:Interaction.Triggers>
            <Image Source="{Binding Image,
Converter={StaticResource BitmapToImageSourceConverter}}"
                                Width="{Binding
Width, RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Height="{Binding
Height, RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

                Canvas.Top="{Binding (Canvas.Top), RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}"
                                Margin="-5"/>
        </buttons:KinectButton>
    </DataTemplate>
</helpers:ScrollableCanvasHorizontal.ItemTemplate>
</helpers:ScrollableCanvasHorizontal>
```



✓ Clase ScrollableCanvas

```

using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Threading;
using appProbadorVirtual3D.View.Buttons.Events;

namespace appProbadorVirtual3D.View.Helpers
{
    public class ScrollableCanvas : ItemsControl
    {
        #region Constants
        /// <summary>
        /// Number of seconds to check position of Hand
        /// </summary>
        private const int EnterTimeout = 4;
        /// <summary>
        /// Translation of controls in panels
        /// </summary>
        private const int Distance = 10;
        /// <summary>
        /// Number of milliseconds of animation
        /// </summary>
        private const int TimeOfAnimation = 10;
        /// <summary>
        /// The minimum height factor
        /// </summary>
        private const double MinHeightFactor = 0.2;
        /// <summary>
        /// The maximum height factor
        /// </summary>
        private const double MaxHeightFactor = 0.4;
        #endregion
        #region Fields
        /// <summary>
        /// Position of LeftPanel
        /// </summary>
        private Point _leftPanelPosition;
        /// <summary>
        /// Determines how much time elapsed since hand position
        over canvas checked
        /// </summary>
        private readonly DispatcherTimer _enterTimer;
        /// <summary>
        /// Number of elapsed ticks for _enterTimer
        /// </summary>
        private int _enterTimerTicks;
    }
}

```



```

/// <summary>
/// Determines if hand is over canvas
/// </summary>
private bool _isHandOverCanvas;
/// <summary>
/// Actual hand position
/// </summary>
private Point _handPosition;
/// <summary>
/// Position of last button in panel
/// </summary>
double _lastButtonPositionY;
/// <summary>
/// Position of first button in panel
/// </summary>
double _firstButtonPositionY;
/// <summary>
/// Start point of animation
/// </summary>
double _startAnimationPoint;
/// <summary>
/// Defines if buttons are moving
/// </summary>
private bool _isMoved;
/// <summary>
/// Top boundary to start scroll up
/// </summary>
private double _canvasMinHeight;
/// <summary>
/// Bottom boundary to start scroll down
/// </summary>
private double _canvasMaxHeight;
#endregion
#region Events
/// <summary>
/// Hand cursor enter event
/// </summary>
public static readonly RoutedEvent HandCursorEnterEvent
=
KinectEvents.HandCursorEnterEvent.AddOwner(typeof(ScrollableC
anvas));
/// <summary>
/// Hand cursor leave event
/// </summary>
public static readonly RoutedEvent HandCursorLeaveEvent
=
KinectEvents.HandCursorLeaveEvent.AddOwner(typeof(Scrollable
Canvas));
/// <summary>

```

```

/// Hand cursor move event
/// </summary>
public static readonly RoutedEvent HandCursorMoveEvent
=
KinectEvents.HandCursorMoveEvent.AddOwner(typeof(ScrollableC
anvas));
#endregion
#region Event handlers
/// <summary>
/// Hand cursor enter event handler
/// </summary>
public event HandCursorEventHandler HandCursorEnter
{
    add { AddHandler(HandCursorEnterEvent, value); }
    remove { RemoveHandler(HandCursorEnterEvent, value); }
}
/// <summary>
/// Hand cursor leave event handler
/// </summary>
public event HandCursorEventHandler HandCursorLeave
{
    add { AddHandler(HandCursorLeaveEvent, value); }
    remove { RemoveHandler(HandCursorLeaveEvent, value); }
}
/// <summary>
/// Hand cursor move event handler
/// </summary>
public event HandCursorEventHandler HandCursorMove
{
    add { AddHandler(HandCursorLeaveEvent, value); }
    remove { RemoveHandler(HandCursorLeaveEvent, value); }
}
#endregion Event handlers
#region .ctor
/// <summary>
/// Initializes a new instance of the <see
cref="ScrollableCanvas"/> class.
/// </summary>
public ScrollableCanvas()
{
    HandCursorEnter += ScrollableCanvas_HandCursorEnter;
    HandCursorLeave += ScrollableCanvas_HandCursorLeave;
    HandCursorMove += ScrollableCanvas_HandCursorMove;

    _enterTimer = new DispatcherTimer { Interval = new
TimeSpan(0, 0, 0, 0, 1) };
    _enterTimer.Ticks = 0;
    _enterTimer.Tick += EnterTimer_Tick;

```




```

        Items.CurrentChanged += (sender, args) => {
_startAnimationPoint = 0; }
    }
    #endregion
    #region Methods
    /// <summary>
    /// Counts the number of timer ticks of _enterTimer
    /// </summary>
    private void EnterTimer_Tick(object sender, EventArgs e)
    {
        _enterTimerTicks++;

        if (_enterTimerTicks < EnterTimeout)
            return;

        _enterTimer.Stop();
        _enterTimerTicks = 0;
        if (_isHandOverCanvas)
            RaiseEvent(new
HandCursorEventArgs(HandCursorEnterEvent, _handPosition));
    }
    /// <summary>
    /// Handles HandCursorMove event
    /// </summary>
    private void ScrollableCanvas_HandCursorMove(object
sender, HandCursorEventArgs args)
    {
        if (_isHandOverCanvas)
            _handPosition = new Point(args.X, args.Y);
    }
    /// <summary>
    /// Handles HandCursorLeave event
    /// </summary>
    private void ScrollableCanvas_HandCursorLeave(object
sender, HandCursorEventArgs args)
    {
        _isHandOverCanvas = false;
    }
    /// <summary>
    /// Handles HandCursorEnter event
    /// </summary>
    private void ScrollableCanvas_HandCursorEnter(object
sender, HandCursorEventArgs args)
    {
        if (!_isHandOverCanvas)
            _handPosition = new Point(args.X, args.Y);
        _isHandOverCanvas = true;
        _isMoved = true;
    }
    }
    #endregion
}

```

```

        StackPanel stackPanel = (Name == "LeftScrollableCanvas")
?        FindChild<StackPanel>(Application.Current.MainWindow,
"LeftStackPanel")
:
(FindChild<StackPanel>(Application.Current.MainWindow,
"RightStackPanel"));
        if (stackPanel.Children.Count == 0)
            return;

        SetPositions(stackPanel);
        if (!CheckHandPosition(stackPanel))
            return;

        if (_isHandOverCanvas)
            _enterTimer.Start();
    }
    /// <summary>
    /// Checks hand position and runs MoveButtons method
    /// </summary>
    /// <param name="stackPanel">Collection of buttons in
panel</param>
    /// <returns>Value if hand is over panel sensitive
area</returns>
    private bool CheckHandPosition(StackPanel stackPanel)
    {
        if (_handPosition.Y > _canvasMinHeight && _handPosition.Y
< _canvasMaxHeight)
            return false;
        if (_handPosition.Y > _canvasMaxHeight)
            while (_isMoved && _lastButtonPositionY +
_startAnimationPoint > _canvasMaxHeight)
                MoveButtons(stackPanel, true);
        else if (_handPosition.Y < _canvasMinHeight)
            while (_isMoved && _firstButtonPositionY +
_startAnimationPoint < _firstButtonPositionY)
                MoveButtons(stackPanel, false);
        return true;
    }
    /// <summary>
    /// Sets positions of first and last buttons in panel
    /// Sets minimum and maximum height of panel sensitive area
    /// </summary>
    /// <param name="stackPanel">Collection of buttons in
panel</param>
    private void SetPositions(StackPanel stackPanel)
    {
        if (_firstButtonPositionY == 0)
            _firstButtonPositionY =
stackPanel.Children[0].TransformToAncestor(Application.Current.M
ainWindow).Transform(new Point(0, 0)).Y;

```

```

        _lastButtonPositionY =
stackPanel.Children[stackPanel.Children.Count
1].TransformToAncestor(Application.Current.MainWindow).Transform(
new Point(0, 0)).Y;

        if (_leftPanelPosition.X == 0 && _leftPanelPosition.Y == 0)
        {
            _leftPanelPosition =
TransformToAncestor(Application.Current.MainWindow).Transform(
new Point(0, 0));
            _canvasMinHeight = ActualHeight * MinHeightFactor +
_leftPanelPosition.Y;
            _canvasMaxHeight = ActualHeight * MaxHeightFactor +
_leftPanelPosition.Y;
        }
    }
    private void MoveButtons(StackPanel stackpanel, bool
moveUp)
    {
        _startAnimationPoint = moveUp ? _startAnimationPoint -
Distance : _startAnimationPoint + Distance;

        Button button;
        TranslateTransform translation = new TranslateTransform();
        DoubleAnimation animation = new DoubleAnimation()
        {
            Duration =
TimeSpan.FromMilliseconds(TimeOfAnimation),
            From = moveUp ? _startAnimationPoint + Distance :
_startAnimationPoint,
            To = moveUp ? _startAnimationPoint :
_startAnimationPoint + Distance
        };

        foreach (var control in stackpanel.Children)
        {
            button = FindChild<Button>(control as ContentPresenter,
            "");
            if (button != null)
                button.RenderTransform = translation;
        }

        translation.BeginAnimation(TranslateTransform.YProperty,
animation);
        _isMoved = !_isMoved;
    }

    private T FindChild<T>(DependencyObject parent, string
childName) where T : DependencyObject

```

```

    {
        if (parent == null)
            return null;

        int childrenCount =
        VisualTreeHelper.GetChildrenCount(parent);
        for (int i = 0; i < childrenCount; i++)
        {
            var child = VisualTreeHelper.GetChild(parent, i);
            T childType = child as T;
            if (childType == null)
            {
                T foundChild = FindChild<T>(child, childName);
                if (foundChild != null)
                    return foundChild;
            }
            else if (!string.IsNullOrEmpty(childName))
            {
                var frameworkElement = child as FrameworkElement;
                if (frameworkElement != null &&
                frameworkElement.Name == childName)
                    return (T)child;
            }
            else
                return (T)child;
        }
        return null;
    }
    #endregion
}
}

```

- *Interacción con voz.*

- ✓ En clase KinectService. Se carga la ingeniería del habla.

```

public void LoadSpeechEngine(KinectSensor sensor) {
    //Creamos una variable de tipo Grammar
    utilizando como parametro a grammarb
    var grammar = DictionarySpeech();

    //Le decimos a nuestra variable speechengine
    que cargue a grammar
    this.speechengine.LoadGrammar(grammar);

    //mandamos llamar al evento SpeechRecognized
    el cual se ejecutara cada vez que una palabra sea
    detectada
    speechengine.SpeechRecognized += new
    EventHandler<SpeechRecognizedEventArgs>(speechengine_Spe
    echRecognized);
}

```

```

try
{
    //speech engine inicia la entrada de
    datos de tipo audio
    SpeechAudioFormatInfo safi=new
    SpeechAudioFormatInfo(EncodingFormat.Pcm, 16000, 16, 1,
    32000, 2, null);
    // grab the audio stream
    IReadOnlyList<AudioBeam> audioBeamList =
    sensor.AudioSource.AudioBeams;
    System.IO.Stream audioStream =
    audioBeamList[0].OpenInputStream();

    // create the convert stream
    KinectAudioStream convertStream = new
    KinectAudioStream(audioStream);
    convertStream.SpeechActive = true;

    speechengine.SetInputToAudioStream(convertStream, safi);

    speechengine.RecognizeAsync(RecognizeMode.Multiple);
    }
    catch (Exception)
    {
        UninitializeKinectSensor(sensor);
        ErrorGridVisibility =
    Visibility.Visible;
        ErrorGridMessage = "Se encontro
    problemas con el Kinect." + Environment.NewLine +
        "Trate de desconectar y volver a
    conectar el dispositivo al ordenador." +
    Environment.NewLine +
        "Asegúrese de la conexiones de
    el Kinect o la fuente de alimentacion.";
    }
}

```

- ✓ Se carga el diccionario speech

```

public Grammar DictionarySpeech() {
    //Creamos esta variable ri que tratara de
    encontrar un language pack valido haciendo uso del
    metodo obtenerLP()
    RecognizerInfo ri
    =getLanguagePack();

    //Si se encontro el language
    pack requerido lo asignaremos a nuestra variable
    speechengine
    if (ri != null)
    {
        this.speechengine = new
    SpeechRecognitionEngine(ri.Id);
    }
}

```



```

//Creamos esta variable
opciones la cual almacenara las opciones de palabras o
frases que podran ser reconocidas por el dispositivo
var opciones = new
Choices();

//Comenzamos a agregar las
opciones comenzando por el valor de opcion que tratamos
reconocer y una llave que identificara a ese valor
//Por ejemplo en esta linea
"blusa" es el valor de opcion y "BLUSA" es la llave
opciones.Add("blusa",
"BLUSA");

//En esta linea "abrigo" es
el valor de opcion y "ABRIGO" es la llave
opciones.Add("abrigo",
"ABRIGO");
opciones.Add("vestido",
"VESTIDO");
opciones.Add("casaca",
"CASACA");
opciones.Add("pantalon",
"PANTALON");
opciones.Add("camisa",
"CAMISA");
opciones.Add("falda",
"FALDA");
opciones.Add("top", "TOP");
opciones.Add("polo",
"POLO");
opciones.Add("traje",
"TRAJE");
opciones.Add("chompa",
"CHOMPA");
opciones.Add("foto",
"FOTO");
opciones.Add("mujer",
"MUJER");
opciones.Add("hombre",
"HOMBRE");
opciones.Add("genero",
"GENERO");
opciones.Add("salir",
"SALIR");
opciones.Add("quitar",
"QUITAR");
opciones.Add("limpiar todo",
"LIMPIARTODO");
opciones.Add(new
SemanticResultValue("cierrate sesamo", "CERRAR"));

opciones.Add(new
SemanticResultValue("kinect veinte y cinco",
"VEINTECINCO"));

```



```

//Esta variable creará todo
el conjunto de frases y palabras en base a nuestro
lenguaje elegido en la variable ri
        var grammarb = new
GrammarBuilder { Culture = ri.Culture };

//Agregamos las opciones de
palabras y frases a grammarb
        grammarb.Append(opciones);
        return new
Grammar(grammarb);
    }
    return null;
}

```

Se asigna el idioma adecuado, en este caso Español

```

private RecognizerInfo getLanguagePack()
{
    //Comienza a checar todos los languagepack que tengamos
    instalados
    foreach (RecognizerInfo recognizer in
SpeechRecognitionEngine.InstalledRecognizers())
    {
        string value;
        recognizer.AdditionalInfo.TryGetValue("Kinect", out
value);
        //Aqui es donde elegimos el lenguaje, si se dan cuenta
        hay una parte donde dice "es-MX" para cambiar el lenguaje a ingles de EU
        basta con cambiar el valor a "en-US"
        if ("True".Equals(value,
StringComparison.OrdinalIgnoreCase) && "es-
MX".Equals(recognizer.Culture.Name, StringComparison.OrdinalIgnoreCase))
        {
            //Si se encontro el language pack solicitado se
            retorna a recognizer
            return recognizer;
        }
        //En caso de que no se encuentre ningun lenguaje pack se
        retorna un valor nulo
        return null;
    }
}

```

Activacion del evento de reconocimiento de la voz:

```

void speechengine_SpeechRecognized(object sender,
SpeechRecognizedEventArgs e)
{
    //la variable igualdad sera el porcentaje de igualdad entre la
    palabra reconocida y el valor de opcion
    //es decir si yo digo "uno" y el valor de opcion es "uno" la
    igualdad sera mayor al 50 %
}

```

```

//Si yo digo "jugo" y el valor de opcion es "uno" notaras que el
sonido es muy similar pero quizas no mayor al 50 %
//El valor de porcentaje va de 0.0 a 1.0, ademas notaras que
le di un valos de .5 lo cual representa el 50% de igualdad
const double igualdad = 0.3;

//Si hay mas del 50% de igualdad con alguna de nuestras
opciones
if (e.Result.Confidence > igualdad)
{
    try{
        //haremos un switch para aquellos valores que se
componen de unicamente una palabra
switch (e.Result.Words[0].Text.ToUpper())
{
    //En caso de que digamos "uno" la llave "UNO" se abraira
y se realizara lo siguiente
    case "BLUSA":
        ExecutedClothingCategory("BlouseItem");
        break;
    case "ABRIGO":
        ExecutedClothingCategory("CoatItem");
        break;

    case "FALDA":
        ExecutedClothingCategory("SkirtItem");
        break;
    case "VESTIDO":
        ExecutedClothingCategory("DressItem");
        break;
    case "CAMISA":
        ExecutedClothingCategory("ShirtItem");
        break;
    case "CHOMPA":
        ExecutedClothingCategory("SweaterItem");
        break;
    case "TOP":
        ExecutedClothingCategory("TopItem");
        break;
    case "TRAJE":
        ExecutedClothingCategory("SuitItem");
        break;
    case "CASACA":
        ExecutedClothingCategory("JacketItem");
        break;
    case "PANTALON":
        ExecutedClothingCategory("PantItem");
        break;
    case "POLO":

```



```

        ExecutedClothingCategory("TshirtItem");
        break;
    case "GENERO":

        TopMenuManager.Instance.MenuButton.ClickExecuted();
        // ObservableCollection<TopMenuButtonViewModel>
        lista = TopMenuManager.Instance.AllButtons;

        TopMenuManager.Instance.AllButtons[0].ClickExecuted();
        break;
    case "MUJER":
        if (TopMenuManager.Instance.AllButtons==null)
        {

            TopMenuManager.Instance.MenuButton.ClickExecuted();
        }

        // ObservableCollection<TopMenuButtonViewModel>
        lista = TopMenuManager.Instance.AllButtons;
        ClothingItemBase.MaleFemaleType genero_male =
        ClothingManager.Instance.ChosenType;
        if (genero_male ==
        ClothingItemBase.MaleFemaleType.Male)
        {

            TopMenuManager.Instance.AllButtons[0].ClickExecuted();
        }

        break;
    case "HOMBRE":
        if (TopMenuManager.Instance.AllButtons == null)
        {

            TopMenuManager.Instance.MenuButton.ClickExecuted();
        }

        // ObservableCollection<TopMenuButtonViewModel>
        lista = TopMenuManager.Instance.AllButtons;
        ClothingItemBase.MaleFemaleType genero_female =
        ClothingManager.Instance.ChosenType;
        if (genero_female ==
        ClothingItemBase.MaleFemaleType.Female)
        {

            TopMenuManager.Instance.AllButtons[0].ClickExecuted();
        }

        break;
    case "QUITAR":

```



```

String accion = request.getParameter("accion");
if (accion != null) {
    if ("updateQuantity".equals(accion)) {
        updateQuantity(request, response);
    } else if ("buy".equals(accion)) {
        buyClothes(request, response);
    }
} else {
    String token = request.getParameter("ts");
    if (token != null) {
        LogicInteraccion logicInteraccion = new
LogicInteraccion();
        Interaccion interaccion =
logicInteraccion.getInteraccion(token);

        if (interaccion != null) {
            List<DetalleInteraccion> lDetalleInteraccion =
interaccion.getlDetalleInteraccion();
            if (lDetalleInteraccion.size() != 0) {
                request.setAttribute("interaccion", interaccion);
                int titt = CalcularTiempo(interaccion);
                String formato = formatoTiempo(titt);
                session.setAttribute("demora", formato);
                session.setAttribute("sesDInteraccion",
lDetalleInteraccion);

                request.getRequestDispatcher("QRinteraccion.jsp").forward(request,
response);
            } else {
                request.setAttribute("tipo", "Restriccionres de
Interaccion");
                request.setAttribute("error", "¡Debe colocarse por lo
menos una categoria de ropa!");

                request.getRequestDispatcher("paginaError.jsp").forward(request,
response);
            }
        } else {
            request.setAttribute("tipo", "Existencia Interaccion");
            request.setAttribute("error", "¡No ha realizado ninguna
interaccion!");
        }
    }
}
    
