



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO**  
**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE**  
**SISTEMAS**

**TESIS**

EVALUACIÓN DE METODOLOGÍAS DE DESARROLLO WEB BAJO EL  
PARADIGMA DE DESARROLLO DIRIGIDO POR MODELOS (MDD)  
CON LA INTEGRACIÓN DE DIRECTRICES PARA LA CAPTURA DE  
REQUISITOS DE USABILIDAD MEDIDO POR LA ISO / IEC 9126 PARA  
LOGRAR LA SATISFACCIÓN DEL CLIENTE.

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autores:

Ortiz Matute Megan Ashley

Huaman Villanueva Brayan Yerak

Pimentel 15 de Abril del 2017

EVALUACIÓN DE METODOLOGÍAS DE DESARROLLO WEB BAJO EL  
PARADIGMA DE DESARROLLO DIRIGIDO POR MODELOS (MDD) CON LA  
INTEGRACIÓN DE DIRECTRICES PARA LA CAPTURA DE REQUISITOS DE  
USABILIDAD MEDIDO POR LA ISO / IEC 9126 PARA LOGRAR LA  
SATISFACCIÓN DEL CLIENTE.

**Aprobación de la tesis**

---

HUAMAN VILLANUEVA  
BRAYAN YERAK

---

ORTIZ MATUTE  
MEGAN ASHLEY

---

**Ing. Mejía Cabrera Heber Ivan**  
Asesor

---

**Ing. Valdivia Salazar Carlos Alberto**  
Presidente

---

**Ing. Cobeñas Sanchez Rosa América**  
Secretaria

---

**Ing. Tuesta Monteza Victor Alexci**  
Vocal

## DEDICATORIA

*“A mis hermanas Dariam, Alexa y a mis primos  
Nicole y Tiziano, quienes son el futuro de mi  
familia.*

*A mi papá y a toda mi familia por su amor,  
trabajo y sacrificio en todos estos años, gracias a  
ustedes he llegado hasta aquí y convertirme en  
la persona que soy.”*

*Megan Ashley Ortiz Matute*

*Primero agradecer a Dios, por haberme  
permitido llegar hasta este punto y haberme  
dado salud para poder lograr mis objetivos, y  
también agradecer a mis padres por ser el pilar  
fundamental en todo lo que soy, en toda mi  
educación, tanto académica, como de la vida,  
por su incondicional apoyo y a cada uno de mis  
familiares que siempre me han brindado su  
apoyo incondicional.  
Todo este trabajo ha sido posible gracias a ellos.*

*Brayan Yerak Huaman Villanueva*

## AGRADECIMIENTO

*“Con el sentimiento más sincero y lleno de amor a Dios porque guía mis pasos y me regala una vida llena de amor y felicidad. A mi papá, hermanas y abuelos por apoyarme de manera incondicional en todo momento, todos mis logros son suyos”.*

*Megan Ashley Ortiz Matute*

*Primero ante nada agradecer a Dios por guiarme y cuidarme siempre, y darme esas fuerzas que necesito para seguir adelante, también quiero agradecer a cada uno de mis familiares por su apoyo incondicional y por las malas noches que les hice pasar. Las cuales son una fuente de inspiración para así poder lograr cada uno de mis objetivos.*