```

```

request.getRequestDispatcher("paginaError.jsp").forward(request,
response);
    }
    } else {
        request.setAttribute("tipo", "Existencia Interaccion");
        request.setAttribute("error", "¡No ha realizado ninguna
interaccion!");

request.getRequestDispatcher("paginaError.jsp").forward(request,
response);
    }
    }
    } catch (Exception e) {
        out.print("error de BD");
        request.setAttribute("tipo", e.getCause());
        request.setAttribute("error", e.getMessage());

request.getRequestDispatcher("paginaError.jsp").forward(request,
response);
    } finally {
        out.close();
    }
}

```

Buyclothes

```

private void buyClothes(HttpServletRequest request,
HttpServletRequest response) throws Exception {
    try {
        HttpSession session = request.getSession();
        LogicInteraccion logicInteraccion=new LogicInteraccion();
        List<DetalleInteraccion> lDetalleInteraccion =
(List<DetalleInteraccion>) session.getAttribute("sesDInteraccion");
        LogicDetalleInteraccion logicDetalleInteraccion = new
LogicDetalleInteraccion();
        Interaccion interaccion =
logicInteraccion.getInteraccion(lDetalleInteraccion.get(0).getoInteraccion
().getToken());
        String[] abecedario = {"A", "B", "C", "D", "E", "F", "G", "H", "I",
"J",
        "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W",
"X", "Y", "Z"};
        int numRandon = (int) Math.round(Math.random() * 26);

```




```

String codigoCompra = interaccion.getId() + "" +
abecedario[numRandom];

logicDetalleInteraccion.modificarDatosInteraccion(IDetalleInteraccion,
codigoCompra);
    Gson gson = new Gson();
    PrintWriter out = response.getWriter();
    Map records = new HashMap<String, Object>();
    records.put("codigoCompra", codigoCompra);
    double costoCompra=0;
    String html = "<span class='label label-success'>Codigo
Compra:"+codigoCompra+"</span><br>"
                + "<span class='label label-success'>Compra
Realizada:</span>"
                + "
                <div
class='row'><div class='col-md-6 col-sm-6 col-xs-6'> <ul class='list-
unstyled'>";
                for (int i = 0; i < IDetalleInteraccion.size(); i++) {
                    DetalleInteraccion objeto=IDetalleInteraccion.get(i);
                    double precio=objeto.getDetalleProducto().getPrecio();
                    double
costoTotal=objeto.getCantidad()*objeto.getDetalleProducto().getPrecio()
;
                    DecimalFormat df= new DecimalFormat("#0.00");
                    String numeroConFormato= df.format(costoTotal);
                    html += "
                    <li><span class='label label-default
margin-left-20'>Producto "+(i+1)+":</span></li>"
                    + "
                    <li><span class='label label-info margin-
left-20'>Id:
"+objeto.getDetalleProducto().getProducto().getId()+"</span></li>"
                    + "
                    <li><span class='label label-info margin-
left-20'>Nombre:
"+objeto.getDetalleProducto().getProducto().getNombre()+"</span></l
i>"
                    + "
                    <li><span class='label label-info margin-
left-20'>Categoria:
"+objeto.getDetalleProducto().getProducto().getDetalleCategoria().get
oCategoria().getNombre()+"</span> </li>"
                    + "
                    <li><span class='label label-info margin-
left-20'>Cantidad: "+objeto.getCantidad()+"</span></li>"
                    + "
                    <li><span class='label label-info margin-
left-20'>Precio Unit.: S/. "+precio+"</span></li>"

```

```

                + "                <li><span class='label label-info margin-
left-20'>Costo Total.: S/. "+numeroConFormato+"</span></li>";
                costoCompra+=costoTotal;
            }
            DecimalFormat df= new DecimalFormat("#0.00");
            String numeroConFormato= df.format(costoCompra);
            html += "                <li><span class='label label-success
margin-left-20'>Costo Total Compra:</span></li>"
                + "                <li><span class='label label-important label-
icon label-primary margin-left-
20'>S/. "+numeroConFormato+"</span></li>"
                + "                </ul></div></div>";
            records.put("html", html);

records.put("foto",interaccion.getlCaptura().get(0).getFoto());
            out.print(gson.toJson(records));

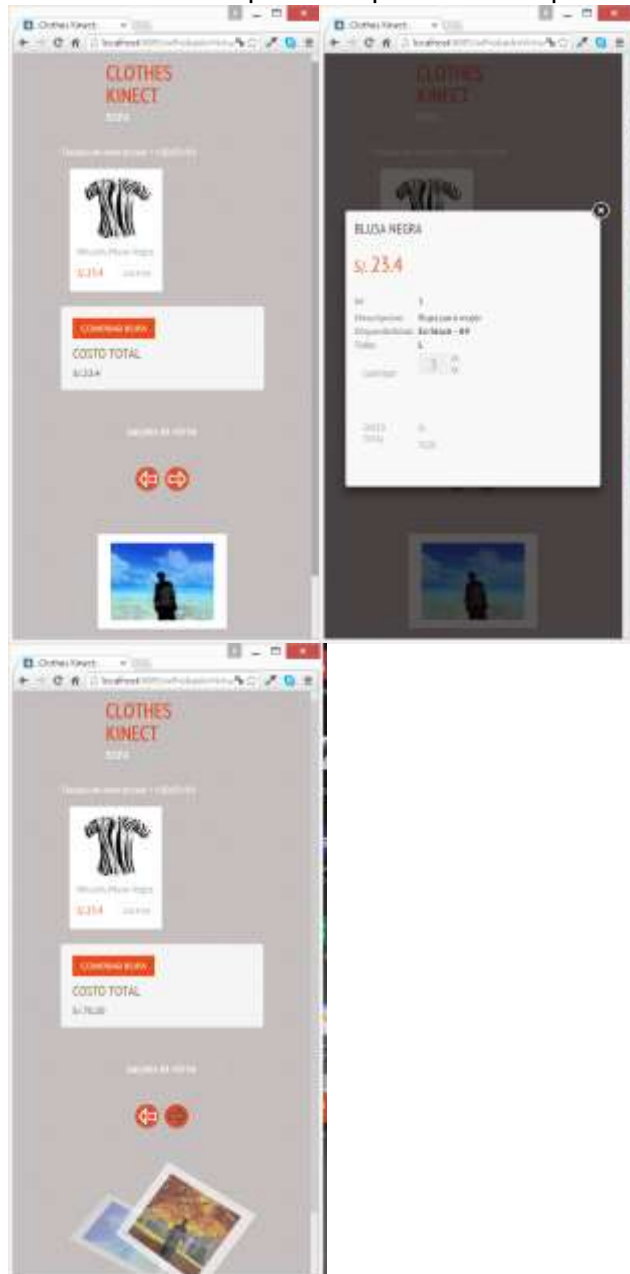
        } catch (Exception e) {
            throw e;
        }
    }
}

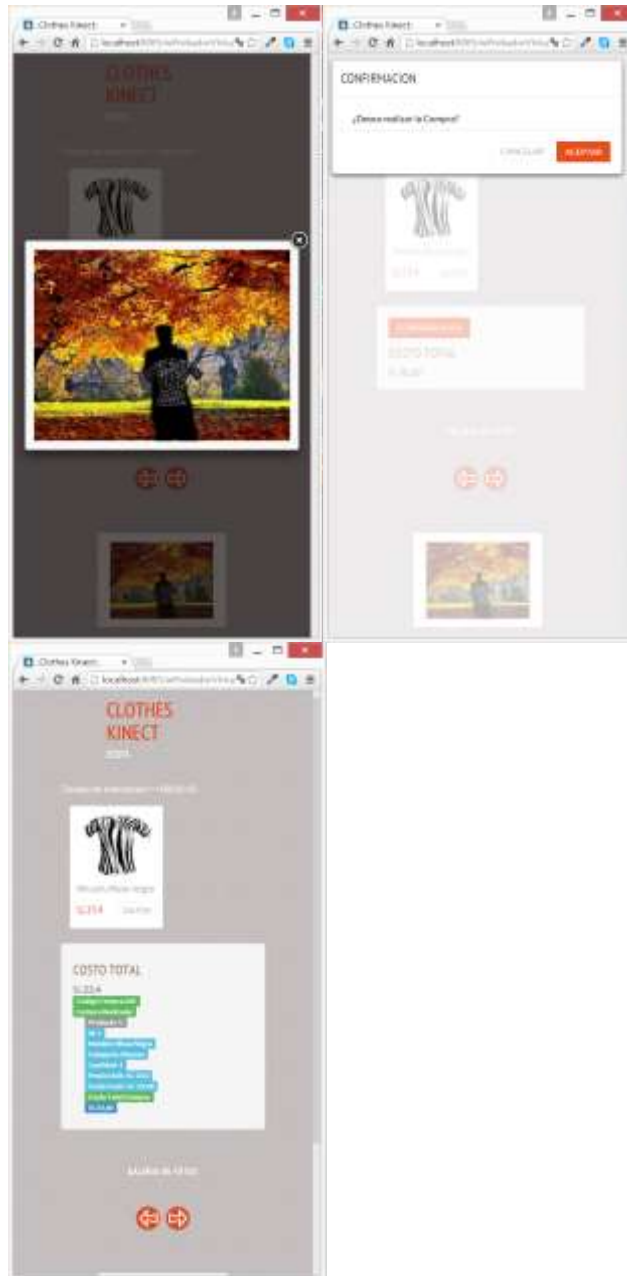
```

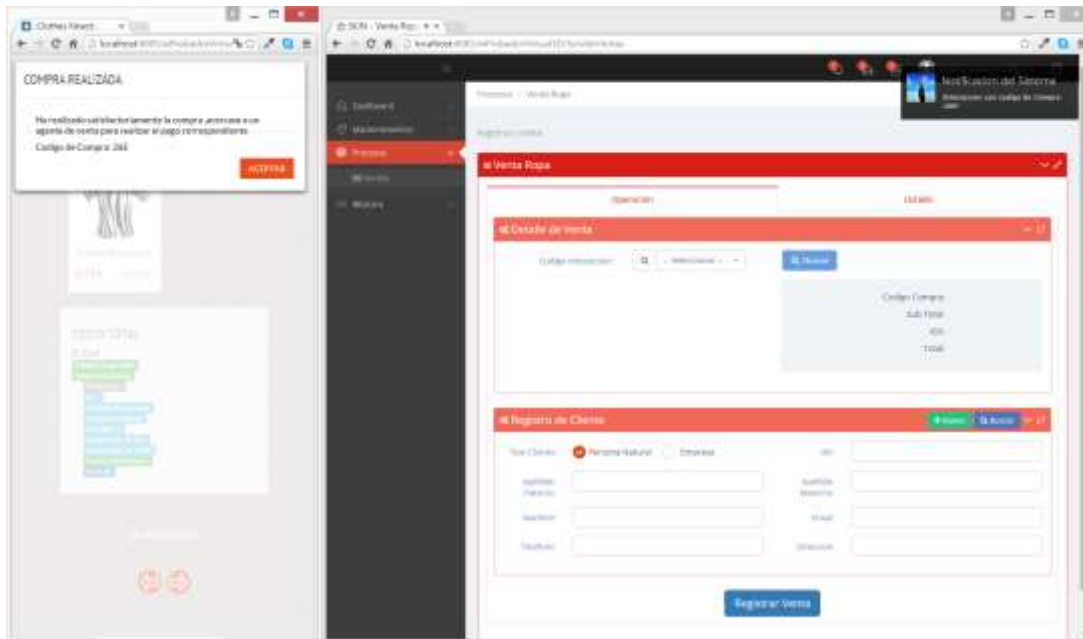


IV. Interacción de la Web y Aplicación

- ✓ Una vez que el usuario lea con el código QR se aparecerá ,este formulario con las prendas que se coloco para comprar.







V. Código Fuente de Reconocimiento del dispositivo Kinect 2.0 en la PC.

Para que realice reconocimiento del dispositivo se verifica por defecto, además el dispositivo kinect 2.0 puede soportar varios servicios abiertos (varias aplicaciones del kinect que comparten el dispositivo a diferencia del kinect 1.0).

- ✓ Se busca los dispositivos kinect.

```
private void DiscoverKinectSensors()
{
    Kinect = KinectSensor.GetDefault();
    this.Kinect.IsAvailableChanged +=
    Kinect_IsAvailableChanged;

    if (Kinect == null)
    {
        ErrorGridVisibility =
        Visibility.Visible;
        ErrorGridMessage = "Por favor conecte el
        kinect";
    }
}
```

- ✓ Se asigna el kinect por defecto a la propiedad de Kinect que significa que si hay valor, se asigna y esta disponible se inicializa el servicio del kinect.

```
public KinectSensor Kinect
{
    get { return _kinectSensor; }
    set
    {
```

```

        if (_kinectSensor != null)
        {
            UninitializeKinectSensor(_kinectSensor);
            _kinectSensor = null;
        }
        if (value != null && !value.IsAvailable)
        {
            _kinectSensor = value;
        }
        InitializeKinectSensor(_kinectSensor);
    }
}

```

- ✓ Se abre el servicio del Kinect, se carga lo demás tecnologías ya comentados como en el caso de speech del mismo Microsoft, se asigna que fuente de datos se va a trabajar como se muestra se asigno :

FrameSourceTypes.Color –Para la cámara de color.

FrameSourceTypes.Depth – Para logra la profundidad como el fondo con la persona.

FrameSourceTypes.Body- Nos indique los puntos del cuerpo.

FrameSourceTypes.BodyIndex –Indicar el cuerpo de la persona sobre un fondo.

```

private void InitializeKinectSensor(KinectSensor sensor)
{
    if (sensor != null)
    {
        try
        {
            sensor.Open();
            LoadSpeechEngine(sensor);
            _reader =
sensor.OpenMultiSourceFrameReader(FrameSourceTypes.Color |
FrameSourceTypes.Depth |
FrameSourceTypes.Body |
FrameSourceTypes.BodyIndex);
            _reader.MultiSourceFrameArrived +=
Reader_MultiSourceFrameArrived;

            //Se asigna el mapping del sensor en caso de
background
            _backgroundRemovalTool = new
BackgroundRemovalTool(sensor.CoordinateMapper);
        }
        catch (Exception)
        {
            UninitializeKinectSensor(sensor);
            Kinect = null;
            ErrorGridVisibility = Visibility.Visible;
            ErrorGridMessage = "Se encontro problemas con el
Kinect." + Environment.NewLine +
"Trate de desconectar y volver a conectar el
dispositivo al ordenador." + Environment.NewLine +

```


"Asegúrese de la conexiones de el Kinect o la fuente de alimentacion.";

```
    }
  }
}
```

VI. Código fuente del Formulario principal de la prueba de Ropa.

El código fuente del formulario se basa en xaml.(WPF –Windows Presentation Foundation).Nos indica el código de la parte de la vista de la aplicación sobre lo paneles y sus botones para la prueba de la ropa

```
<Window
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:converters="clr-namespace:appProbadorVirtual3D.Converters"
  xmlns:buttons="clr-namespace:appProbadorVirtual3D.View.Buttons"
  xmlns:helpers="clr-namespace:appProbadorVirtual3D.View.Helpers"
  xmlns:d="http://schemas.microsoft.com/expressiredyon/blend/2008"
  xmlns:i="clr-
namespace:System.Windows.Interactivity;assembly=System.Windows.Interactiv
ity"
  xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
  xmlns:h="http://helix-toolkit.org/wpf"
  xmlns:k="http://schemas.microsoft.com/kinect/2013"
  xmlns:appProbadorVirtual3D="clr-namespace:appProbadorVirtual3D"
  xmlns:topMenuButtons="clr-
namespace:appProbadorVirtual3D.ViewModel.ButtonItems.TopMenuButtons"
  xmlns:clothingItems="clr-
namespace:appProbadorVirtual3D.Model.ClothingItems"
  xmlns:viewModel="clr-namespace:appProbadorVirtual3D.ViewModel"
  xmlns:d1="http://schemas.microsoft.com/expression/blend/2008"
x:Class="appProbadorVirtual3D.MainWindow"
  xmlns:qr="clr-
namespace:Gma.QrCodeNet.Encoding.Windows.WPF;assembly=Gma.QrCodeNet.Encod
ing"
  mc:Ignorable="d d1"
  WindowState="Maximized"
  WindowStyle="None"
  Title="Probador virtual 3D"
  Height="768" Width="1366"
  Cursor="None">
  <!--Recursos dentro MainWindow-->
  <Window.Resources>
    <converters:ReducedValueConverter x:Key="ReducedValueConverter"/>
    <converters:BitmapToImageSourceConverter
x:Key="BitmapToImageSourceConverter"/>
    <converters:IncreasedValueConverter
x:Key="IncreasedValueConverter"/>
    <converters:BoolToVisibilityConverter
x:Key="BoolToVisibilityConverter"/>
    <converters:ValueToVisibilityConverter
x:Key="ValueToVisibilityConverter"/>
    <converters:DebugConverter x:Key="DebugConverter"/>
    <helpers:DataTemplate3D x:Key="ModelTemplate">
```

```

        <ModelVisual3D Content="{Binding Model}"/>
    </helpers:DataTemplate3D>
</Window.Resources>
<Window.DataContext>
    <Binding Path="KinectViewModel" Source="{StaticResource
KinectViewModelLoader}"/>
</Window.DataContext>
<Window.Style>
    <Style>
        <Setter
Property="appProbadorVirtual3D:MainWindow.LeftPosition" Value="{Binding
KinectService.Hand.LeftPosition}"/>
        <Setter
Property="appProbadorVirtual3D:MainWindow.RightPosition" Value="{Binding
KinectService.Hand.RightPosition}"/>
    </Style>
</Window.Style>
<!--Main Grid. Mantiene todos los elementos de interfaz de usuario.-->
<Grid>
    <Grid.Background>
        <ImageBrush x:Name="fondo" Stretch="UniformToFill" />
    </Grid.Background>
    <!--Image from the Kinect camera {Binding
KinectService.KinectCameraImage}-->
    <Image x:Name="ImageArea" Source="{Binding
KinectService.KinectCameraImage}"
        helpers:SizeObserver.Observe="True"
        helpers:SizeObserver.ObservedWidth="{Binding
KinectService.Width, Mode=OneWayToSource}"
        helpers:SizeObserver.ObservedHeight="{Binding
KinectService.Height, Mode=OneWayToSource}"/>

    <Image Name ="Backdrop"
        helpers:SizeObserver.Observe="True"
        helpers:SizeObserver.ObservedWidth="{Binding
KinectService.Width, Mode=OneWayToSource}"
        helpers:SizeObserver.ObservedHeight="{Binding
KinectService.Height, Mode=OneWayToSource}"
        Visibility="Collapsed"
        />