*Brayan Yerak Human Villanueva*

# ÍNDICE

DEDICATORIA .....	3
AGRADECIMIENTO .....	4
RESUMEN .....	14
ABSTRACT .....	15
INTRODUCCION .....	17
CAPITULO I: PROBLEMA DE INVESTIGACIÓN .....	19
1.1. Situación problemática .....	19
1.2. Formulación del problema .....	21
1.3. Delimitación de la investigación .....	21
1.4. Justificación e importancia .....	22
1.5. Limitaciones de la Investigación .....	23
1.6. Objetivos .....	23
CAPÍTULO II: MARCO TEÓRICO .....	24
2.1. Antecedentes de la investigación .....	24
2.2. Estado del arte .....	36
2.3. Bases teóricas científicas .....	43
2.3.5. Ingeniería de software .....	43
2.3.1.1. Definición de software .....	43
2.3.1.2. Definición de Ingeniería de Software .....	43
2.3.1.3. Proceso de desarrollo de software .....	44
2.3.1.4. Ciclo de vida de software .....	45
2.3.1.4.1. Modelos de ciclo de vida .....	46
2.3.5. Paradigma de desarrollo de software .....	51
2.3.2.1. Definición .....	51
2.3.2.2. Tipos de Paradigmas .....	52
2.3.2.2.1. Lineal y secuencial .....	52
2.3.2.2.2. Evolutivos .....	52
2.3.2.2.3. Prototipado .....	53



2.3.2.2.4.	Ensamblaje de componentes .....	53
2.3.5.	Desarrollo de software dirigido por modelos (MDD).....	53
2.3.3.1.	Introducción .....	53
2.3.3.2.	Paradigma MDD.....	55
2.3.3.3.	El ciclo de vida dirigido por modelos .....	56
2.3.3.4.	Beneficios del ciclo de vida por modelos.....	59
2.3.3.5.	El rol de UML en MDD .....	61
2.3.5.	Metodologías de desarrollo de aplicaciones Web .....	62
2.3.4.1.	Aplicaciones Web.....	62
2.3.4.2.	Atributos de las aplicaciones basadas en la Web .....	63
2.3.4.3.	Atributos de la calidad.....	64
2.3.4.4.	Metodologías de desarrollo Web .....	65
2.3.4.4.1.	NDT .....	65
2.3.4.4.2.	UWE .....	70
2.3.4.5.	Ingeniería de Requerimiento.....	74
2.3.4.5.1.	Definición de requerimientos: .....	74
2.3.4.5.2.	Características de un requisito .....	75
2.3.4.5.3.	Captura de requisitos .....	75
2.3.4.5.4.	Análisis de requisitos: .....	76
2.3.4.5.5.	Clasificación de requisitos .....	76
2.3.4.5.6.	Especificación de requisitos .....	79
2.3.4.5.7.	Validación de requisitos .....	79
2.3.5.	Calidad del Software .....	79
2.3.5.1.	¿Qué es la calidad? .....	79
2.3.5.2.	Calidad de Software.....	79
2.3.5.3.	Tipo de calidad en el desarrollo de un proyecto.....	80
2.3.5.	¿Qué es la ISO? .....	80
2.3.5.1.	¿Qué son normas o estándares? .....	80
2.3.5.3.	Usabilidad.....	83



2.3.5.3.1.	Importancia de la Usabilidad .....	85
2.3.5.3.2.	Ventajas y dificultades en la implantación de sitios web usables	86
2.3.5.3.3.	Mecanismo de usabilidad .....	86
2.3.6.	Herramientas de Desarrollo .....	89
2.3.6.1.	PHP .....	89
2.3.6.2.	Postgres .....	89
2.3.6.3.	XAMP .....	90
2.3.6.4.	NDTSuite .....	90
2.3.6.5.	MagicUWE .....	92
2.4.	Definición de la terminología .....	92
<b>CAPITULO III: MARCO METODOLÓGICO .....</b>		<b>94</b>
3.1.	Tipo y diseño de la investigación.....	94
3.1.1.	Tipo de investigación.....	94
3.1.2.	Diseño de la investigación.....	94
3.2.	Población y muestra .....	95
3.2.1.	Población .....	95
3.2.2.	Muestra.....	96
3.3.	Hipótesis .....	96
3.4.	Variables .....	96
3.4.1.	Variables dependientes.....	96
3.4.2.	Variables independientes.....	96
3.5.	Operacionalización.....	97
3.6.	Abordaje metodológico, técnicas e instrumentos de recolección de datos	100
3.6.1.	Abordaje metodológico.....	100
3.6.2.	Técnicas de recolección de datos.....	100
3.6.3.	Instrumentos de recolección de datos .....	101
	Formato de entrevista .....	101
3.7.	Procedimiento para la recolección de dato .....	102