    <Image Name ="MaskedColor"
        helpers:SizeObserver.Observe="True"
        helpers:SizeObserver.ObservedWidth="{Binding
KinectService.Width, Mode=OneWayToSource}"
        helpers:SizeObserver.ObservedHeight="{Binding
KinectService.Height, Mode=OneWayToSource}"
        Visibility="Collapsed"/>

    <!--3D control-->
    <helpers:HelixViewport3DEx CameraRotationMode="Turntable"
        ModelUpDirection="0, 1, 0"
        x:Name="ClothesArea"
        ShowCoordinateSystem="False"

```

```

        Visibility="{Binding
KinectService.ClothesAreaVisibility}"
        ShowViewCube="{Binding DebugModeOn}"
        ShowFrameRate="{Binding DebugModeOn}"
        ShowCameraInfo="{Binding DebugModeOn}"
        IsHitTestVisible="False"
        ClipToBounds="False"
        Width="{Binding KinectService.Width}"
        Height="{Binding KinectService.Height}"
        CameraTransform="{Binding CameraTransform,
Mode=OneWayToSource, Source={x:Static
viewModel:ClothingManager.Instance}}"
        ViewportTransform="{Binding ViewportTransform,
Mode=OneWayToSource, Source={x:Static
viewModel:ClothingManager.Instance}}">
        <helpers:HelixViewport3DEx.Camera>
            <OrthographicCamera Position="0,0,10"
LookDirection="0,0,-10" UpDirection="0,1,0"/>
        </helpers:HelixViewport3DEx.Camera>
        <helpers:HelixViewport3DEx.ItemsPanel>
            <ItemsPanelTemplate>
                <Grid Width="{Binding KinectService.Width}"
Height="{Binding KinectService.Height}"/>
            </ItemsPanelTemplate>
        </helpers:HelixViewport3DEx.ItemsPanel>
        <h:DefaultLights/>
        <helpers:ItemsVisual3D ItemTemplate="{StaticResource
ModelTemplate}"
            ItemsSource="{Binding
ChosenClothesModels.Values, Source={x:Static
viewModel:ClothingManager.Instance}}"/>
        </helpers:HelixViewport3DEx>
        <!--Canvas que contiene el panel de botones-->
        <Canvas x:Name="ButtonPanelsCanvas" Width="{Binding
KinectService.Width}" Height="{Binding KinectService.Height}">
            <Canvas x:Name="LeftPanel"
                Width="160"
                Height="{Binding ActualHeight,
ConverterParameter=40, Converter={StaticResource ReducedValueConverter},
RelativeSource={RelativeSource AncestorType={x:Type Canvas}},
Mode=FindAncestor}}"
                Background="LightBlue"
                Opacity="0.2"
                Margin="20"/>
        <!--StackPanel que contiene todos los botones con la ropa
categoría-->
        <helpers:ScrollableCanvas x:Name="LeftScrollableCanvas"
            ItemsSource="{Binding
ActualClothingCategories, Source={x:Static
viewModel:ClothingManager.Instance}}"
            ClipToBounds="True"
            Width="{Binding ActualWidth,
ElementName=LeftPanel}"
            Height="{Binding ActualHeight,
ElementName=LeftPanel}"

```

```

Canvas.Left="{Binding (Canvas.Left),
ElementName=LeftPanel}"
Canvas.Top="{Binding (Canvas.Top),
ElementName=LeftPanel}"
Margin="{Binding Margin,
ElementName=LeftPanel}">
    <helpers:ScrollableCanvas.ItemsPanel>
        <ItemsPanelTemplate>
            <StackPanel x:Name="LeftStackPanel"/>
        </ItemsPanelTemplate>
    </helpers:ScrollableCanvas.ItemsPanel>
    <helpers:ScrollableCanvas.ItemTemplate>
        <DataTemplate>
            <buttons:KinectButton Width="{Binding Width,
ConverterParameter=30, Converter={StaticResource ReducedValueConverter},
ElementName=LeftPanel}"
                                Height="{Binding Width,
ConverterParameter=30, Converter={StaticResource ReducedValueConverter},
ElementName=LeftPanel}"
                                Margin="5"
                                Style="{StaticResource GlassButton}">
                <i:Interaction.Triggers>
                    <i:EventTrigger
EventName="HandCursorClick">
                        <i:InvokeCommandAction
Command="{Binding ClickCommand}" />
                    </i:EventTrigger>
                </i:Interaction.Triggers>
                <Image Source="{Binding Image,
Converter={StaticResource BitmapToImageSourceConverter}}"
                                Width="{Binding
Width, RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Height="{Binding
Height, RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Canvas.Top="{Binding (Canvas.Top), RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}"
                                Margin="-5"/>
            </buttons:KinectButton>
        </DataTemplate>
    </helpers:ScrollableCanvas.ItemTemplate>
</helpers:ScrollableCanvas>
<Canvas x:Name="RightPanel"
        Width="160"
        Height="{Binding ActualHeight,
ConverterParameter=40, Converter={StaticResource ReducedValueConverter},
RelativeSource={RelativeSource AncestorType={x:Type Canvas},
Mode=FindAncestor}}"
        Background="LightBlue"
        Opacity="0.2"

```



```

        Canvas.Right="0"
        Margin="20"/>
        <!--StackPanel que contiene todos los botones con la ropa-->
        <helpers:ScrollableCanvas x:Name="RightScrollableCanvas"
ItemsSource="{Binding Clothing, Source={x:Static
viewModel:ClothingManager.Instance}}">
            ClipToBounds="True"
            Width="{Binding ActualWidth,
ElementName=RightPanel}"
            Height="{Binding ActualHeight,
ElementName=RightPanel}"
            Canvas.Right="{Binding (Canvas.Right),
ElementName=RightPanel}"
            Canvas.Top="{Binding (Canvas.Top),
ElementName=RightPanel}"
            Margin="{Binding Margin,
ElementName=RightPanel}">
            <helpers:ScrollableCanvas.ItemsPanel>
                <ItemsPanelTemplate>
                    <StackPanel x:Name="RightStackPanel"/>
                </ItemsPanelTemplate>
            </helpers:ScrollableCanvas.ItemsPanel>
            <helpers:ScrollableCanvas.ItemTemplate>
                <DataTemplate>
                    <buttons:KinectButton Width="{Binding Width,
ConverterParameter=20, Converter={StaticResource ReducedValueConverter},
ElementName=LeftPanel}"
                                Height="{Binding Width,
ConverterParameter=20, Converter={StaticResource ReducedValueConverter},
ElementName=LeftPanel}"
                                Margin="5"
                                Style="{StaticResource GlassButton}">
                        <i:Interaction.Triggers>
                            <i:EventTrigger
EventName="HandCursorClick">
                                <i:InvokeCommandAction
Command="{Binding ClickCommand}"/>
                            </i:EventTrigger>
                        </i:Interaction.Triggers>
                        <Image Source="{Binding Image,
Converter={StaticResource BitmapToImageSourceConverter}}">
                                Width="{Binding
Width, RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Height="{Binding
Height, RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Canvas.Top="{Binding (Canvas.Top), RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}"/>
                            </buttons:KinectButton>
                        </DataTemplate>

```



```

        </helpers:ScrollableCanvas.ItemTemplate>
    </helpers:ScrollableCanvas>
    <Canvas x:Name="BottomPanel"
        Width="570"
        Height="160"
        Opacity="0.2"
        Canvas.Left="230"
        Canvas.Top="580"
        Visibility="Collapsed" >

        <Canvas.Background>
            <LinearGradientBrush EndPoint="0.5,1"
StartPoint="0.5,0">
                <GradientStop Color="LightBlue" Offset="0.393"/>
                <GradientStop Color="Transparent"
Offset="0.723"/>
            </LinearGradientBrush>
        </Canvas.Background>
    </Canvas>
    <!--StackPanel que contiene todos los botones con fotos -->
    <helpers:ScrollableCanvasHorizontal
x:Name="BottomScrollableCanvas"
        ItemsSource="{Binding PhotoButtons,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}"
        ClipToBounds="True"
        Width="570"
        Height="{Binding ActualHeight,
ElementName=BottomPanel}"
        Canvas.Left="{Binding (Canvas.Left),
ElementName=BottomPanel}"
        Canvas.Top="{Binding (Canvas.Top),
ElementName=BottomPanel}"
        Margin="{Binding Margin,
ElementName=BottomPanel}" >
        <helpers:ScrollableCanvasHorizontal.ItemsPanel>
            <ItemsPanelTemplate>
                <StackPanel x:Name="BottomStackPanel"
Orientation="Horizontal" Canvas.Left="230"/>
            </ItemsPanelTemplate>
        </helpers:ScrollableCanvasHorizontal.ItemsPanel>
        <helpers:ScrollableCanvasHorizontal.ItemTemplate>
            <DataTemplate>
                <buttons:KinectButton Width="{Binding Width,
ConverterParameter=30, Converter={StaticResource ReducedValueConverter}},
ElementName=LeftPanel}"
                    Height="{Binding Width,
ConverterParameter=30, Converter={StaticResource ReducedValueConverter}},
ElementName=LeftPanel}"
                    Margin="5" >
                    <i:Interaction.Triggers>
                        <i:EventTrigger
EventName="HandCursorClick">
                            <i:InvokeCommandAction
Command="{Binding ClickCommand}" />
                        </i:EventTrigger>
                    </i:Interaction.Triggers>
            </DataTemplate>
        </helpers:ScrollableCanvasHorizontal.ItemTemplate>
    </helpers:ScrollableCanvasHorizontal.ItemsPanel>
    </helpers:ScrollableCanvasHorizontal>

```




```

        <Image Source="{Binding Image,
Converter={StaticResource BitmapToImageSourceConverter}}"
        Width="{Binding
Width, RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Height="{Binding
Height, RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

        Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

        Canvas.Top="{Binding (Canvas.Top), RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}"
        Margin="-5"/>
    </buttons:KinectButton>
</DataTemplate>
</helpers:ScrollableCanvasHorizontal.ItemTemplate>
</helpers:ScrollableCanvasHorizontal>
<!--StackPanel que contiene todos los botones del menú
superior-->
<StackPanel
        Orientation="Horizontal"
        Width="730"
        Height="80"
        Canvas.Left="{Binding ActualWidth,
ConverterParameter=20, Converter={StaticResource
IncreasedValueConverter}, ElementName=LeftPanel}"
        Canvas.Top="20"
        Visibility="{Binding
KinectService.KinectCameraImage, Converter={StaticResource
ValueToVisibilityConverter}}">
    <buttons:KinectRepeatableButton Width="70" Height="70"
Margin="5,0,5,5" Style="{StaticResource RoundButton}"
        DataContext="{Binding MenuButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <i:Interaction.Triggers>
            <i:EventTrigger eventName="HandCursorClick">
                <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
            </i:EventTrigger>
        </i:Interaction.Triggers>
        <Image Source="{Binding Image,
Converter={StaticResource BitmapToImageSourceConverter}}"
        Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Canvas.Left="{Binding
(Canvas.Left), RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

```

```

                Canvas.Top="{Binding
(Canvas.Top), RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
                </buttons:KinectRepeatableButton>
                <ItemsControl ItemsSource="{Binding
ChangeSizePositionButtons, Source={x:Static
topMenuButtons:TopMenuManager.Instance}}">
                    Visibility="{Binding
SizePositionButtonsVisibility, Source={x:Static
topMenuButtons:TopMenuManager.Instance}}">
                <ItemsControl.ItemsPanel>
                <ItemsPanelTemplate>
                <StackPanel Orientation="Horizontal"/>
                </ItemsPanelTemplate>
                </ItemsControl.ItemsPanel>
                <ItemsControl.ItemTemplate>
                <DataTemplate>
                    <buttons:KinectRepeatableButton Width="70"
Height="70" Margin="5,0,5,5" Style="{StaticResource RoundButton}">
                        <i:Interaction.Triggers>
                            <i:EventTrigger
EventName="HandCursorClick">
                                <i:InvokeCommandAction
Command="{Binding ClickCommand}" />
                                    </i:EventTrigger>
                                </i:Interaction.Triggers>
                                <Image Source="{Binding Image,
Converter={StaticResource BitmapToImageSourceConverter}}">
                                    Width="{Binding Width, RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}">
                                    Height="{Binding Height, RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}">
                                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}">
                                Canvas.Top="{Binding (Canvas.Top), RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}"/>
                                </buttons:KinectRepeatableButton>
                            </DataTemplate>
                        </ItemsControl.ItemTemplate>
                    </ItemsControl>
                    <buttons:KinectScreenshotButton x:Name="ScreenshotButton"
Width="70" Height="70" Margin="5,0,5,5" Style="{StaticResource
RoundButton}">
                        Visibility="{Binding
CameraButtonVisibility, Source={x:Static
topMenuButtons:TopMenuManager.Instance}}">
                        DataContext="{Binding CameraButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
                        AreSoundsOn="{Binding SoundsOn,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
                            <i:Interaction.Triggers>

```

```

        <i:EventTrigger EventName="HandCursorClick">
            <i:InvokeCommandAction Command="{Binding
ClickCommand}" />
        </i:EventTrigger>
    </i:Interaction.Triggers>
    <Image Source="{Binding Image,
Converter={StaticResource BitmapToImageSourceConverter}}"
        Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Canvas.Left="{Binding
(Canvas.Left), RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Canvas.Top="{Binding
(Canvas.Top), RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
    </buttons:KinectScreenshotButton>

    <ItemsControl ItemsSource="{Binding ActualTopMenuButtons,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <ItemsControl.ItemsPanel>
            <ItemsPanelTemplate>
                <StackPanel Orientation="Horizontal"/>
            </ItemsPanelTemplate>
        </ItemsControl.ItemsPanel>
        <ItemsControl.ItemTemplate>
            <DataTemplate>
                <buttons:KinectButton Width="70" Height="70"
Margin="5,0,5,5" Style="{StaticResource RoundButton}">
                    <i:Interaction.Triggers>
                        <i:EventTrigger
EventName="HandCursorClick">
                            <i:InvokeCommandAction
Command="{Binding ClickCommand}" />
                        </i:EventTrigger>
                    </i:Interaction.Triggers>
                    <Image Source="{Binding Image,
Converter={StaticResource BitmapToImageSourceConverter}}"
        Width="{Binding Width, RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}"
        Height="{Binding Height, RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}"
        Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Canvas.Top="{Binding (Canvas.Top), RelativeSource={RelativeSource
AncestorType={x:Type buttons:KinectButton}, Mode=FindAncestor}}"/>
                </buttons:KinectButton>
            </DataTemplate>
        </ItemsControl.ItemTemplate>
    </ItemsControl>

```

```

        </ItemsControl.ItemTemplate>
    </ItemsControl>
</StackPanel>
<!--StackPanel containing all the buttons from photos-->

</Canvas>
<!--Grids used for closing application-->
<Grid Width="{Binding ActualWidth, ElementName=ImageArea}"
Background="Black" Opacity="0.5"
        Visibility="{Binding CloseAppGridVisibility,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
    <TextBlock Text="¿Seguro que desea salir del programa?"
Opacity="1"
                FontSize="30" TextAlignment="Center"
VerticalAlignment="Center" Margin="0,0,0,200" Foreground="White"/>
    </Grid>
    <Grid x:Name="CloseAppGrid" Width="{Binding ActualWidth,
ElementName=ImageArea}"
        Visibility="{Binding CloseAppGridVisibility,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>
        <buttons:KinectButton Grid.Column="0" Width="70" Height="70"
Margin="5,0,5,5" Style="{StaticResource RoundButton}"
                DataContext="{Binding YesCloseButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
            <i:Interaction.Triggers>
                <i:EventTrigger EventName="HandCursorClick">
                    <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
                </i:EventTrigger>
            </i:Interaction.Triggers>
            <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
                    Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                    Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                    Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                    Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
        </buttons:KinectButton>
        <buttons:KinectButton Grid.Column="1" Width="70" Height="70"
Margin="5,0,5,5" Style="{StaticResource RoundButton}"
                DataContext="{Binding NoCloseButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
            <i:Interaction.Triggers>
                <i:EventTrigger EventName="HandCursorClick">

```

```

        <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
        </i:EventTrigger>
    </i:Interaction.Triggers>
    <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
        Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
        Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
    </buttons:KinectButton>
</Grid>
<!--Grids used for question screenshot application-->

    <Grid x:Name="QuestionScreenshotGrid" Width="{Binding
ActualWidth, ElementName=ImageArea}" Background="Black" Opacity="0.5"
        Visibility="Collapsed">
        <TextBlock Text="{?Desea tomarse otra foto?" Opacity="1"
            FontSize="30" TextAlignment="Center"
VerticalAlignment="Center" Margin="0,0,0,200" Foreground="White"/>
    </Grid>
    <Grid x:Name="QuestionScreenshotAppGrid" Width="{Binding
ActualWidth, ElementName=ImageArea}"
        Visibility="Collapsed">
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>
        <buttons:KinectScreenshotButton Grid.Column="0" Width="70"
Height="70" Margin="5,0,5,5" Style="{StaticResource RoundButton}"
            DataContext="{Binding
YesQuestionScreenshotButton, Source={x:Static
topMenuButtons:TopMenuManager.Instance}}">
            <i:Interaction.Triggers>
                <i:EventTrigger EventName="HandCursorClick">
                    <i:InvokeCommandAction Command="{Binding
ClickCommand}" />
                </i:EventTrigger>
            </i:Interaction.Triggers>
            <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
                Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

```



```

                                Canvas.Left="{Binding
(Canvas.Left), RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Canvas.Top="{Binding
(Canvas.Top), RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
                                </buttons:KinectScreenshotButton>
                                <buttons:KinectButton Grid.Column="1" Width="70" Height="70"
Margin="5,0,5,5" Style="{StaticResource RoundButton}"
                                DataContext="{Binding
NoQuestionScreenshotButton, Source={x:Static
topMenuButtons:TopMenuManager.Instance}}">
                                <i:Interaction.Triggers>
                                <i:EventTrigger EventName="HandCursorClick">
                                <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
                                </i:EventTrigger>
                                </i:Interaction.Triggers>
                                <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
                                Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                                Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
                                </buttons:KinectButton>
                                </Grid>
                                <!--Grids used for social medias application-->
                                <Grid Width="{Binding ActualWidth, ElementName=ImageArea}"
Background="Black" Opacity="0.5"
                                Visibility="{Binding SocialMediasAppGridVisibility,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
                                <TextBlock Text="Redes Sociales" Opacity="1"
                                FontSize="30" TextAlignment="Center"
VerticalAlignment="Center" Margin="0,0,0,200" Foreground="White"/>
                                </Grid>
                                <Grid x:Name="SocialMediasAppGrid" Width="{Binding ActualWidth,
ElementName=ImageArea}"
                                Visibility="{Binding SocialMediasAppGridVisibility,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
                                <Grid.ColumnDefinitions>
                                <ColumnDefinition/>
                                <ColumnDefinition/>
                                <ColumnDefinition/>
                                </Grid.ColumnDefinitions>
                                <!--<Button Content="Click me">
                                <Button.Margin>
                                <Thickness Left="10" Top="20" Right="10" Bottom="30"/>
                                </Button.Margin>

```



```

</Button>
    <Button Margin="10,20,10,30" Content="Click me"/>
    -->
    <buttons:KinectButton Grid.Column="0" Width="70" Height="70"
Margin="10,0,1,1" Style="{StaticResource RoundButton}"
        DataContext="{Binding FacebookButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <i:Interaction.Triggers>
            <i:EventTrigger EventName="HandCursorClick">
                <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
            </i:EventTrigger>
        </i:Interaction.Triggers>
        <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
            Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
    </buttons:KinectButton>
    <buttons:KinectButton Grid.Column="1" Width="70" Height="70"
Margin="1,0,1,1" Style="{StaticResource RoundButton}"
        DataContext="{Binding TwitterButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <i:Interaction.Triggers>
            <i:EventTrigger EventName="HandCursorClick">
                <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
            </i:EventTrigger>
        </i:Interaction.Triggers>
        <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
            Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
    </buttons:KinectButton>
    <buttons:KinectButton Grid.Column="2" Width="70" Height="70"
Margin="1,0,1,1" Style="{StaticResource RoundButton}"

```