3.8.	Análisis estadístico e interpretación de los datos .....	104
3.9.	Principios éticos .....	112
3.10.	Criterios de rigor científico .....	113
CAPÍTULO IV: RESULTADOS DE LA INVESTIGACIÓN .....		114
4.1	Resultados de la Evaluación: .....	114
4.2	Discusión de Resultados: .....	133
CAPITULO V: PROPUESTA DE LA INVESTIGACIÓN .....		135
5.1	Identificar los diferentes factores de calidad de la usabilidad. ....	136
5.2	Identificar metodologías de desarrollo web, cuyo enfoque metodológico traten a la usabilidad. ....	139
5.3	Seleccionar las metodologías de desarrollo web, cuyo enfoque metodológico trata requerimientos de usabilidad .....	142
5.4	Aplicar las metodologías con la integración de las directrices para la captura de requerimientos de usabilidad en un caso de estudio.....	145
CAPITULO VI: CONLUCIONES Y RECOMENDACIONES .....		255
6.1	Conclusiones.....	255
6.2	Recomendaciones: .....	258
REFERENCIAS .....		259
ANEXO I .....		267
ANEXO II .....		269
ANEXO III .....		272
ANEXO IV .....		274
ANEXO V .....		277





## INDICE DE TABLAS

Tabla 1: CHAOS, Factores de que determinan el éxito de un proyecto de desarrollo de Software. Fuente: CHAOS REPORT (2015). .....	20
Tabla 2: Comparación de Metodologías de desarrollo Web, según Atributos de calidad. Fuente: Del Valle Rojo & Oliveros (2012,Buenos Aires) .....	26
Tabla 3: Requerimientos No funcionales – Restricciones y las metodologías Web. Fuente: Del Valle Rojo & Oliveros (2012,Buenos Aires) .....	27
Tabla 4: Mapa de las publicaciones seleccionadas. Fuentes: (Ormeño y Panach 2013 - España),.....	40
Tabla 5 Descripción de las características funcionales de usabilidad (FUFs). Basado en el trabajo de (Panach Navarrete, 2010).....	87
Tabla 6: Metodologías de Desarrollo Web. Fuente: Elaboración propia .....	95
Tabla 7: Análisis Estadístico de Tiempo .....	106
Tabla 8: Análisis Estadístico de Total de Presentaciones.....	106
Tabla 9: Análisis Estadístico del Total de Artefactos.....	106
Tabla 10: Análisis Estadístico del Tiempo de Aprendizaje.....	107
Tabla 11: Análisis Estadístico de la Valoración de Aprender.....	107
Tabla 12: Análisis Estadístico del Tiempo de Operación .....	108
Tabla 13: Análisis Estadístico del Comprensión de Funcionalidades.....	109
Tabla 14: Análisis Estadístico del Salida y Entrada de Elementos .....	109
Tabla 15: Análisis Estadístico de Coherencia en Uso.....	110
Tabla 16: Análisis Estadístico de Corrección de errores.....	110
Tabla 17: Análisis Estadístico de la Capacidad de Entender los mensajes de alerta	111
Tabla 18: Análisis Estadístico de Interacción Atractiva Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2 .....	111
Tabla 19: Análisis Estadístico de Cumplimiento de la Usabilidad Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2.....	112
Tabla 20 Tiempo de desarrollo de la metodología NDT.....	114
Tabla 21 Tiempo de desarrollo de la metodología UWE.....	115
Tabla 22 Total de presentaciones realizadas por cada metodología.....	116
Tabla 23 Análisis de las características funcional de Usabilidad, en cada una de las etapas de modelado del sistema de las metodologías en estudio.....	120
Tabla 24 Preguntas para la valorización de aprendizaje.....	121
Tabla 25 Valorización de la facilidad de aprendizaje del sistema.....	122
Tabla 26 Preguntas para obtener el nivel de atraktividad del sistema.....	129



Tabla 27: Valorización de la facilidad de atraktividad del sistema. ....	129
Tabla 28 Preguntas del Cuestionario SUS .....	131
Tabla 29 Análisis de las respuestas del cuestionario SUS .....	132
Tabla 30: Puntuación SUS basada en: (Finstad, 4 Agosto 2016).....	132
Tabla 31: La perspectiva del concepto de usabilidad por los distintos estándares. Fuente, elaboración propia.....	136
Tabla 32: Factores de Calidad de la Usabilidad. Fuente: Elaboración propia .....	137
Tabla 33: Factores de calidad de la Usabilidad, considerados por la SO/IEC 9126, ISO/IEC 9241, ISO 25000, Fuente: Elaboración propia .....	138
Tabla 34 Requisitos que maneja cada Enfoque, Fuente: ESCALONA y KOCH (2004), página 206 .....	139
Tabla 35: Las técnicas utilizadas en la fases de captura, definición y validación, Fuente: ESCALONA y KOCH (2004), página 207 .....	140
Tabla 36: Grado de detalle en los procesos, técnicas y productos. Fuente: ESCALONA y KOCH (2004, página 209).....	141
Tabla 37: Requerimientos identificados por cada metodología. Fuente: (Rojo, 2012, página 77).....	142
Tabla 38: Requerimientos No Funcionales identificados y tratados por cada metodología de desarrollo Web – Parcial. Fuente: (Rojo, 2012, página 78) .....	143
Tabla 39: Técnicas de Ingeniería de requerimientos y herramientas CASE aplicadas a RNF. Fuente: (Rojo, 2012, página 84) .....	143
Tabla 40 System Status Feedback Elaboración propia.....	170
Tabla 41 Definición de elementos a utilizar, para las alertas System Status Feedback .....	179
Tabla 42: Una lista preliminar de características de usabilidad funcionales.....	274
Tabla 43 Los mecanismos de usabilidad para los que se han desarrollado Usabilidad Obtención y Guías de Especificaciones .....	275



## INDICE DE FIGURAS

Figura 1: La Moderna Resolución (OnTime, OnBudget, con un resultado satisfactorio) de todos los proyectos de software desde el año fiscal 2011-2015. En la nueva base de datos CHAOS. Fuente: CHAOS REPORT (2015).....	19
Figura 2: Modelo de ciclo de vida en cascada. Fuente: Ingeniería del software un enfoque desde la guía Swebok (2012), página 40. ....	47
Figura 3: Modelo en “V”. Fuente: (Sánchez, Sicilia, & Rodríguez, 2012, pág. 41) .....	47
Figura 4: Modelo en espiral de Boehn para el proceso del software. Fuente: (Pressman S., 2010). ....	49
Figura 5: Modelo incremental. Fuente: (Pressman S., 2010).....	51
Figura 6: Desarrollo de software dirigido por modelos. Fuente: (Pons, Giandini, & Pérez, 2010). ....	56
Figura 7: Ciclo de vida del desarrollo de software dirigido por modelos. Fuente: (Pons, Giandini, & Pérez, 2010, pág. 31) .....	57
Figura 8: Ejemplo de transformación de PIM a PSM y de PSM a código. Fuente:(Pons, Giandini, & Pérez, 2010, pág. 54) .....	59
Figura 9: Transformación de un modelo PIM escrito en UML a un modelo de implementación escrito en Java.(Pons, Giandini, & Pérez, 2010) .....	62
Figura 10: Árbol de requisitos de calidad Fuente: (Pressman S., 2010). ....	65
Figura 11: Diagrama con el proceso de desarrollo de NDT. (Del Valle Rojo, S. 2012),66	
Figura 12 Diagrama de Proceso de NDT. Fuente: Del Valle Rojo, Silvana (2012) .....	70
Figura 13 Diagrama de Proceso de UWE. Fuente: Elaboración propia, basado en Del Valle Rojo, Silvana (2012).....	74
Figura 14: Requerimientos no funcionales. Fuente: (Sommerville I. , 2005).....	78
Figura 15: Calidad Externa e Interna. Fuente:(Fernández, 2014) .....	83
Figura 16: Calidad en Uso. Fuente: (Fernández, 2014).....	83