```

                DataContext="{Binding YoutubeButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <i:Interaction.Triggers>
            <i:EventTrigger EventName="HandCursorClick">
                <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
            </i:EventTrigger>
        </i:Interaction.Triggers>
        <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
                Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
    </buttons:KinectButton>
</Grid>
<!--Grids used for TypeProduct application-->
<Grid Width="{Binding ActualWidth, ElementName=ImageArea}"
Background="Black" Opacity="0.5"
        Visibility="{Binding TypeProductAppGridVisibility,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
    <TextBlock Text="¡Selección de Talla!" Opacity="1"
                FontSize="30" TextAlignment="Center"
VerticalAlignment="Center" Margin="0,0,0,200" Foreground="White"/>
</Grid>
    <Grid x:Name="TypeProductAppGrid" Width="{Binding ActualWidth,
ElementName=ImageArea}"
        Visibility="{Binding TypeProductAppGridVisibility,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
            <ColumnDefinition/>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>

        <buttons:KinectButton Grid.Column="0" Width="70" Height="70"
Margin="10,0,1,1" Style="{StaticResource RoundButton}"
                DataContext="{Binding XSButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
            <i:Interaction.Triggers>
                <i:EventTrigger EventName="HandCursorClick">
                    <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
                </i:EventTrigger>
            </i:Interaction.Triggers>

```

```

        <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
            Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
    </buttons:KinectButton>
    <buttons:KinectButton Grid.Column="1" Width="70" Height="70"
Margin="1,0,1,1" Style="{StaticResource RoundButton}"
        DataContext="{Binding SButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <i:Interaction.Triggers>
            <i:EventTrigger EventName="HandCursorClick">
                <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
            </i:EventTrigger>
        </i:Interaction.Triggers>
        <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
            Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
            Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
    </buttons:KinectButton>
    <buttons:KinectButton Grid.Column="2" Width="70" Height="70"
Margin="1,0,1,1" Style="{StaticResource RoundButton}"
        DataContext="{Binding MButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
        <i:Interaction.Triggers>
            <i:EventTrigger EventName="HandCursorClick">
                <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
            </i:EventTrigger>
        </i:Interaction.Triggers>
        <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
            Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

```



```

                Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
            </buttons:KinectButton>
            <buttons:KinectButton Grid.Column="3" Width="70" Height="70"
Margin="1,0,1,1" Style="{StaticResource RoundButton}"
                DataContext="{Binding LButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
                <i:Interaction.Triggers>
                    <i:EventTrigger EventName="HandCursorClick">
                        <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
                    </i:EventTrigger>
                </i:Interaction.Triggers>
                <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
                Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
            </buttons:KinectButton>
            <buttons:KinectButton Grid.Column="4" Width="70" Height="70"
Margin="1,0,1,1" Style="{StaticResource RoundButton}"
                DataContext="{Binding XLButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
                <i:Interaction.Triggers>
                    <i:EventTrigger EventName="HandCursorClick">
                        <i:InvokeCommandAction Command="{Binding
ClickCommand}"/>
                    </i:EventTrigger>
                </i:Interaction.Triggers>
                <Image Source="{Binding Image, Converter={StaticResource
BitmapToImageSourceConverter}}"
                Width="{Binding Width,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Height="{Binding Height,
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"
                Canvas.Left="{Binding (Canvas.Left),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"

```

```

                Canvas.Top="{Binding (Canvas.Top),
RelativeSource={RelativeSource AncestorType={x:Type
buttons:KinectButton}, Mode=FindAncestor}}"/>
            </buttons:KinectButton>
        </Grid>
        <!--Hand Cursor canvas-->
        <Canvas Width="{Binding KinectService.Width}" Height="{Binding
KinectService.Height}">
            <Canvas x:Name="HandCursor" Width="70" Height="70"
Visibility="Collapsed">
                <Ellipse Width="{Binding ActualWidth,
RelativeSource={RelativeSource AncestorType={x:Type Canvas}}}"
                    Height="{Binding ActualHeight,
RelativeSource={RelativeSource AncestorType={x:Type Canvas}}}"
                    Fill="GhostWhite"
                    Opacity="0.5"
                    Stroke="Black"
                    StrokeThickness="2"/>
                <Image Source="/Resources/Icons/Hand.png" Width="60"
Height="60" Margin="5,5,5,5"/>
            </Canvas>
        </Canvas>
        <!--Grid containing Skeleton parts for debugging-->
        <ItemsControl ItemsSource="{Binding
KinectService.SkeletonManager.SkeletonParts}"
            Visibility="{Binding DebugModeOn,
Converter={StaticResource BoolToVisibilityConverter}}">
            <ItemsControl.ItemsPanel>
                <ItemsPanelTemplate>
                    <Grid Width="{Binding KinectService.Width}"
Height="{Binding KinectService.Height}"/>
                </ItemsPanelTemplate>
            </ItemsControl.ItemsPanel>
        </ItemsControl>
        <!--Grid used for Code QR-->

        <Grid x:Name="QRCodeGrid" Opacity="0.5" Background="Black"
Visibility="Collapsed">
            <!-- <buttons:KinectButton Width="160" Height="160"
Margin="5,0,5,5" Style="{StaticResource RectangleButton}"
                DataContext="{Binding QRCodeButton,
Source={x:Static topMenuButtons:TopMenuManager.Instance}}">
                <qr:QrCodeImgControl x:Name="qrControl"
HorizontalAlignment="Stretch" VerticalAlignment="Stretch" />
            </buttons:KinectButton-->
            <qr:QrCodeImgControl Width="250" Height="250"
x:Name="qrControl" />
        </Grid>
        <!--Grid used for taking screenshots-->
        <Grid x:Name="ScreenshotGrid" Opacity="0.7" Background="Black"
Visibility="Collapsed">
            <TextBlock x:Name="ScreenshotText" FontSize="40"
HorizontalAlignment="Center" Opacity="0.7" VerticalAlignment="Center"
Foreground="White"/>
        </Grid>
        <!--Grid used for viewing error messages-->
    
```



```
<Grid x:Name="ErrorGrid" Opacity="0.7" Background="Black"
Visibility="{Binding KinectService.ErrorGridVisibility}">
    <TextBlock Text="{Binding KinectService.ErrorGridMessage}"
                FontSize="20" TextAlignment="Center"
Opacity="0.7" VerticalAlignment="Center" Foreground="White"/>
    </Grid>
</Grid>
</Window>
```

VII. Código Fuente del almacenamiento y selección de archivos de las prendas (Textura, Obj., etc.).

- ✓ En la aplicación para que se extrae los archivos de las prendas a partir de la base de datos, es primero conecta a la base de datos SQL Server y realizar consultas sobre las tablas que se corresponde.

En la capa Model >Conexión> DataSQLServerClass.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Data;
```

```
namespace appProbadorVirtual3D.Model.Conexion
```

```
{
    class DataSQLServerClass
    {
        #region Variables y Propiedades
        private string cadena = string.Empty;

        public SqlConnection oConnection { get; set; }

        public SqlCommand oComand { get; set; }

        public IDataReader Lector { get; set; }
        #endregion

        #region Constructor
        public DataSQLServerClass()
        {
            cadena = @"Data Source=JOSEPH;Initial
Catalog=db_RopaVirtual3D;Integrated Security=Yes";
            oConnection = new SqlConnection();
            oComand = new SqlCommand();
        }
        #endregion

        #region metodos
        public void Open() {
```



```

if (oConnection.State == ConnectionState.Open)
{
    return;
}
oConnection.ConnectionString = cadena;

try
{
    oConnection.Open();
}
catch (Exception ex)
{
    throw ex;
}
finally {
}
}

public void Close() {
    if(oConnection.State== ConnectionState.Closed){
        return;
    }

    oConnection.Close();
}

//Este metodo
public      Int32      ExecuteNonQuery(CommandType
tipoComando, String consulta) {
    oComand.Connection = oConnection;
    oComand.CommandType = tipoComando;
    oComand.CommandText = consulta;

    int retorno = 0;

    try
    {
        retorno = oComand.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        throw ex;
    }
    return retorno;
}

public      IDataReader      ExecuteReader(CommandType
tipoComando, String consulta)

```

```

{
    Lector = null;
    oComand.Connection = oConnection;
    oComand.CommandType = tipoComando;
    oComand.CommandText = consulta;

    try
    {
        Lector = oComand.ExecuteReader();
    }
    catch (Exception ex)
    {
        throw ex;
    }
    return Lector;
}
#endregion
}
}

```

Código Fuente para la mapear los objetos capturados de la bd y asignar los datos en la aplicación como la textura el objeto ,etc.

```

public List<ClothingButtonViewModel> DaoClothing(Int32 dCategory_id)
{
    List<ClothingButtonViewModel> lClothing = new
List<ClothingButtonViewModel>();
    IDataReader reader = null;
    StringBuilder sb_clothing = new StringBuilder();

    sb_clothing.Append("select ");
    sb_clothing.Append("tb_categoria.tb_categoria_id
'categoria',");
    sb_clothing.Append("tb_categoria_nom,");
    sb_clothing.Append("tb_producto_id, ");
    sb_clothing.Append("tb_producto_nom,tb_producto_fot ,");

    sb_clothing.Append("tb_configuracion3d_bbjts,tb_configuracion3d_vbjts,");

    sb_clothing.Append("tb_configuracion3d_bdel,tb_configuracion3d_vdel,");

    sb_clothing.Append("tb_configuracion3d_brat,tb_configuracion3d_vrat ,");
    sb_clothing.Append("tb_genero.tb_genero_id 'genero',");
    sb_clothing.Append("tb_genero_nom ");
    sb_clothing.Append("from ");
    sb_clothing.Append("tb_detallecategoria ");
    sb_clothing.Append("inner join tb_categoria ");
    sb_clothing.Append("on
tb_categoria.tb_categoria_id=tb_detallecategoria.tb_categoria_id ");
    sb_clothing.Append("inner join tb_producto ");
    sb_clothing.Append("on
tb_detallecategoria.tb_detallecategoria_id=tb_producto.tb_detallecategori
a_id ");
}
}

```

```

        sb_clothing.Append("inner join tb_configuracion3d ");
        sb_clothing.Append("on
tb_configuracion3d.tb_configuracion3d_id=tb_producto.tb_configuracion3d_i
d ");
        sb_clothing.Append("inner join tb_genero ");
        sb_clothing.Append("on
tb_genero.tb_genero_id=tb_detallecategoria.tb_genero_id ");
        sb_clothing.Append("where
tb_producto.tb_detallecategoria_id=" + dCategory_id + "");
        // int cantidad = 0;
        try
        {
            DataSQLServerClass oSQLServerDatos = new
DataSQLServerClass();
            oSQLServerDatos.Open();
            String query = sb_clothing.ToString();
            reader = oSQLServerDatos.ExecuteReader(CommandType.Text,
query);

            // string lines = "\r\n.";
            while (reader.Read())
            {
                // cantidad++;
                String genero = reader["genero"].ToString();
                String genero_nom =
reader["tb_genero_nom"].ToString();
                Int32 category_id = (Int32)reader["categoria"];
                Int32 clothing_id = (Int32)reader["tb_producto_id"];
                String category_nom =
reader["tb_categoria_nom"].ToString();
                String clothing_fot =
reader["tb_producto_fot"].ToString();
                Boolean bbjts =
(Boolean)reader["tb_configuracion3d_bbjts"];
                Boolean bdel =
(Boolean)reader["tb_configuracion3d_bdel"];
                Boolean brat =
(Boolean)reader["tb_configuracion3d_brat"];

                String vbjts =
DBNull.Value.Equals(reader["tb_configuracion3d_vbjts"]) ? "" :
reader["tb_configuracion3d_vbjts"].ToString();
                JointType Jbjts = JointType.HipRight;
                Double vdel =
DBNull.Value.Equals(reader["tb_configuracion3d_vdel"]) ? 0 :
(Double)reader["tb_configuracion3d_vdel"];
                Double vrat =
DBNull.Value.Equals(reader["tb_configuracion3d_vrat"]) ? 0 :
(Double)reader["tb_configuracion3d_vrat"];

                Bitmap ImageClothing;
                switch (vbjts)
                {
                    case "FootLeft":Jbjts = JointType.FootLeft;break;
                    case "FootRight":Jbjts =
JointType.FootRight;break;

```

```

                case "AnkleLeft":Jbjts =
JointType.AnkleLeft;break;
                case "AnkleRight":Jbjts =
JointType.AnkleRight;break;
                case "ElbowLeft":Jbjts = JointType.ElbowLeft;
break;
                case "ElbowRight":Jbjts =
JointType.ElbowRight;break;
                case "HandLeft":Jbjts = JointType.HandLeft;break;
                case "HandRight":Jbjts =
JointType.HandRight;break;
                case "Head": Jbjts = JointType.Head;break;
                case "HipCenter":Jbjts =
JointType.HipRight;break;
                case "HipLeft":Jbjts = JointType.HipLeft; break;
                case "KneeLeft":Jbjts = JointType.KneeLeft;break;
                case "KneeRight":Jbjts =
JointType.KneeRight;break;
                case "ShoulderLeft":Jbjts =
JointType.ShoulderLeft; break;
                case "ShoulderRight":Jbjts =
JointType.ShoulderRight;break;
                case "SpineBase":Jbjts =
JointType.SpineBase;break;
                case "WristLeft":Jbjts = JointType.WristLeft;
break;
                case "WristRight":Jbjts =
JointType.WristRight;break;
                default:break;
            }

            string path2 =
"Resources\\Images\\"+category_nom+"\\"+clothing_fot+".png";
            string path3 = Path.GetFullPath(path2);

```

Esto sería agregar los productos en la categoría y en todas las categorías, se muestra en las imágenes algunas condiciones.

```

Uri uriSource = new Uri(path3);
ImageClothing = new Bitmap(path3);

// ImageClothing =
(Bitmap)Properties.Resources.ResourceManager.GetObject(clothing_fot);
ClothingButtonViewModel oCC = null;

if (category_id == 1 && category_nom == "Blouses")
{
    BlouseButtonViewModel objeto = null;

```

```

        if (bbjts && brat && bdel) objeto = new
BlouseButtonViewModel(ClothingItemBase.ClothingType.BlouseItem,
@".\Resources\Models\Blouses\" + clothing_fot + ".obj") { Id=clothing_id,
Image = ImageClothing, BottomJointToTrackScale = Jbjts, Ratio = vrat,
DeltaY = vdel };
        else if (bbjts && brat) objeto = new
BlouseButtonViewModel(ClothingItemBase.ClothingType.BlouseItem,
@".\Resources\Models\Blouses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat };
        else if (bdel && brat) objeto = new
BlouseButtonViewModel(ClothingItemBase.ClothingType.BlouseItem,
@".\Resources\Models\Blouses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat };
        else if (bbjts && brat) objeto = new
BlouseButtonViewModel(ClothingItemBase.ClothingType.BlouseItem,
@".\Resources\Models\Blouses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat };
        else if (brat) objeto = new
BlouseButtonViewModel(ClothingItemBase.ClothingType.BlouseItem,
@".\Resources\Models\Blouses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, Ratio = vrat };
        else if (bdel) objeto = new
BlouseButtonViewModel(ClothingItemBase.ClothingType.BlouseItem,
@".\Resources\Models\Blouses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel };
        else if (bbjts) objeto = new
BlouseButtonViewModel(ClothingItemBase.ClothingType.BlouseItem,
@".\Resources\Models\Blouses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts };
        else objeto = new
BlouseButtonViewModel(ClothingItemBase.ClothingType.BlouseItem,
@".\Resources\Models\Blouses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing };
        oCC = objeto;
    }else if (category_id == 2 && category_nom ==
"Coats"){
        CoatButtonViewModel objeto = null;
        if (bbjts && brat && bdel) new
CoatButtonViewModel(ClothingItemBase.ClothingType.CoatItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Coats\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat, DeltaY = vdel };
        else if (bdel && brat) objeto = new
CoatButtonViewModel(ClothingItemBase.ClothingType.CoatItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Coats\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel, Ratio = vrat };
        else if (bdel && bbjts) objeto = new
CoatButtonViewModel(ClothingItemBase.ClothingType.CoatItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?