## INDICE DE GUI's

GUI 1Modelo de Navegación para Alumnos, NDT .....	225
GUI 2Login del Sistema de Exámenes Online, NDT.....	225
GUI 3Menú del Docente, NDT. ....	225
GUI 4Menú del Alumno, NDT. ....	226
GUI 5Listado de Cursos del docente a cargo, disponibles para Crear Nuevo Examen, NDT.....	226
GUI 6Crear Exámenes, NDT.....	226
GUI 7Recuperación y Listado de Exámenes, NDT .....	227
GUI 8Recuperación y Listado de Exámenes, NDT .....	227
GUI 9Filtros para Generar reporte de Notas, NDT.....	227
GUI 10Recuperación y Listado de Exámenes para Alumnos, NDT.....	228
GUI 11Realizar Examen, NDT.....	228
GUI 12 Recuperar y Listar Ayuda, NDT.....	228
GUI 13 Mostrar Ayuda, NDT .....	229
GUI 14 Registrar configuración de espacio personal, NDT. ....	229
GUI 15 Login del Sistema de Exámenes Online .....	250
GUI 16 Mecanismo de Usabilidad Warning, Sistema de Exámenes Online.....	250
GUI 17 Interfaz principal del sistema, perfil del usuario .....	251
GUI 18 Mecanismo de Usabilidad, User Profile. ....	251
GUI 19 Personalización del espacio del usuario, Choose Skin. ....	252
GUI 20 Mecanismo de Usabilidad, Help. ....	252
GUI 21Mecanismo de Usabilidad, Wizard .....	253
GUI 22: Mecanismo de usabilidad, Warning, confirma el estado de registro correcto. .....	253
GUI 23 Mecanismo de usabilidad Undo cancel para cancelar el examen.....	254
GUI 24 Mecanismo de usabilidad Progress Feedback para el envío del examen .....	254



## INDICE DE GRÁFICOS

Gráfico 1 Comparación del tiempo de desarrollo de las metodologías. ....	115
Gráfico 2 Total de presentaciones realizadas por cada metodología. ....	116
Gráfico 3 Tiempo de duración total de reuniones convocadas por cada una de las metodologías. ....	117
Gráfico 4 Cantidad de artefactos generados por cada una de las metodologías Web. ....	118
Gráfico 5 Cantidad de Artefactos que integran a la Usabilidad. ....	118
Gráfico 6 Cantidad de mecanismo USEP's integrados a las Metodologías. ....	119
Gráfico 7 Tiempo de Aprendizaje. ....	121
Gráfico 8 Análisis de la valorización de la facilidad de aprendizaje del sistema. ....	122
Gráfico 9 Tiempo de Operación de las actividades asignadas al docente. ....	123
Gráfico 10 Tiempo de operación de las actividades asignadas al alumno. ....	124
Gráfico 11 Comprensión de Funcionalidades de acuerdo al tipo de usuario. ....	125
Gráfico 12 Salida y Entrada de elementos de las actividades asignadas a los docentes. ....	126
Gráfico 13 Salida y Entrada de elementos de las actividades asignadas a los alumnos. ....	127
Gráfico 14 Operatividad de los usuarios en el sistema. ....	128
Gráfico 15 Análisis de la valorización de la atractividad del sistema. ....	130
Gráfico 16: Comparación del cumplimiento de usabilidad. ....	131



## RESUMEN

Esta investigación surge debido a que los sistemas de software son cada vez más complejos e importantes en la sociedad actual como hace referencia (ITI-UPV, 2016), informes como el Reporte Chaos publicados en el 2015 manifiesta que la tasa de éxito de los proyectos software, desarrollados en la actualidad, representa el 29% de los 50.000 proyectos evaluados, y el éxito de estos está en representar con claridad y estabilidad los requisitos, para lograr la satisfacción del usuario en base a elaborar un software acorde a sus necesidades, y que sea fácil de aprender, operar, eficiente e intuitivo en el manejo de errores, en otras palabras que tenga el componente de Usabilidad en todos sus aspectos. La usabilidad es un factor de calidad considerado en las últimas etapas del ciclo de vida de desarrollo de software, en la etapa de pruebas, y atender este factor de calidad implica cambios en el producto software, que requieren recursos extras a lo planteado.

El presente trabajo tuvo como objetivo la evaluación de las metodologías de desarrollo web UWE Y NDT, con la integración de directrices USEP's (Usability Elicitation Patterns) para la captura de requerimientos funcionales de usabilidad, para ello se identificó los diferentes factores de calidad de la usabilidad, de un producto software, propuesto por los diferentes estándares como lo son la ISO/IEC 9126, ISO/IEC 9241 y la ISO 25000. Así mismo se identificó y se seleccionó las diferentes metodologías de desarrollo web, para lo cual se tuvo como punto de partida el trabajo de investigación de ESCALONA y KOCH (2004), donde se hace énfasis a los tipos de requerimientos tratados por cada enfoque metodológico de desarrollo de software Web.

Seleccionando las metodologías NDT y UWE, se procedió a integrar los requerimientos funcionales de usabilidad en los artefactos propuestos por cada una de las metodologías. Realizada la integración se evaluó y eligió la metodología más óptima con la cual se siguió a la etapa de codificación, con el fin de obtener un producto software. El cual se procedió a evaluar la usabilidad como característica de calidad externa, medido por la ISO/IEC 9126-2.

Como resultados obtenidos en esta investigación se tuvo que la NDT es la metodología que pudo integrar de una manera más eficiente, en cuanto tiempo y cantidad de mecanismos de usabilidad, propuestos por las USEP's.

Aplicada la evaluación con la ISO/IEC 9126-2 para medir la calidad externa de usabilidad al prototipo funcional, realizado bajo metodología de desarrollo web NDT y la integración de directrices USEP's para la captura de requerimientos de usabilidad en etapas tempranas de desarrollo de software, se logró la satisfacción del cliente.

**Palabras claves:** Usabilidad, metodologías Web, requerimientos funcionales, requerimientos no funcionales, NDT, UWE, MDD, USEPs.

## ABSTRACT

This research arises because software Systems are increasingly complex and important in today's society as a reference (ITI-UPV, 2016), reports such as the Chaos Report published in 2015 show that the success rate of software projects, currently developed, represents 29% of The 50,000 projects evaluated, and the success of these is to represent with clarity and stability the requirements, to achieve user satisfaction based on the elaboration of software according to their needs, and that is easy to learn, operate, efficient and intuitive in The handling of errors, in other words that has the Usability component in all its aspects. Usability is a factor of quality considering in the last stages the software development lifecycle, in the testing stage, and attending this quality factor implies changes in the software product, which requires extra resources to what is raised.

The present work aims to evaluate UWE and NDT web development methodologies, with the integration of Usability Elicitation Patterns (UEP) guidelines for the capture of functional usability requirements, for which different usability quality factors were identified. A software product, proposed by the different standards such as ISO / IEC 9126, ISO / IEC 9241 and ISO 25000. Likewise, the different methodologies of web development were identified and selected, for which the starting point was The research work of ESCALONA and KOCH (2004), which emphasizes the types of requirements addressed by each methodological approach to Web software development.