```




```

ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Coats\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, DeltaY = vdel };
        else if (bbjts && brat) objeto = new
CoatButtonViewModel(ClothingItemBase.ClothingType.CoatItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Coats\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat };
        else if (bbjts) objeto = new
CoatButtonViewModel(ClothingItemBase.ClothingType.CoatItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Coats\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts };
        else if (bdel) objeto = new
CoatButtonViewModel(ClothingItemBase.ClothingType.CoatItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Coats\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel };
        else if (brat) objeto = new
CoatButtonViewModel(ClothingItemBase.ClothingType.CoatItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Coats\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, Ratio =
vrat };
        else objeto = new
CoatButtonViewModel(ClothingItemBase.ClothingType.CoatItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Coats\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing };
        }else if (category_id == 4 && category_nom ==
"Dresses"){
                DressButtonViewModel objeto = null;
                if (bbjts && brat && bdel) objeto = new
DressButtonViewModel(ClothingItemBase.ClothingType.DressItem,
@".\Resources\Models\Dresses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat, DeltaY = vdel };
                else if (bdel && brat) objeto = new
DressButtonViewModel(ClothingItemBase.ClothingType.DressItem,
@".\Resources\Models\Dresses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel, Ratio = vrat };
                else if (bdel && bbjts) objeto = new
DressButtonViewModel(ClothingItemBase.ClothingType.DressItem,
@".\Resources\Models\Dresses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
DeltaY = vdel };

```




```

        else if (bbjts && brat) objeto = new
DressButtonViewModel(ClothingItemBase.ClothingType.DressItem,
@".\Resources\Models\Dresses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat };
        else if (bbjts) objeto = new
DressButtonViewModel(ClothingItemBase.ClothingType.DressItem,
@".\Resources\Models\Dresses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts };
        else if (bdel) objeto = new
DressButtonViewModel(ClothingItemBase.ClothingType.DressItem,
@".\Resources\Models\Dresses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel };
        else if (brat) objeto = new
DressButtonViewModel(ClothingItemBase.ClothingType.DressItem,
@".\Resources\Models\Dresses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, Ratio = vrat };
        else objeto = new
DressButtonViewModel(ClothingItemBase.ClothingType.DressItem,
@".\Resources\Models\Dresses\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing };
        oCC = objeto;
    }else if (category_id == 6 && category_nom ==
"Jackets"){
        JacketButtonViewModel objeto = null;

        if (bbjts && brat && bdel) new
JacketButtonViewModel(ClothingItemBase.ClothingType.JacketItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Jackets\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat, DeltaY = vdel };
        else if (bdel && brat) objeto = new
JacketButtonViewModel(ClothingItemBase.ClothingType.JacketItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Jackets\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel, Ratio = vrat };
        else if (bdel && bbjts) objeto = new
JacketButtonViewModel(ClothingItemBase.ClothingType.JacketItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Jackets\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, DeltaY = vdel };
        else if (bbjts && brat) objeto = new
JacketButtonViewModel(ClothingItemBase.ClothingType.JacketItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Jackets\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat };
        else if (bbjts) objeto = new
JacketButtonViewModel(ClothingItemBase.ClothingType.JacketItem, (genero

```



```

== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Jackets\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts };
    else if (bdel) objeto = new
JacketButtonViewModel(ClothingItemBase.ClothingType.JacketItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Jackets\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel };
    else if (brat) objeto = new
JacketButtonViewModel(ClothingItemBase.ClothingType.JacketItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Jackets\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, Ratio =
vrat };
    else objeto = new
JacketButtonViewModel(ClothingItemBase.ClothingType.JacketItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Jackets\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing };
        occ = objeto;
    } else if (category_id == 7 && category_nom ==
"Pants") {
        PantButtonViewModel objeto = null;

        if (bbjts && brat && bdel) new
PantButtonViewModel(ClothingItemBase.ClothingType.PantItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Pants\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat, DeltaY = vdel };
        else if (bdel && brat) objeto = new
PantButtonViewModel(ClothingItemBase.ClothingType.PantItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Pants\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel, Ratio = vrat };
        else if (bdel && bbjts) objeto = new
PantButtonViewModel(ClothingItemBase.ClothingType.PantItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Pants\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, DeltaY = vdel };
        else if (bbjts && brat) objeto = new
PantButtonViewModel(ClothingItemBase.ClothingType.PantItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Pants\" +

```



```

clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat };
        else if (bbjts) objeto = new
PantButtonViewModel(ClothingItemBase.ClothingType.PantItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Pants\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts };
        else if (bdel) objeto = new
PantButtonViewModel(ClothingItemBase.ClothingType.PantItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Pants\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel };
        else if (brat) objeto = new
PantButtonViewModel(ClothingItemBase.ClothingType.PantItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Pants\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, Ratio =
vrat };
        else objeto = new
PantButtonViewModel(ClothingItemBase.ClothingType.PantItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Pants\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing };
        oCC = objeto;
    } else if (category_id == 9 && category_nom ==
"Shirts"){
        ShirtButtonViewModel objeto = null;

        if (bbjts && brat && bdel) objeto = new
ShirtButtonViewModel(ClothingItemBase.ClothingType.ShirtItem,
@".\Resources\Models\Shirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat, DeltaY = vdel };
        else if (bdel && brat) objeto = new
ShirtButtonViewModel(ClothingItemBase.ClothingType.ShirtItem,
@".\Resources\Models\Shirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel, Ratio = vrat };
        else if (bdel && bbjts) objeto = new
ShirtButtonViewModel(ClothingItemBase.ClothingType.ShirtItem,
@".\Resources\Models\Shirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
DeltaY = vdel };
        else if (bbjts && brat) objeto = new
ShirtButtonViewModel(ClothingItemBase.ClothingType.ShirtItem,
@".\Resources\Models\Shirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat };
        else if (bbjts) objeto = new
ShirtButtonViewModel(ClothingItemBase.ClothingType.ShirtItem,

```

```

@".\Resources\Models\Shirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts };
        else if (bdel) objeto = new
ShirtButtonViewModel(ClothingItemBase.ClothingType.ShirtItem,
@".\Resources\Models\Shirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel };
        else if (brat) objeto = new
ShirtButtonViewModel(ClothingItemBase.ClothingType.ShirtItem,
@".\Resources\Models\Shirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, Ratio = vrat };
        else objeto = new
ShirtButtonViewModel(ClothingItemBase.ClothingType.ShirtItem,
@".\Resources\Models\Shirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing };
        oCC = objeto;
    }else if (category_id == 10 && category_nom ==
"Skirts"){
        SkirtButtonViewModel objeto = null;
        if (bbjts && brat && bdel) objeto =
new SkirtButtonViewModel(ClothingItemBase.ClothingType.SkirtItem,
@".\Resources\Models\Skirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat, DeltaY = vdel };
        else if (bdel && brat) objeto = new
SkirtButtonViewModel(ClothingItemBase.ClothingType.SkirtItem,
@".\Resources\Models\Skirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel, Ratio = vrat };
        else if (bdel && bbjts) objeto = new
SkirtButtonViewModel(ClothingItemBase.ClothingType.SkirtItem,
@".\Resources\Models\Skirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
DeltaY = vdel };
        else if (bbjts && brat) objeto = new
SkirtButtonViewModel(ClothingItemBase.ClothingType.SkirtItem,
@".\Resources\Models\Skirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat };
        else if (bbjts) objeto = new
SkirtButtonViewModel(ClothingItemBase.ClothingType.SkirtItem,
@".\Resources\Models\Skirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts };
        else if (bdel) objeto = new
SkirtButtonViewModel(ClothingItemBase.ClothingType.SkirtItem,
@".\Resources\Models\Skirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel };
        else if (brat) objeto = new
SkirtButtonViewModel(ClothingItemBase.ClothingType.SkirtItem,
@".\Resources\Models\Skirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, Ratio = vrat };
        else objeto = new
SkirtButtonViewModel(ClothingItemBase.ClothingType.SkirtItem,
@".\Resources\Models\Skirts\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing };
        oCC = objeto;
    }

```



```

}else if (category_id == 11 && category_nom ==
"Sweaters") {
    SweaterButtonViewModel objeto =
null;

    if (bbjts && brat && bdel) objeto
= new SweaterButtonViewModel(ClothingItemBase.ClothingType.SweaterItem,
@".\Resources\Models\Sweaters\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat, DeltaY = vdel };
    else if (bdel && brat) objeto =
new SweaterButtonViewModel(ClothingItemBase.ClothingType.SweaterItem,
@".\Resources\Models\Sweaters\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel, Ratio = vrat };
    else if (bdel && bbjts) objeto =
new SweaterButtonViewModel(ClothingItemBase.ClothingType.SweaterItem,
@".\Resources\Models\Sweaters\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
DeltaY = vdel };
    else if (bbjts && brat) objeto =
new SweaterButtonViewModel(ClothingItemBase.ClothingType.SweaterItem,
@".\Resources\Models\Sweaters\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts,
Ratio = vrat };
    else if (bbjts) objeto = new
SweaterButtonViewModel(ClothingItemBase.ClothingType.SweaterItem,
@".\Resources\Models\Sweaters\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, BottomJointToTrackScale = Jbjts };
    else if (bdel) objeto = new
SweaterButtonViewModel(ClothingItemBase.ClothingType.SweaterItem,
@".\Resources\Models\Sweaters\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, DeltaY = vdel };
    else if (brat) objeto = new
SweaterButtonViewModel(ClothingItemBase.ClothingType.SweaterItem,
@".\Resources\Models\Sweaters\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing, Ratio = vrat };
    else objeto = new
SweaterButtonViewModel(ClothingItemBase.ClothingType.SweaterItem,
@".\Resources\Models\Sweaters\" + clothing_fot + ".obj") { Id =
clothing_id, Image = ImageClothing };
    oCC = objeto;
}else if (category_id == 12 && category_nom ==
"Tops"){
    TopButtonViewModel objeto =
null;

    if (bbjts && brat && bdel)
objeto = new TopButtonViewModel(ClothingItemBase.ClothingType.TopItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tops\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat, DeltaY = vdel };
    else if (bbjts && brat)
objeto = new TopButtonViewModel(ClothingItemBase.ClothingType.TopItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :

```



```

ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tops\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat };
                else if (bdel && brat) objeto
= new TopButtonViewModel(ClothingItemBase.ClothingType.TopItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tops\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel, Ratio = vrat };
                else if (bdel && bbjts)
objeto = new TopButtonViewModel(ClothingItemBase.ClothingType.TopItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tops\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, DeltaY = vdel };
                else if (bbjts) objeto = new
TopButtonViewModel(ClothingItemBase.ClothingType.TopItem, (genero == "1"
? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tops\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts };
                else if (bdel) objeto = new
TopButtonViewModel(ClothingItemBase.ClothingType.TopItem, (genero == "1"
? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tops\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel };
                else if (brat) objeto = new
TopButtonViewModel(ClothingItemBase.ClothingType.TopItem, (genero == "1"
? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tops\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, Ratio =
vrat };
                else objeto = new
TopButtonViewModel(ClothingItemBase.ClothingType.TopItem, (genero == "1"
? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tops\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing };
                oCC = objeto;
            }else if (category_id == 13 && category_nom ==
"Tshirts"){
                TshirtButtonViewModel
objeto = null;
                if (bbjts && brat &&
bdel) new TshirtButtonViewModel(ClothingItemBase.ClothingType.TshirtItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tshirts\" +

```




```

clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat, DeltaY = vdel };
                                else if (bdel && brat)
objeto = new
TshirtButtonViewModel(ClothingItemBase.ClothingType.TshirtItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tshirts\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel, Ratio = vrat };
                                else if (bdel && bbjts)
objeto = new
TshirtButtonViewModel(ClothingItemBase.ClothingType.TshirtItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tshirts\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, DeltaY = vdel };
                                else if (bbjts && brat)
objeto = new
TshirtButtonViewModel(ClothingItemBase.ClothingType.TshirtItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tshirts\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat };
                                else if (bbjts) objeto =
new TshirtButtonViewModel(ClothingItemBase.ClothingType.TshirtItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tshirts\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts };
                                else if (bdel) objeto =
new TshirtButtonViewModel(ClothingItemBase.ClothingType.TshirtItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tshirts\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel };
                                else if (brat) objeto =
new TshirtButtonViewModel(ClothingItemBase.ClothingType.TshirtItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tshirts\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, Ratio =
vrat };
                                else objeto = new
TshirtButtonViewModel(ClothingItemBase.ClothingType.TshirtItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Tshirts\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing };
                                occ = objeto;
                                } else if (category_id == 14 && category_nom ==
"Suits") {

```



SuitButtonViewModel

```

objeto = null;

        if (bbjts && brat &&
bdel) new SuitButtonViewModel(ClothingItemBase.ClothingType.SuitItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Suits\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat, DeltaY = vdel };
        else if (bdel &&
brat) objeto = new
SuitButtonViewModel(ClothingItemBase.ClothingType.SuitItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Suits\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel, Ratio = vrat };
        else if (bdel &&
bbjts) objeto = new
SuitButtonViewModel(ClothingItemBase.ClothingType.SuitItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Suits\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, DeltaY = vdel };
        else if (bbjts &&
brat) objeto = new
SuitButtonViewModel(ClothingItemBase.ClothingType.SuitItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Suits\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts, Ratio = vrat };
        else if (bbjts)
objeto = new SuitButtonViewModel(ClothingItemBase.ClothingType.SuitItem,
(genero == "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2"
? ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Suits\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing,
BottomJointToTrackScale = Jbjts };
        else if (bdel) objeto
= new SuitButtonViewModel(ClothingItemBase.ClothingType.SuitItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Suits\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, DeltaY
= vdel };
        else if (brat) objeto
= new SuitButtonViewModel(ClothingItemBase.ClothingType.SuitItem, (genero
== "1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Suits\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing, Ratio =
vrat };
    
```



```

else objeto = new
SuitButtonViewModel(ClothingItemBase.ClothingType.SuitItem, (genero ==
"1" ? ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
ClothingItemBase.MaleFemaleType.Male :
ClothingItemBase.MaleFemaleType.Both)), @".\Resources\Models\Suits\" +
clothing_fot + ".obj") { Id = clothing_id, Image = ImageClothing };
oCC = objeto;
}

        lClothing.Add(oCC);
    }
    reader.Close();
    return lClothing;
}
catch (Exception ex)
{
    // int valor = cantidad;
    throw ex;
}
finally
{
}
}

public List<ClothingCategoryButtonViewModel>
DaoClothingCategory()
{
    List<ClothingCategoryButtonViewModel> lClothingCategory = new
List<ClothingCategoryButtonViewModel>();
    DataSQLServerClass oSQLServerDatos = new
DataSQLServerClass();
    IDataReader reader = null;
    StringBuilder sb_category = new StringBuilder();

    sb_category.Append("select ");
    sb_category.Append("* ");
    sb_category.Append("from ");
    sb_category.Append("tb_detallecategoria");
    try
    {

        oSQLServerDatos.Open();
        reader = oSQLServerDatos.ExecuteReader(CommandType.Text,
sb_category.ToString());

        while (reader.Read())
        {
            String genero = reader["tb_genero_id"].ToString();
            Int32 dcategory_id =
(Int32)reader["tb_detallecategoria_id"];
            String dcategory_fot =
reader["tb_detallecategoria_fot"].ToString();
            Bitmap ImageCategory;

```

```

        List<ClothingButtonViewModel> lClothing =
        DaoClothing(dcategory_id);
        ImageCategory =
        (Bitmap)Properties.Resources.ResourceManager.GetObject(dcategory_fot);

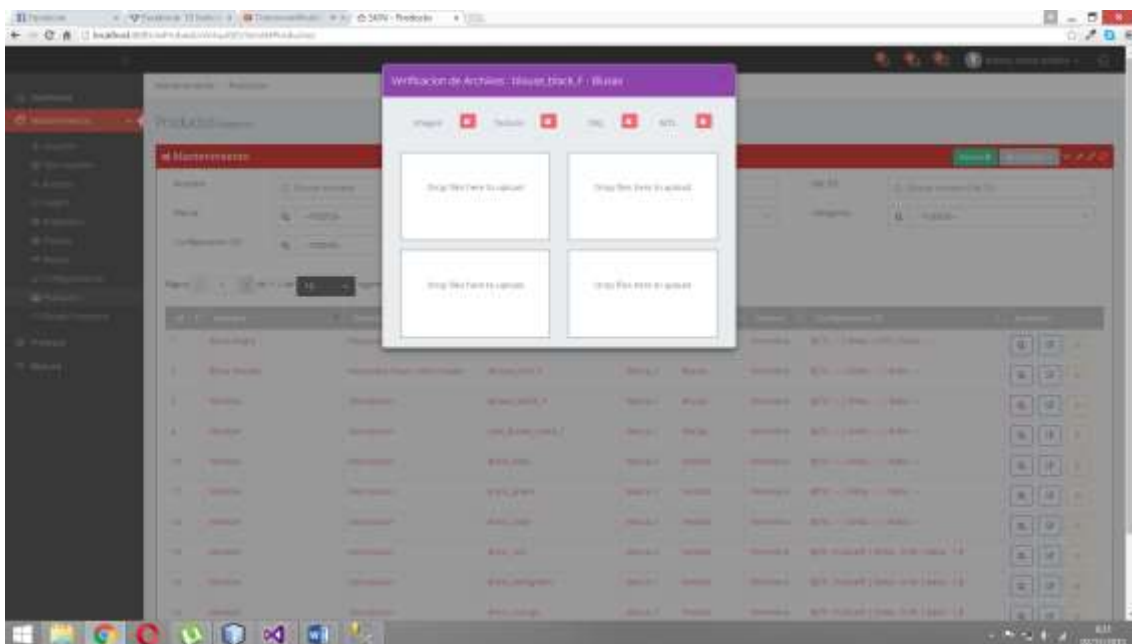
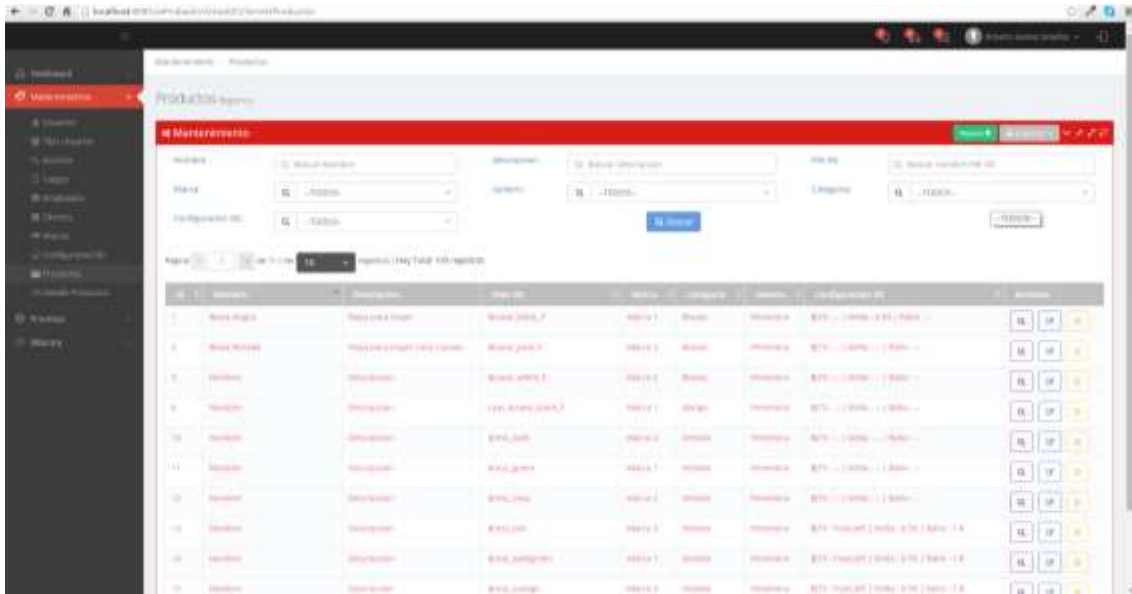
        ClothingCategoryButtonViewModel oCC = new
        ClothingCategoryButtonViewModel
            (genero == "1" ?
        ClothingItemBase.MaleFemaleType.Female : (genero == "2" ?
        ClothingItemBase.MaleFemaleType.Male :
        ClothingItemBase.MaleFemaleType.Both))
        {
            Image = ImageCategory,
            Clothes = lClothing
        };

        lClothingCategory.Add(oCC);
    }
    reader.Close();
    return lClothingCategory;
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    oSQLServerDatos.Close();
}
}
#endregion Private Methods
}

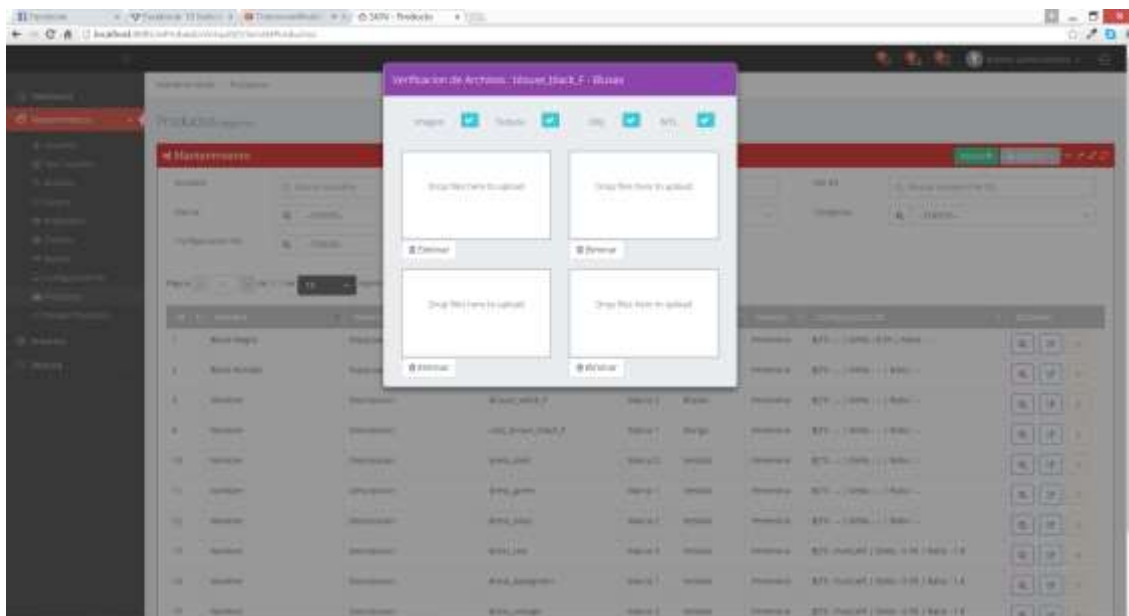
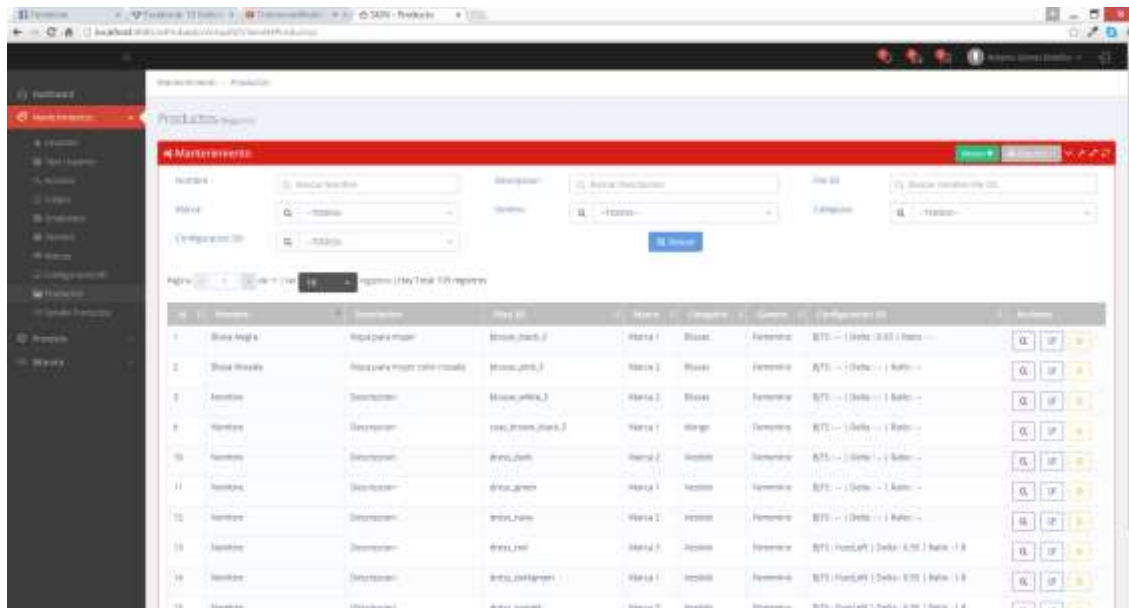
```