Selecting the NDT and UWE methodologies, we proceeded to integrate the functional requirements of usability in the artifacts proposed by each of the methodologies. After the integration, we proceeded to evaluate and choose the most optimal methodology with which we followed the coding stage, in order to obtain a software product, which proceeded to evaluate usability as an external quality characteristic, as measured by the ISO / IEC 9126-2.

As results obtained in this research was that the NDT is the methodology that could be more efficiently integrated in the time and quantity of usability mechanisms proposed by the USEPs, and applied the evaluation with ISO / IEC 9126-2 to measure the external quality of usability to the functional prototype, realized under NDT web development methodology and the integration of USEP's guidelines for the capture of



usability requirements in early stages of software development, Customer satisfaction was achieved.

**Keywords:** Usability, Web methodologies, functional requirements, nonfunctional requirements, NDT, UWE, MDD, USEPs.





## INTRODUCCIÓN

Los sistemas de software son cada vez más importantes en la sociedad actual y crecen rápidamente en tamaño y complejidad (ITI-UPV, 2016), estos están dirigidos a un público cada vez más amplio, donde podemos encontrar usuarios menos expertos en el manejo de sistemas informáticos, hace referencia (Ferré Grau, 2001).

Estudios realizados, como el Chaos Report publicado por Standish Group en el 2015 manifiesta que la tasa de éxito de los proyectos software, desarrollados en la actualidad, representa el 29%, y el éxito de estos conlleva a representar con claridad y estabilidad los requisitos, con el fin de lograr la satisfacción del usuario, y que el 71% de proyectos realizados han fracasado o han sido cancelados, porque no terminaron a tiempo, superaron el presupuesto del proyecto o simplemente no era lo que el usuario esperaba, etc.

Según encuesta realizada por Lang y Fitzgerald revela que el 60% de las organizaciones siguen considerando que uno de los principales problemas en sus proyectos está relacionado con la claridad y la estabilidad de los requisitos.

Según las investigaciones realizadas como la de Moreno J. y Marciszack M. (2013) hace referencia que muchas veces los arquitectos de software solo tienen en cuenta a los requerimientos funcionales, dejando de lado los requerimientos no funcionales.

En la investigación realizada por Chian Wen Too & Sa'adah Hassan (2015) resalta que la falta de atención a los requisitos de usabilidad especialmente durante la primera etapa de proceso de desarrollo de software, aumenta el costo de solución de problemas de usabilidad y hace que los productos software finales sean pobres en calidad, es por ello que esta investigación parte de la interrogante ¿Cómo capturar los requerimientos de usabilidad en etapas tempranas del desarrollo de software?

Para responder a esta interrogante se procedió a realizar la selección de las metodologías, para lo cual se tuvo como punto de partida el trabajo de investigación de ESCALONA y KOCH (2004), donde se hace énfasis a los tipos de requerimientos tratados por cada enfoque metodológico de desarrollo de software Web, donde encontramos a los Requerimiento No funcionales, recordando que la usabilidad es una característica de calidad considerada

como un requerimiento no funcional, seleccionando las metodologías de desarrollo web NDT y UWE, con las cuales se realizó la integración de directrices USEP's (Usability Elicitation Patterns), en sus fases de captura de requerimiento con el fin de definir requerimientos funcionales de usabilidad, las cuales fueron incluidas en cada uno de sus modelos.

Realizada la integración se procedió a evaluar y elegir la metodología más óptima de acuerdo al tiempo que requirió desarrollar cada metodología integrando las directrices, de acuerdo a su alcance de las mismas, así como también se analizó la cantidad de artefactos generados por cada una de las metodologías segregando los artefactos que integran a la usabilidad y a los artefactos que no plasman un requerimiento funcional de usabilidad.

También se evaluó a las metodologías bajo el número de mecanismos de usabilidad que estas integraron en cada una de sus fases así como el total de ellas, para luego determinar la metodología con la cual se procedió a la etapa de codificación, obteniendo un producto software el cual se evaluó la usabilidad como característica de calidad externa, medido por la ISO/IEC 9126-2.