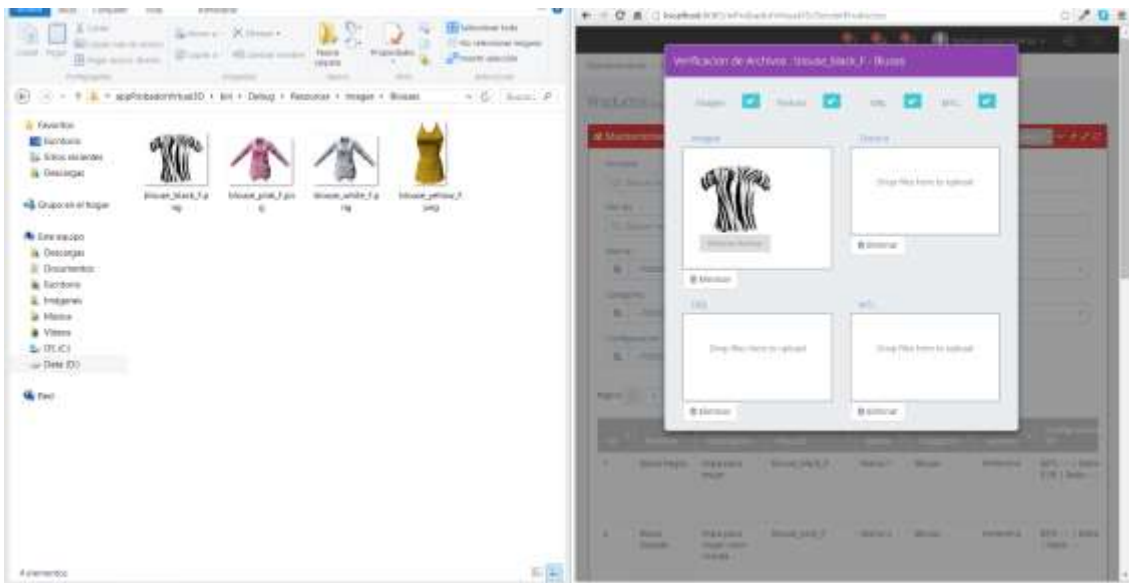
VIII. Código Fuente de alimentación de la información de las prendas.

En la aplicación web estas serían las imágenes de alimentación del panel de control de los archivos de las prendas como la textura, imagen, obj., mtl. La cual el rojo nos indica que no existe archivo o algún archivo no se encuentra en la ruta de la aplicación como, ruta de la aplicación de probador virtual y ruta dentro de la web.



Parte de que nos indica que existe las imágenes o archivos involucrados a las prendas.





Código Fuente de este módulo de la web.

- ✓ En la parte de la Vista con Javascript. Se utilizó la librería Dropzone y se validó también, el tipo de archivo que se debe colocar en cada ítem por ejemplo en el primero que son las imágenes, segundo la textura (cualquier imagen), tipo obj, y el mtl que da el diseño al obj. 3D.

```
function addDropzone(val) {
    var dropZoneld = "my-dropzone-" + val;
    Dropzone.autoDiscover = false; // De lo contrario será
    inicializado dos veces
    var extension = "";
    if (val === 1)
        extension = ".png";
    if (val === 2)
        extension = "image/*";
    if (val === 3)
        extension = ".obj";
    if (val === 4)
        extension = ".mtl";
```

```
var myDropzone = new Dropzone("#" + dropZoneld, {
    maxFiles: 1,
    maxFileSize: 3,
    acceptedFiles: extension,
    url: "ServletProductos",
    params: {"accion": "archivo", "tipo": val},
```



```

success: function (file, response) {
    var _this = this;
    //          $("##static_files").on("click",function(e){
eliminarFileDZ(e,_this,file);});
    console.log(response);
    var datos = eval('(' + response + ')');
    console.log(datos);
    console.log(datos.msg);
    if (datos.msg === "V1") { // error
        file.previewElement.classList.add("dz-error");
        return file.previewElement.querySelector("[data-dz-
errormessage]").textContent = "Error al cargar";
    } else if (datos.msg === "V2") { // error
        file.previewElement.classList.add("dz-error");
        return file.previewElement.querySelector("[data-dz-
errormessage]").textContent = "Tipo de Archivo Incorrecto";
    } else if (datos.msg === "OK") {
        dataTable.ajax.reload();
        verFiles3D(datos.producto, false);
        return file.previewElement.classList.add("dz-success");
    }
    return;
},
init: function () {
    this.on("addedfile", function (file) {
        var removeButton = Dropzone.createElement("<button
class='btn btn-sm btn-block' data-dz-remove >Eliminar
Archivo</button>");
        var _this = this;
        removeButton.addEventListener("click", function (e) {
            e.preventDefault();
            e.stopPropagation();
            _this.removeFile(file);
        });
        file.previewElement.appendChild(removeButton);
    });
}
});
}
}

```



✓ En la parte del controlador : Servlet seria:

```

} else if (ServletFileUpload.isMultipartContent(request)) {
    subirArchivo3D(request, response);
} else {
    cargarDatosReferencia(session, request);
    request.getRequestDispatcher("productos.jsp").forward(request, response);
}
} else {

```

Este código en Java procesa la imagen .

```

private void subirArchivo3D(HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    PrintWriter out = response.getWriter();
    Gson gson = new Gson();
    Map records = new HashMap<>();

    try {
        FileItemFactory factory = new DiskFileItemFactory();
        ServletFileUpload upload = new ServletFileUpload(factory);
        List items = upload.parseRequest(request);
        Iterator iterator = items.iterator();

        String accion = "";
        String tipo = "";
        HttpSession session = request.getSession();
        LogicProducto logicProducto = new LogicProducto();
        Producto producto = (Producto)
session.getAttribute("sesProductoFiles");
        while (iterator.hasNext()) {
            FileItem item = (FileItem) iterator.next();
            if (!item.isFormField()) {
                String root = getServletContext().getRealPath("");
                String fileName = item.getName();
                String fileNameUpload = producto.getFoto();
                String extension = Cadenas.getExtension(fileName);
                if (validarExtension(tipo, extension)) {
                    if (accion.equals("archivo")) {
                        File uploadedFileWeb = null;
                        String rutaAppGeneral =
"/../..../appProbadorVirtual3D/bin/Debug/Resources";
                        String tipoRuta = "";
                        String categoria =
producto.getDetalleCategoria().getCategoria().getNombre();
                        switch (tipo) {
                            case "1":

```

```

        tipoRuta = "/Images/" + categoria;
        File pathW = new File(root +
"/../web/img/CategoriaRopa/" + categoria);
        uploadedFileWeb = new File(pathW + "/" +
fileNameUpload + extension);
        File path = new File(root + rutaAppGeneral +
tipoRuta);
        File uploadedFile = new File(path + "/" +
fileNameUpload + extension);
        item.write(uploadedFile);

copyFile_Java7(uploadedFile.getAbsolutePath(),
uploadedFileWeb.getAbsolutePath());
        break;
    case "2":
        tipoRuta = "/Materials";
        producto.setMaterialArchivo(fileNameUpload +
extension);
        path = new File(root + rutaAppGeneral +
tipoRuta);
        uploadedFile = new File(path + "/" +
fileNameUpload + extension);
        item.write(uploadedFile);

    logicProducto.modificarMaterialProducto(producto);
        break;
    case "3":
    case "4":
        tipoRuta = "/Models/" + categoria;
        path = new File(root + rutaAppGeneral +
tipoRuta);
        uploadedFile = new File(path + "/" +
fileNameUpload + extension);
        item.write(uploadedFile);
        List<String> busquedaRuta = new
ArrayList<>();

        String ruta = "";
        int vtipo = Integer.parseInt(tipo);
        if (tipo.equals("3")) {
            busquedaRuta.add("mtllib");
            ruta = producto.getFoto() + ".mtl";
        } else {
            ruta = "..\\..\\Materials\\" +
producto.getMaterialArchivo();
            busquedaRuta.add("map_Kd");
            busquedaRuta.add("map_Ka");
        }
        List<String> listaRuta = listaLineasRuta(("" +
uploadedFile), busquedaRuta, ruta, vtipo);

```

```

removeLineFromFile(("" + uploadedFile),
listaRuta, busquedaRuta, ruta, vtipo);
break;
}
records.put("producto", producto.getId());
records.put("msg", "OK");
}
} else {
records.put("msg", "V2");
}
out.print(gson.toJson(records));
return;
} else {
switch (item.getFieldName()) {
case "accion":
accion =
obtenerDatoDeFormulario(item.getInputStream());
break;
case "tipo":
tipo =
obtenerDatoDeFormulario(item.getInputStream());
break;
}
}
}
} catch (Exception e) {
records.put("msg", "V1");
out.print(gson.toJson(records));
}
}

```

Este código solo copia el archivo puesto en el formulario y pega distribuyéndolo los archivos en la rutas dispuestos . solo el nombre del material se modifica en la Base de datos, como se muestra en modificarMaterialProducto, también se cambia el nombre en general de files3D.

✓ Parte de Logica.

```
public boolean modificarMaterialProducto(Producto beanProducto) throws Exception
    DAOProducto dAOProducto = new DAOProducto();
    try {
        dAOProducto.Conexion();
        return dAOProducto.updateTextura(beanProducto);
    } catch (Exception e) {
        throw e;
    } finally {
        dAOProducto.close();
    }
}
```

✓ Parte de Modelo – DAO

```
public boolean updateTextura(Producto objeto) throws Exception {
    try {
        String dml = "update tb_producto set";
        String textura =objeto.getMaterialArchivo()==null?"NULL":"" + objeto.getMaterialArchivo() + "";
        dml = dml + " tb_producto_maf=" + textura + "";
        dml = dml + " where";
        dml = dml + " tb_producto_id = " + objeto.getId() + "";
        dml = dml + ";";

        return this.EjecutaSQL(dml);
    } catch (Exception e) {
        throw e;
    }
}
```

Codigo Para el Correo ,lo que hace es cuando el usuario se olvide la contrasera , la aplicación se enviara a su bandeja de su correo eletronico registrado, indicando que es único como usuario
package com.commons;

```
import java.util.Properties;
import javax.mail.BodyPart;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;
```

```
public class Correo {
    public boolean SendMail(String Subject, String Mensage, String To) {
        final String Username = "kinectclothes@outlook.es";
        final String Password = "<Password>";
```



```

Properties props = new Properties();
props.put("mail.smtp.auth", "true");
props.put("mail.smtp.starttls.enable", "true");
props.put("mail.smtp.host", "smtp-mail.outlook.com");
props.put("mail.smtp.user", "kinectclothes@outlook.es");
props.put("mail.smtp.port", "587");

Session session = Session.getInstance(props,
    new javax.mail.Authenticator() {
        @Override
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(Username, Password);
        }
    });

try {

    Message message = new MimeMessage(session);
    message.setFrom(new InternetAddress(Username));
    message.setRecipients(Message.RecipientType.TO,
        InternetAddress.parse(To));
    message.setSubject(Subject);
    message.setText(Mensaje);

    Transport.send(message);
    return true;

} catch (MessagingException e) {
    System.out.println(e);
    return false;
}
}
}
}

```

Codigo para la web service en caso que seleccione como persona jurídica se activara el ruc y este mediante un webservice se vera el ruc correspondiente desde la misma sunat, dando la validez de la sunat.

/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

```
package com.common;
```

```
import com.Beans.wsContribuyente;
import com.webservice.sunat.*;
```

```
import java.io.StringReader;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.Hashtable;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.CharacterData;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.InputSource;
```

```
/**
```

```
*
```

```
* @author CIXTIC03
```

```
*/
```

```
public class RucServiceImpl implements RucService {
```

```
    @Override
```

```
    public List<Object> obtenerContribuyente(String numruc) {
```

```
        List<Object> listaRespuesta = new ArrayList<Object>();
```

```
        Servicegetruc_Impl oConsultaRUC = new Servicegetruc_Impl();
```

```
        String response;
```

```
        wsContribuyente oContribuyente = new wsContribuyente();
```

```
        try {
```

```
            response =
```

```
oConsultaRUC.getServicegetrucPort().consultaRUC(numruc,
"softimc@hotmail.com", "<código_licencia>", "");
```

```
String xml = response;
DocumentBuilder builder =
DocumentBuilderFactory.newInstance().newDocumentBuilder();
InputStream src = new InputStream();
src.setCharacterStream(new StringReader(xml));

Document doc = builder.parse(src);

NodeList nodes = doc.getElementsByTagName("root");
String status_msg = "";
String status_date = "";
String ruc = "";
String alias = "";
String condicion = "";
String ubigeo = "";
String direccion = "";
String estado = "";
// iterate the employees
for (int i = 0; i < nodes.getLength(); i++) {
    Element element = (Element) nodes.item(i);
    if (element.hasChildNodes()) {
        NodeList n1_ruc = element.getElementsByTagName("n1_ruc");
        if (n1_ruc != null && n1_ruc.getLength() > 0) {
            Element line = (Element) n1_ruc.item(0);
            ruc = getCharacterDataFromElement(line);
            //System.out.println("n1_ruc: " +
getCharacterDataFromElement(line));

            NodeList n1_alias =
element.getElementsByTagName("n1_alias");
            line = (Element) n1_alias.item(0);
            alias = getCharacterDataFromElement(line);
            //System.out.println("n1_alias: " +
getCharacterDataFromElement(line));

            NodeList n1_estado =
element.getElementsByTagName("n1_estado");
            line = (Element) n1_estado.item(0);
            estado = getCharacterDataFromElement(line);
            // System.out.println("n1_estado: " +
getCharacterDataFromElement(line));

            NodeList n1_condicion =
element.getElementsByTagName("n1_condicion");
```

```
        line = (Element) n1_condicion.item(0);
        condicion = getCharacterDataFromElement(line);
        // System.out.println("n1_condicion: " +
getCharacterDataFromElement(line));

        NodeList n1_ubigeo =
element.getElementsByTagName("n1_ubigeo");
        line = (Element) n1_ubigeo.item(0);
        ubigeo = getCharacterDataFromElement(line);
        // System.out.println("n1_ubigeo: " +
getCharacterDataFromElement(line));

        NodeList n1_direccion =
element.getElementsByTagName("n1_direccion");
        line = (Element) n1_direccion.item(0);
        direccion = getCharacterDataFromElement(line);
        // System.out.println("n1_direccion: " +
getCharacterDataFromElement(line));

        oContribuyente.setRuc(ruc);
        oContribuyente.setAlias(alias);
        oContribuyente.setDireccion(direccion);
        oContribuyente.setCondicion(condicion);
        oContribuyente.setFechaRegistro(status_date);
        oContribuyente.setUbigeo(ubigeo);
        oContribuyente.setEstado(estado);
        status_msg = "OK";

    } else {
        oContribuyente = null;
        status_msg = "No existe contribuyente con ese ruc";
    }
    } else {
        oContribuyente = null;
        status_msg = "No existe contribuyente con ese ruc";
    }
}

listaRespuesta.add(status_msg);
listaRespuesta.add(oContribuyente);

} catch (Exception ex) {
```

```

Logger.getLogger(RucServiceImpl.class.getName()).log(Level.SEVERE,
null, ex);
    }

    return listaRespuesta;
}

public static String getCharacterDataFromElement(Element e) {
    Node child = e.getFirstChild();
    if (child instanceof CharacterData) {
        CharacterData cd = (CharacterData) child;
        return cd.getData();
    }
    return "?";
}
}
}

```

Codigo Para Compras

```
package com.Servlets;
```

```

import com.Beans.Captura;
import com.Beans.Interaccion;
import com.Beans.TipoComprobante;
import com.Beans.UsuarioCliente;
import com.Logics.LogicInteraccion;
import com.Logics.LogicTipoComprobante;
import com.google.gson.Gson;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *

```

```

* @author ERICK
*/
@WebServlet(name = "ServletCompras", urlPatterns =
{"/ServletCompras"})
public class ServletCompras extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException, Exception {
        response.setContentType("text/html;charset=UTF-8");
        response.setCharacterEncoding("UTF-8");
        PrintWriter out = response.getWriter();
        try {
            HttpSession session = request.getSession();
            UsuarioCliente beanUsuario = (UsuarioCliente)
session.getAttribute("sesUsuarioCliente");
            if (beanUsuario != null) {

                String accion = request.getParameter("accion");
                if (accion != null) {
                    switch (accion) {
                        case "ver":
                            verAjax(request, response);
                            break;
                        case "guardar":
                            guardarAjax(request, response);
                            break;
                        case "ajaxCaptura":
                            ajaxCaptura(request, response);
                            break;
                        default:
                            break;
                    }
                } else {
                    session.setAttribute("sesReporte", "compra");
                    String valor = (String) session.getAttribute("asignar_cliente");
                    if (valor != null) {
                        if (valor.equals("SI")) {
                            asignarClienteInteraccion(request, response);
                        }
                    }
                    LogicInteraccion logicInteraccion = new LogicInteraccion();
                    LogicTipoComprobante logicTipoComprobante = new
LogicTipoComprobante();

```



```

        List<Interaccion> listaInteraccionRC =
logicInteraccion.listaInteraccionReservacionCompra(beanUsuario.getoClie
nte().getId());
        List<TipoComprobante> lTipoComprobante =
logicTipoComprobante.listaTipoComprobante(new TipoComprobante(),
null, null, null, null);
        request.setAttribute("listaInteraccionRC", listaInteraccionRC);
        request.setAttribute("listaTipoComprobante",
lTipoComprobante);
        request.getRequestDispatcher("compras.jsp").forward(request,
response);
    }

    } else {
        request.setAttribute("error", "LA SESIÓN A EXPIRADO, INICIE
SESIÓN NUEVAMENTE");
        request.setAttribute("tipo", "usuario");
        request.getRequestDispatcher("login.jsp").forward(request,
response);
    }

    } catch (Exception e) {
        out.print("error de BD");
        request.setAttribute("tipo", e.getCause());
        request.setAttribute("error", e.getMessage());
        request.getRequestDispatcher("paginaError.jsp").forward(request,
response);
    } finally {
        out.close();
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override

```

```
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (Exception ex) {
```

```
Logger.getLogger(ServletCompras.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
```

```
/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (Exception ex) {
```

```
Logger.getLogger(ServletCompras.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
```

```
/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
```

```

private int calcularTiempo(Interaccion interaccion) throws
NumberFormatException {
    String tf = interaccion.getTiempoFinal();
    String tf_ho = tf.substring(11, 13);
    String tf_mm = tf.substring(14, 16);
    String tf_ss = tf.substring(17, 19);
    int tft = Integer.parseInt(tf_ho) * 3600 + Integer.parseInt(tf_mm) * 60
+ Integer.parseInt(tf_ss);
    String ti = interaccion.getTiempoInicial();
    String ti_ho = ti.substring(11, 13);
    String ti_mm = ti.substring(14, 16);
    String ti_ss = ti.substring(17, 19);
    int tit = Integer.parseInt(ti_ho) * 3600 + Integer.parseInt(ti_mm) * 60
+ Integer.parseInt(ti_ss);
    int titt = tft - tit;
    return titt;
}

public String formatoTiempo(int segundos) {
    String tiempo = "";
    String strHora = "";
    String strMinutos = "";
    String strSegundos = "";
    int h = (segundos / 3600);
    int m = (segundos - (h * 3600)) / 60;
    int s = segundos % 60;
    if (String.valueOf(h).length() == 1) {
        strHora = "0" + h;
    } else {
        strHora = String.valueOf(h);
    }
    if (String.valueOf(m).length() == 1) {
        strMinutos = "0" + m;
    } else {
        strMinutos = String.valueOf(m);
    }
    if (String.valueOf(s).length() == 1) {
        strSegundos = "0" + s;
    } else {
        strSegundos = String.valueOf(s);
    }
    tiempo = strHora + ":" + strMinutos + ":" + strSegundos;
    return tiempo;
}

```

```

private void ajaxCaptura(HttpServletRequest request,
HttpServletRequest response) throws Exception {
    try {
        Map registro = new HashMap<String, Object>();
        Gson gson = new Gson();
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession();

        String idInteraccion = request.getParameter("interaccion");
        LogicInteraccion logicInteraccion = new LogicInteraccion();
        Interaccion beanInteraccion =
logicInteraccion.getInteraccion(idInteraccion != null &&
!idInteraccion.equals("") ? Integer.parseInt(idInteraccion) : 0);
        List<Captura> lCaptura = beanInteraccion.getlCaptura();
        int titt = calcularTiempo(beanInteraccion);
        String detalleHTML;
        String accordionHTML = "";
        String formato = formatoTiempo(titt);
        detalleHTML = "
                <div class=\"row\">
                + \"
                <div class=\"col-md-12\">
                + \"
                <span class=\"label label-primary col-md-5\">\n\"
                + \"
                Tiempo Interaccion \n\"
                + \"
                </span>
                \n\"
                + \"
                <span class=\"label label-default col-md-7\">\"
                + formato
                + \"
                </span> \n\"
                + \"
                </div>
                \n\"
                + \"
                </div>
                \n\";
        detalleHTML += "
                <div class=\"row\">
                + \"
                <div class=\"col-md-12\">
                + \"
                <span class=\"label label-primary col-md-5\">\n\"
                + \"
                Tiempo Inicial\n\"
                + \"
                </span>
                \n\"
                + \"
                <span class=\"label label-default col-md-7\">\"
                + beanInteraccion.getTiempoInicial()
                + \"
                </span> \n\"
                + \"
                </div>
                \n\"
                + \"
                </div>
                \n\";
        detalleHTML += "
                <div class=\"row\">
                + \"
                <div class=\"col-md-12\">
                + \"
                <span class=\"label label-primary col-md-5\">\n\"
                + \"
                Tiempo Final\n\"
                + \"
                </span>
                \n\"
    
```



```

+ "      <span class=\"label label-default col-md-7\">\"
+ beanInteraccion.getTiempoFinal()
+ "      </span> \n\"
+ "      </div>          \n\"
+ "      </div>          \n\";
for (int i = 0; i < lCaptura.size(); i++) {
    String rutaF = \"img/Photos/\" + lCaptura.get(i).getFoto();
    accordionHTML += \"<div class=\"panel\">\n\"
        + \"      <div class=\"panel-heading bg-red-thunderbird bg-
font-red-thunderbird\">\n\"
        + \"      <h4 class=\"panel-title\">\n\"
        + \"      <a class=\"accordion-toggle accordion-toggle-
styled collapsed\" data-toggle=\"collapse\" data-parent=\"#accordion1\"
href=\"#collapse_1_\" + (i + 1) + \"\"> <span class=\"hidden-
480\">Fecha:</span>\" + lCaptura.get(i).getFecha() + \"</a>\n\"
        + \"      </h4>\n\"
        + \"      </div>\n\"
        + \"      <div id=\"collapse_1_\" + (i + 1) + \"\" class=\"panel-
collapse collapse\">\n\"
        + \"      <div class=\"panel-body\">\n\"
        + \"      <div class=\"row\">\"
        + \"      <div class=\"col-lg-3 col-md-5 col-sm-6\">\"
        + \"      <a href=\"\" + rutaF + \"\" class=\"btn btn-default
fancybox-button\" >\n\" +
        \"      <img class=\"img-responsive\"
src=\"\" + rutaF + \"\" alt=\"\"> </a>\"
        /* + \"      <a href=\"\" + rutaF + \"\" class=\"btn btn-
default fancybox-button\" target=\"_blank\">\n\"
        + \"      <img src=\"\" + rutaF + \"\"
style=\"width: 100%; height: 200px;\"/></a>\"*/
        + \"      </div>\"
        + \"      </div>\"
        + \"      </div>\n\"
        + \"      </div>\n\"
        + \"      </div>\";
    }
    registro.put(\"accordion\", accordionHTML);
    registro.put(\"detalleinteraccion\", detalleHTML);
    // registro.put(\"numerocapturas\", lCaptura.size());
    out.print(gson.toJson(registro));
} catch (Exception e) {

}
}
}

```

```

private void verAjax(HttpServletRequest request, HttpServletResponse
response) throws Exception {
    try {

        } catch (Exception e) {
            throw e;
        }
    }

private void asignarClienteInteraccion(HttpServletRequest request,
HttpServletResponse response) throws Exception {

    HttpSession session = request.getSession();
    try {
        LogicInteraccion logicInteraccion = new LogicInteraccion();
        String token = (String) session.getAttribute("token_compra");
        UsuarioCliente beanUsuario = (UsuarioCliente)
session.getAttribute("sesUsuarioCliente");
        if (token != null) {
            Interaccion oInteraccion = logicInteraccion.getInteraccion(token);

            if (logicInteraccion.asignarCliente(oInteraccion.getId(),
beanUsuario.getoCliente().getId())) {
                session.setAttribute("token_compra", null);
                session.setAttribute("asignar_cliente", null);
            }
        }
    } catch (Exception e) {

    }

private void guardarAjax(HttpServletRequest request,
HttpServletResponse response) throws Exception {
    Gson gson = new Gson();
    PrintWriter out = response.getWriter();
    HttpSession session = request.getSession();
    Map records = new HashMap<>();
    try {
        out.print(gson.toJson(records));

    } catch (Exception e) {

```



```

        out.print(gson.toJson("ERROR"));
    }
}
}

```

Código de Java, Servlet de Dirección que, un usuario de la aplicación final puede tener más de una dirección para que seleccione para la entrega del producto.

```

package com.Servlets;

```

```

import com.Beans.Captura;
import com.Beans.Interaccion;
import com.Beans.TipoComprobante;
import com.Beans.UsuarioCliente;
import com.Logics.LogicInteraccion;
import com.Logics.LogicTipoComprobante;
import com.google.gson.Gson;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

```

```

/**
 *
 * @author ERICK
 */
@WebServlet(name = "ServletCompras", urlPatterns =
{"/ServletCompras"})
public class ServletCompras extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException, Exception {
        response.setContentType("text/html;charset=UTF-8");
        response.setCharacterEncoding("UTF-8");

```

```

PrintWriter out = response.getWriter();
try {
    HttpSession session = request.getSession();
    UsuarioCliente beanUsuario = (UsuarioCliente)
session.getAttribute("sesUsuarioCliente");
    if (beanUsuario != null) {

        String accion = request.getParameter("accion");
        if (accion != null) {
            switch (accion) {
                case "ver":
                    verAjax(request, response);
                    break;
                case "guardar":
                    guardarAjax(request, response);
                    break;
                case "ajaxCaptura":
                    ajaxCaptura(request, response);
                    break;
                default:
                    break;
            }
        }
        else {
            session.setAttribute("sesReporte", "compra");
            String valor = (String)
session.getAttribute("asignar_cliente");
            if (valor != null) {
                if (valor.equals("SI")) {
                    asignarClienteInteraccion(request, response);
                }
            }
            LogicInteraccion logicInteraccion = new LogicInteraccion();
            LogicTipoComprobante logicTipoComprobante = new
LogicTipoComprobante();
            List<Interaccion> listaInteraccionRC =
logicInteraccion.listaInteraccionReservacionCompra(beanUsuario.ge
toCliente().getId());
            List<TipoComprobante> lTipoComprobante =
logicTipoComprobante.listaTipoComprobante(new
TipoComprobante(), null, null, null, null);
            request.setAttribute("listaInteraccionRC",
listaInteraccionRC);
            request.setAttribute("listaTipoComprobante",
lTipoComprobante);
        }
    }

```

```

request.getRequestDispatcher("compras.jsp").forward(request,
response);
    }

    } else {
        request.setAttribute("error", "LA SESIÓN A EXPIRADO, INICIE
SESIÓN NUEVAMENTE");
        request.setAttribute("tipo", "usuario");
        request.getRequestDispatcher("login.jsp").forward(request,
response);
    }

    } catch (Exception e) {
        out.print("error de BD");
        request.setAttribute("tipo", e.getCause());
        request.setAttribute("error", e.getMessage());

request.getRequestDispatcher("paginaError.jsp").forward(request,
response);
    } finally {
        out.close();
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (Exception ex) {

```

```

Logger.getLogger(ServletCompras.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

```

```

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request,
HttpServletRequest response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (Exception ex) {

```

```

Logger.getLogger(ServletCompras.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

```

```

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>

```

```

private int calcularTiempo(Interaccion interaccion) throws
NumberFormatException {
    String tf = interaccion.getTiempoFinal();
    String tf_ho = tf.substring(11, 13);
    String tf_mm = tf.substring(14, 16);
    String tf_ss = tf.substring(17, 19);

```

```

        int tft = Integer.parseInt(tf_ho) * 3600 + Integer.parseInt(tf_mm)
* 60 + Integer.parseInt(tf_ss);
        String ti = interaccion.getTiempoInicial();
        String ti_ho = ti.substring(11, 13);
        String ti_mm = ti.substring(14, 16);
        String ti_ss = ti.substring(17, 19);
        int tit = Integer.parseInt(ti_ho) * 3600 + Integer.parseInt(ti_mm)
* 60 + Integer.parseInt(ti_ss);
        int titt = tft - tit;
        return titt;
    }

```

```

public String formatoTiempo(int segundos) {
    String tiempo = "";
    String strHora = "";
    String strMinutos = "";
    String strSegundos = "";
    int h = (segundos / 3600);
    int m = (segundos - (h * 3600)) / 60;
    int s = segundos % 60;
    if (String.valueOf(h).length() == 1) {
        strHora = "0" + h;
    } else {
        strHora = String.valueOf(h);
    }
    if (String.valueOf(m).length() == 1) {
        strMinutos = "0" + m;
    } else {
        strMinutos = String.valueOf(m);
    }
    if (String.valueOf(s).length() == 1) {
        strSegundos = "0" + s;
    } else {
        strSegundos = String.valueOf(s);
    }
    tiempo = strHora + ":" + strMinutos + ":" + strSegundos;
    return tiempo;
}

```

```

private void ajaxCaptura(HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    try {
        Map registro = new HashMap<String, Object>();
        Gson gson = new Gson();
    }
}

```

```

PrintWriter out = response.getWriter();
HttpSession session = request.getSession();

String idInteraccion = request.getParameter("interaccion");
LogicInteraccion logicInteraccion = new LogicInteraccion();
Interaccion beanInteraccion =
logicInteraccion.getInteraccion(idInteraccion != null &&
!idInteraccion.equals("") ? Integer.parseInt(idInteraccion) : 0);
List<Captura> lCaptura = beanInteraccion.getlCaptura();
int titt = calcularTiempo(beanInteraccion);
String detalleHTML;
String accordionHTML = "";
String formato = formatoTiempo(titt);
detalleHTML = " <div class=\"row\"> \n" \n"
+ " <div class=\"col-md-12\"> \n" \n"
+ " <span class=\"label label-primary col-md-
5\">\n"
+ " Tiempo Interaccion \n"
+ " </span> \n" \n"
+ " <span class=\"label label-default col-md-7\">"
+ formato
+ " </span> \n" \n"
+ " </div> \n" \n"
+ " </div> \n";
detalleHTML += " <div class=\"row\"> \n" \n"
+ " <div class=\"col-md-12\"> \n" \n"
+ " <span class=\"label label-primary col-md-
5\">\n"
+ " Tiempo Inicial\n"
+ " </span> \n" \n"
+ " <span class=\"label label-default col-md-7\">"
+ beanInteraccion.getTiempoInicial()
+ " </span> \n" \n"
+ " </div> \n" \n"
+ " </div> \n";
detalleHTML += " <div class=\"row\"> \n" \n"
+ " <div class=\"col-md-12\"> \n" \n"
+ " <span class=\"label label-primary col-md-
5\">\n"
+ " Tiempo Final\n"
+ " </span> \n" \n"
+ " <span class=\"label label-default col-md-7\">"
+ beanInteraccion.getTiempoFinal()
+ " </span> \n"

```




```

+ "      </div>          \n"
+ "      </div>          \n";
for (int i = 0; i < lCaptura.size(); i++) {
    String rutaF = "img/Photos/" + lCaptura.get(i).getFoto();
    accordionHTML += "<div class=\"panel\">\n"
+ "      <div class=\"panel-heading bg-red-thunderbird
bg-font-red-thunderbird\">\n"
+ "      <h4 class=\"panel-title\">\n"
+ "      <a class=\"accordion-toggle accordion-toggle-
styled collapsed\" data-toggle=\"collapse\" data-
parent=\"#accordion1\" href=\"#collapse_1_\" + (i + 1) + \"\"> <span
class=\"hidden-480\">Fecha:</span> + lCaptura.get(i).getFecha() +
\"</a>\n"
+ "      </h4>\n"
+ "      </div>\n"
+ "      <div id=\"collapse_1_\" + (i + 1) + \"\" class=\"panel-
collapse collapse\">\n"
+ "      <div class=\"panel-body\">\n"
+ "      <div class=\"row\">
+ "      <div class=\"col-lg-3 col-md-5 col-sm-6\">
+ "      <a href=\"\" + rutaF + \"\" class=\"btn btn-
default fancybox-button\" >\n" +
"      <img class=\"img-responsive\"
src=\"\" + rutaF + \"\" alt=\"\"> </a>
/* + "      <a href=\"\" + rutaF + \"\" class=\"btn btn-
default fancybox-button\" target=\"_blank\">\n"
+ "      <img src=\"\" + rutaF + \"\"
style=\"width: 100%; height: 200px;\"/></a>*/
+ "      </div>
+ "      </div>
+ "      </div>\n"
+ "      </div>\n"
+ "      </div>";
}
registro.put("accordion", accordionHTML);
registro.put("detalleinteraccion", detalleHTML);
// registro.put("numerocapturas", lCaptura.size());
out.print(gson.toJson(registro));
} catch (Exception e) {
}
}
}

```



```
private void verAjax(HttpServletRequest request,
HttpServletRequest response) throws Exception {
    try {

    } catch (Exception e) {
        throw e;
    }
}
```

```
private void asignarClienteInteraccion(HttpServletRequest request,
HttpServletRequest response) throws Exception {
```

```
    HttpSession session = request.getSession();
    try {
        LogicInteraccion logicInteraccion = new LogicInteraccion();
        String token = (String) session.getAttribute("token_compra");
        UsuarioCliente beanUsuario = (UsuarioCliente)
session.getAttribute("sesUsuarioCliente");
        if (token != null) {
            Interaccion oInteraccion =
logicInteraccion.getInteraccion(token);

            if (logicInteraccion.asignarCliente(oInteraccion.getId(),
beanUsuario.getoCliente().getId())) {
                session.setAttribute("token_compra", null);
                session.setAttribute("asignar_cliente", null);
            }
        }
    } catch (Exception e) {

    }
}
```

```
private void guardarAjax(HttpServletRequest request,
HttpServletRequest response) throws Exception {
```

```
    Gson gson = new Gson();
    PrintWriter out = response.getWriter();
    HttpSession session = request.getSession();
    Map records = new HashMap<>();
    try {
        out.print(gson.toJson(records));

    } catch (Exception e) {
```

```

        out.print(gson.toJson("ERROR"));
    }
}
}

```

Código de Datos personales del Usuario sea perfil o datos como comprado, sea Ubigeo, tipo cliente, etc.

```
package com.Servlets;
```

```

import com.Beans.Cliente;
import com.Beans.Configuracion;
import com.Beans.Genero;
import com.Beans.TipoCliente;
import com.Beans.Ubigeo;
import com.Beans.UsuarioCliente;
import com.Beans.wsContribuyente;
import com.Logics.LogicCliente;
import com.Logics.LogicConfiguracion;
import com.Logics.LogicUbigeo;
import com.Logics.LogicUsuarioCliente;
import com.commons.Cadenas;
import com.commons.RucServiceImpl;
import com.google.gson.Gson;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

```

```

@WebServlet(name = "ServletDatosPersonales", urlPatterns =
{"/ServletDatosPersonales"})
public class ServletDatosPersonales extends HttpServlet {

```

```

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)

```

```
throws ServletException, IOException, Exception {
response.setContentType("text/html;charset=UTF-8");
response.setCharacterEncoding("UTF-8");
PrintWriter out = response.getWriter();
try {
    HttpSession session = request.getSession();
    UsuarioCliente beanUsuario = (UsuarioCliente)
session.getAttribute("sesUsuarioCliente");

    if (beanUsuario != null) {
        String accion = request.getParameter("accion");
        if (accion != null) {
            switch (accion) {
                case "ver":
                    verAjax(request, response);
                    break;
                case "ver_sunat":
                    verSUNAT(request, response);
                    break;
                case "guardar":
                    guardarClienteAjax(request, response);
                    break;
                case "guardar_perfil":
                    guardarPerfilUsuario(request, response);
                    break;
                case "ver_provincias":
                    ajaxProvincias(request, response);
                    break;
                case "ver_distritos":
                    ajaxDistritos(request, response);
                    break;
                default:
                    break;
            }
        } else {
            session.setAttribute("sesReporte", "datospersonales");
            LogicUbigeo logicUbigeo = new LogicUbigeo();

            logicUbigeo.listaTodosDepartamentos();
            List<Ubigeo> lista =
logicUbigeo.listaTodosDepartamentos();
            request.setAttribute("listaDepartamento", lista);
        }
    }
}
```

```

request.getRequestDispatcher("datospersonales.jsp").forward(reque
st, response);
    }
    } else {
        request.setAttribute("error", "LA SESIÓN A EXPIRADO, INICIE
SESIÓN NUEVAMENTE");
        request.setAttribute("tipo", "usuario");
        request.getRequestDispatcher("login.jsp").forward(request,
response);
    }

    } catch (Exception e) {
        out.print("error de BD");
        request.setAttribute("tipo", e.getCause());
        request.setAttribute("error", e.getMessage());

request.getRequestDispatcher("paginaError.jsp").forward(request,
response);
    } finally {
        out.close();
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet
methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (Exception ex) {

Logger.getLogger(ServletDatosPersonales.class.getName()).log(Level
.SEVERE, null, ex);

```

```

    }
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (Exception ex) {

        Logger.getLogger(ServletDatosPersonales.class.getName()).log(Level
            .SEVERE, null, ex);
    }
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>

private void ajaxProvincias(HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    try {
        LogicUbigeo li = new LogicUbigeo();
        String departamento = request.getParameter("departamento")
            == null ? "0" : request.getParameter("departamento").equals("") ? "0"
            : request.getParameter("departamento");
        List<Ubigeo> lista =
            li.listaProvinciasxDepartamento(departamento);
    }
}

```




```

String html = "<div class=\"input-group\">\n"
    + "        <span class=\"input-group-
addon\">\n"
    + "            <i class=\"fa fa-search\"></i>\n"
    + "        </span>\n"
    + "        <select class=\"form-control
select2\" data-placeholder=\"Select...\" id=\"provincia\"
name=\"provincia\">\n"
    + "            <option value=\"\">--
Seleccionar--</option>;
    for (int i = 0; i < lista.size(); i++) {
        html += "<option value=\"" + lista.get(i).getProvincia() + "\"
>" + lista.get(i).getProvincia() + "</option>";
    }

    html += " </select>\n"
        + "</div>";
Gson gson = new Gson();
PrintWriter out = response.getWriter();

    out.print(gson.toJson(html));
} catch (Exception e) {
    throw e;
} //To change body of generated methods, choose Tools |
Templates.
}

private void ajaxDistritos(HttpServletRequest request,
HttpServletRequest response) throws Exception {
    try {
        LogicUbigeo li = new LogicUbigeo();
        String departamento = request.getParameter("departamento");
        String provincia = request.getParameter("provincia");
        Ubigeo ubigeo = new Ubigeo();
        ubigeo.setDepartamento(departamento);
        ubigeo.setProvincia(provincia);
        List<Ubigeo> lista = li.listaUbigeo(ubigeo, null, null);

        Map registro = new HashMap<String, Object>();

        String html = "<div class=\"input-group\">\n"
            + "        <span class=\"input-group-
addon\">\n"
            + "            <i class=\"fa fa-search\"></i>\n"

```

```

        + "                </span>\n"
        + "                <select class=\"form-control
select2\" data-placeholder=\"Select...\" id=\"distrito\"
name=\"distrito\">\n"
        + "                <option value=\"\">--
Seleccionar--</option>>";
        for (int i = 0; i < lista.size(); i++) {
            html += "<option value=\"" + lista.get(i).getCodigo() + "\" >" +
lista.get(i).getDistrito() + "</option>";
        }

        html += " </select>\n"
            + "</div>";
        Gson gson = new Gson();
        PrintWriter out = response.getWriter();

        out.print(gson.toJson(html));
    } catch (Exception e) {
        throw e;
    } //To change body of generated methods, choose Tools |
Templates.
    }

    private void verAjax(HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        try {
            HttpSession session = request.getSession();
            Gson gson = new Gson();
            PrintWriter out = response.getWriter();
            UsuarioCliente usuarioCliente = (UsuarioCliente)
session.getAttribute("sesUsuarioCliente");
            LogicUsuarioCliente logicUsuarioCliente = new
LogicUsuarioCliente();
            UsuarioCliente objeto =
logicUsuarioCliente.verUsuario(usuarioCliente.getId());
            out.print(gson.toJson(objeto));
        } catch (Exception e) {
            throw e;
        }
    }

    private boolean verificarCambio(Cliente beanCliente, Cliente
objeto) {
        boolean cambio = true;

```

```

        if
        (beanCliente.getApellidoPaterno().equals(objeto.getApellidoPaterno
        ()
            &&
        beanCliente.getApellidoMaterno().equals(objeto.getApellidoMaterno
        ()
            && beanCliente.getNombre().equals(objeto.getNombre())
            && beanCliente.getEmail().equals(objeto.getEmail())
            && beanCliente.getTelefono().equals(objeto.getTelefono())
            && beanCliente.getEstado().equals(objeto.getEstado())
            && beanCliente.getDni().equals(objeto.getDni())
            && beanCliente.getDireccion().equals(objeto.getDireccion())
            &&
        beanCliente.getFechaNacimiento().equals(objeto.getFechaNacimient
        o())
            && beanCliente.getoGenero().getId().intValue() ==
        objeto.getoGenero().getId()
            &&
        beanCliente.getoUbigeo().getCodigo().equals(objeto.getoUbigeo().get
        Codigo())
            && beanCliente.getoTipoCliente().getId() ==
        objeto.getoTipoCliente().getId().intValue()) {
            if (beanCliente.getoTipoCliente().getId() == 2) {
                if (beanCliente.getRuc().equals(objeto.getRuc())
                    && beanCliente.getRazon().equals(objeto.getRazon())
                    && beanCliente.getAlias().equals(objeto.getAlias())) {
                    cambio = false;
                }
            } else {
                cambio = false;
            }
        }
        return cambio;
    }

```

```

    private void guardarPerfilUsuario(HttpServletRequest request,
    HttpServletResponse response) throws Exception {
        HttpSession session = request.getSession();
        try {
            UsuarioCliente beanUsuario = (UsuarioCliente)
            session.getAttribute("sesUsuarioCliente");
            UsuarioCliente usuario = new UsuarioCliente();

            String nick = request.getParameter("usuario");

```

```

String clave = request.getParameter("clave");
String repetirclave = request.getParameter("repetirclave");
PrintWriter out = response.getWriter();
Gson gson = new Gson();
boolean validacion = true;
Map records = new HashMap<String, Object>();
List<String> validaciones = new ArrayList<String>();
LogicUsuarioCliente logicUsuarioCliente = new
LogicUsuarioCliente();

if (nick == null || nick.trim().equals("")) {
    validacion = false;
    validaciones.add("V1");
}
if (clave == null || clave.trim().equals("")) {
    validacion = false;
    validaciones.add("V4");
}
if (repetirclave == null || repetirclave.trim().equals("")) {
    validacion = false;
    validaciones.add("V5");
}
if (validacion) {
    if (!repetirclave.equals(clave)) {
        validacion = false;
        validaciones.add("V6");
    }
}
if (validaciones.isEmpty()) {
    UsuarioCliente usuarioNick = null;
    if
(!beanUsuario.getNick().toUpperCase().equals(nick.to
UpperCase())) {
        usuarioNick = logicUsuarioCliente.verPorNick(nick);
    }
    if (usuarioNick == null) {
        usuario.setNick(nick);
        usuario.setPassword(clave);
        usuario.setId(beanUsuario.getId());
        if (logicUsuarioCliente.modificarPerfilUsuario(usuario)) {
            records.put("msj", "OK");
            beanUsuario.setNick(nick);
            beanUsuario.setPassword(clave);
            session.setAttribute("sesUsuarioCliente",beanUsuario);
        }
    }
}

```

```

        } else {
            records.put("msj", "ERROR");
        }
    }else{
        validaciones.add("V7");
    }
}
records.put("validacion", validaciones);
out.print(gson.toJson(records));
} catch (Exception e) {
}
}

```

```

private void guardarClienteAjax(HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    try {
        HttpSession session = request.getSession();
        Cliente beanCliente = new Cliente();
        UsuarioCliente beanUsuario = (UsuarioCliente)
session.getAttribute("sesUsuarioCliente");
        beanCliente.setId(beanUsuario.getCliente().getId());
        beanCliente.setNombre(request.getParameter("nombre"));

        beanCliente.setApellidoPaterno(request.getParameter("apellidopaterno"));

        beanCliente.setApellidoMaterno(request.getParameter("apellidomaterno"));
        beanCliente.setEmail(request.getParameter("email"));
        beanCliente.setTelefono(request.getParameter("telefono"));
        beanCliente.setDni(request.getParameter("dni"));
        beanCliente.setDireccion(request.getParameter("direccion"));

        beanCliente.setFechaNacimiento(request.getParameter("fechanacimiento"));
        beanCliente.setTipoCliente(new
TipoCliente(Integer.parseInt(request.getParameter("tipocliente"))));
        ;
        beanCliente.setGenero(new
Genero(request.getParameter("genero") == null ? null :
request.getParameter("genero").equals("") ||
request.getParameter("genero").equals("0") ? null :
Integer.parseInt(request.getParameter("genero"))));
    }
}

```

```

        beanCliente.setoUbigeo(new
Ubigeo(request.getParameter("ubigeo") == null ? null :
request.getParameter("ubigeo").equals("") ? null :
(request.getParameter("ubigeo"))));
        List<String> validaciones = new ArrayList<String>();
        if (beanCliente.getoTipoCliente().getId() == 2) {
            beanCliente.setRazon(request.getParameter("razon"));
            beanCliente.setRuc(request.getParameter("ruc"));
            RucServiceImpl rucServiceImpl = new RucServiceImpl();
            List<Object> lista =
rucServiceImpl.obtenerContribuyente(beanCliente.getRuc());
            wsContribuyente wsContribuyente = new wsContribuyente();
            if (!lista.isEmpty()) {
                if (lista.get(0).equals("OK")) {
                    wsContribuyente = (wsContribuyente) lista.get(1);
                    beanCliente.setAlias(wsContribuyente.getAlias());
                    beanCliente.setoUbigeo(new
Ubigeo(wsContribuyente.getUbigeo()));
                }
            }
            beanCliente.setDireccion(wsContribuyente.getDireccion());
            } else {
                validaciones.add("V14");
            }
            } else {
                validaciones.add("V15");
            }
            } else {
                beanCliente.setRazon("");
                beanCliente.setRuc("");
                beanCliente.setAlias("");
            }
            beanCliente.setEstado("H");
            LogicCliente logicCliente = new LogicCliente();

            PrintWriter out = response.getWriter();
            Gson gson = new Gson();
            boolean validacion = true;
            Map records = new HashMap<String, Object>();
            if (beanCliente.getEmail() == null ||
beanCliente.getEmail().equals("")) {
                validacion = false;
                validaciones.add("V16");
            }
    
```



```
        if (beanCliente.getFechaNacimiento() == null ||
beanCliente.getFechaNacimiento().equals("")) {
            validacion = false;
            validaciones.add("V17");
        }

        if (beanCliente.getTipoCliente().getId() == null ||
beanCliente.getTipoCliente().getId() == 0) {
            validacion = false;
            validaciones.add("V7");
        }

        if (beanCliente.getNombre() == null ||
"".equals(beanCliente.getNombre())) {
            validacion = false;
            validaciones.add("V1");
        }
        if (beanCliente.getApellidoPaterno() == null ||
"".equals(beanCliente.getApellidoPaterno())) {
            validacion = false;
            validaciones.add("V4");
        }
        if (beanCliente.getApellidoMaterno() == null ||
"".equals(beanCliente.getApellidoMaterno())) {
            validacion = false;
            validaciones.add("V5");
        }
        if (beanCliente.getDni() == null ||
"".equals(beanCliente.getDni())) {
            validacion = false;
            validaciones.add("V6");
        }
        if (beanCliente.getTipoCliente().getId() == 2) {
            if (beanCliente.getRazon() == null ||
"".equals(beanCliente.getRazon())) {
                validacion = false;
                validaciones.add("V8");
            }
            if (beanCliente.getRuc() == null ||
"".equals(beanCliente.getRuc())) {
                validacion = false;
                validaciones.add("V9");
            }
        }
    }
}
```

```

if (beanCliente.getoGenero() == null) {
    validacion = false;
    validaciones.add("V12");
} else if (beanCliente.getoGenero().getId() == null) {
    validacion = false;
    validaciones.add("V12");
}
if (beanCliente.getoUbigeo() == null) {
    validacion = false;
    validaciones.add("V13");
} else if (beanCliente.getoUbigeo().getCodigo() == null) {
    validacion = false;
    validaciones.add("V13");
}
}

if (validacion) {
    Cliente objeto = logicCliente.verCliente(beanCliente.getId());
    boolean cambio = verificarCambio(beanCliente, objeto);
    if (cambio) {
        Cliente clienteDNI = null;
        Cliente clienteConsultaDNI = new Cliente();
        clienteConsultaDNI.setDni(beanCliente.getDni());
        if (!objeto.getDni().equals(beanCliente.getDni())) {
            clienteDNI = logicCliente.verCliente(clienteConsultaDNI);
        }

        if (clienteDNI != null) {
            validaciones.add("V10");
        } else if (clienteDNI == null) {
            Cliente clienteRuc = null;
            Cliente clienteConsultaRuc = new Cliente();
            clienteConsultaRuc.setRuc(beanCliente.getRuc());
            if (beanCliente.getoTipoCliente().getId() == 2) {
                if (!beanCliente.getRuc().equals(objeto.getRuc())) {
                    clienteRuc =
logicCliente.verCliente(clienteConsultaRuc);
                }
            }
            if (clienteRuc != null) {
                validaciones.add("V11");
            } else if (clienteRuc == null) {

beanCliente.setNombre(Cadenas.corregirEspacios(beanCliente.getNo
mbre()));

```

```
beanCliente.setApellidoPaterno(Cadenas.corregirEspacios(beanClien  
te.getApellidoPaterno()));
```

```
beanCliente.setApellidoMaterno(Cadenas.corregirEspacios(beanClien  
te.getApellidoMaterno()));
```

```
        if (logicCliente.modificarCliente(beanCliente)) {  
            records.put("msj", "OK");  
        } else {  
            records.put("msj", "ERROR");  
        }  
    } else {  
        records.put("msj", "ERROR");  
    }  
} else {  
    records.put("msj", "ERROR");  
}  
}  
}  
    records.put("validacion", validaciones);  
    out.print(gson.toJson(records));  
} catch (Exception e) {  
  
}  
}
```

```
private void verSUNAT(HttpServletRequest request,  
HttpServletResponse response) throws Exception {  
    try {
```

```
        Gson gson = new Gson();  
        PrintWriter out = response.getWriter();  
        RucServiceImpl rucServiceImpl = new RucServiceImpl();  
        String ruc = request.getParameter("ruc");  
        Map records = new HashMap<>();  
        if (ruc != null) {  
            if (ruc.length() == 11) {  
                List<Object> lista =  
rucServiceImpl.obtenerContribuyente(ruc);  
                wsContribuyente wsContribuyente = new  
wsContribuyente();
```

```

        if (!lista.isEmpty()) {
            if (lista.get(0).equals("OK")) {
                wsContribuyente = (wsContribuyente) lista.get(1);
                if
                (!wsContribuyente.getCondicion().toUpperCase().equals("HABIDO")
                || !wsContribuyente.getEstado().toUpperCase().equals("ACTIVO")) {
                    records.put("ha", "NO");
                }
                records.put("msj", "OK");
                records.put("contribuyente", wsContribuyente);
                LogicUbigeo logicUbigeo = new LogicUbigeo();
                Ubigeo ubigeo =
                logicUbigeo.verUbigeo(wsContribuyente.getUbigeo());
                records.put("ubigeo", ubigeo);
            } else {
                records.put("msj", "E1");
            }
        } else {
            records.put("msj", "E2");
        }
    } else {
        records.put("msj", "E3");
    }
}

        out.print(gson.toJson(records));
    } catch (Exception e) {
        throw e;
    }
}

```

```

private List<String> listarValidacionRegistro(Configuracion
beanConfiguracion) {
    List<String> validaciones = new ArrayList<>();
    if (beanConfiguracion.getDireccionIP() == null
        || beanConfiguracion.getDireccionIP().equals("_._._._.")
        || beanConfiguracion.getDireccionIP().trim().equals("")) {
        validaciones.add("V1");
    }
    if (beanConfiguracion.getPuerto() == null
        || beanConfiguracion.getPuerto().equals("")) {
        validaciones.add("V4");
    }
    return validaciones;
}

```

```

}

private boolean verificarCambio(Configuracion configuracion,
Configuracion objeto) {
    boolean cambio = true;
    if
    (configuracion.getDireccionIP().equals(objeto.getDireccionIP())
    && configuracion.getPuerto().equals(objeto.getPuerto())) {
        cambio = false;
    }
    return cambio;
}

private void guardarAjax(HttpServletRequest request,
HttpServletRequest response) throws Exception {
    Gson gson = new Gson();
    PrintWriter out = response.getWriter();
    HttpSession session = request.getSession();
    Map records = new HashMap<>();
    try {
        Configuracion beanConfiguracion = new Configuracion();
        beanConfiguracion.setDireccionIP(request.getParameter("ip")
        == null ? "127.0.0.1" : request.getParameter("ip").trim());
        beanConfiguracion.setPuerto(request.getParameter("puerto")
        == null ? "8080" : request.getParameter("puerto").trim());
        LogicConfiguracion logicConfiguracion = new
        LogicConfiguracion();

        Configuracion configuracion = (Configuracion)
        session.getAttribute("sesConfiguracion");
        List<String> validaciones =
        listarValidacionRegistro(beanConfiguracion);
        if (configuracion == null) {
            if (validaciones.isEmpty()) {
                if
                (logicConfiguracion.registrarConfiguracion(beanConfiguracion)) {
                    session.setAttribute("sesConfiguracion", configuracion);
                    records.put("msj", "OK");
                } else {
                    records.put("msj", "ERROR");
                }
            }
        } else if (validaciones.isEmpty()) {

```

```
        boolean cambio = verificarCambio(beanConfiguracion,  
configuracion);  
        if (cambio) {  
            beanConfiguracion.setId(configuracion.getId());  
            if  
(logicConfiguracion.modificarConfiguracion(beanConfiguracion)) {  
                session.setAttribute("sesConfiguracion", configuracion);  
                records.put("msj", "OK");  
            } else {  
                records.put("msj", "ERROR");  
            }  
        } else {  
            records.put("msj", "V2");  
        }  
    }  
    records.put("validacion", validaciones);  
    out.print(gson.toJson(records));  
  
    } catch (Exception e) {  
        out.print(gson.toJson("ERROR"));  
    }  
}  
  
}
```