Según la ISO/IEC 9126-1, la usabilidad es una de las características que determinan la calidad de los sistemas junto con la funcionalidad, seguridad, eficiencia, mantenibilidad, y portabilidad, dado a que si un sistema no es usable no es de calidad, aunque se haya abordado todos los requerimientos funcionales, hace referencia (Panach Navarrete, 2010). Es por ello que en esta investigación se pretende atender un atributo de calidad que se presenta en la mayoría de clasificaciones, que es la usabilidad. La cual especifica el grado en que los usuarios pueden utilizar un producto de forma gratificante, logrando alcanzar aspectos como efectividad y eficiencia en el contexto de uso. Debido a que la usabilidad es un aspecto crítico en la interactividad de sistemas software, es considerado un factor determinante en la aceptación de los proyectos software, implicando el éxito del mismo.

# CAPITULO I: PROBLEMA DE INVESTIGACIÓN

## 1.1. Situación problemática

Los sistemas de software son cada vez más complejos e importantes en la sociedad actual como hace referencia (ITI-UPV, 2016), estos están dirigidos a un público cada vez más amplio, donde podemos encontrar usuarios menos expertos en el manejo de sistemas informáticos, hace referencia (Ferré Grau, 2001).

Estudios realizados, como el Chaos Report 2015 publicado por Standish Group, son una instantánea del estado de la industria de desarrollo de software, ponen en manifiesto que en la actualidad existen fracasos de Software, como podemos apreciar en la figura N°1, la cual manifiesta que la tasa de éxito de los proyectos, desarrollados en la actualidad, representa el 29% de los 50.000 proyectos analizados, 52% han fallado dado a que estos fueron entregados, culminados tarde, por encima del presupuesto, y / o con menos de las características necesarias y funciones; y el 19% fueron cancelado antes de la finalización o entregado y nunca se utilizaron.

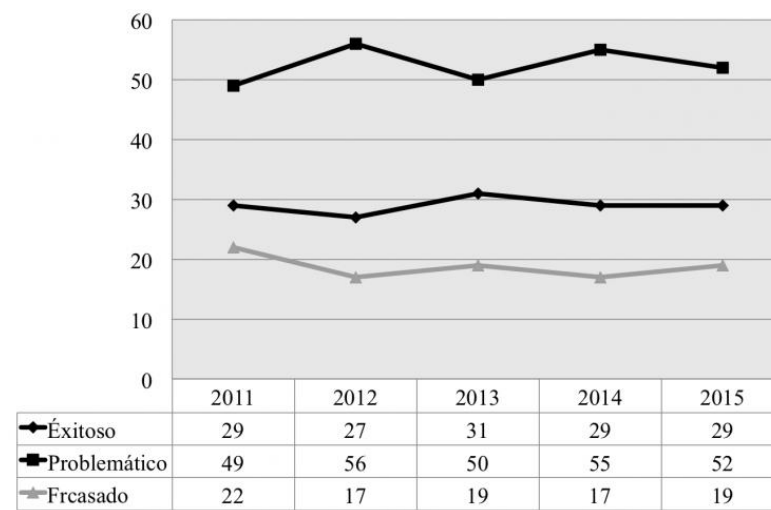


Figura 1: LaModerna Resolución (OnTime, OnBudget, con un resultado satisfactorio) de todos los proyectos de software desde el año fiscal 2011-2015. En la nueva base de datos CHAOS. Fuente: CHAOS REPORT (2015)



El Chaos Report 2015 pone en manifiesto los factores que debe tener un proyecto de desarrollo de software para que este sea considerado exitoso, para ello se evalúan aspectos como: el proyecto fue presentado a tiempo, dentro del presupuesto, con características y funcionalidades requeridas.

Tabla 1: CHAOS, Factores de que determinan el éxito de un proyecto de desarrollo de Software.  
Fuente: CHAOS REPORT (2015).

Factores de éxito	Puntaje	Inversión
Patrocinio ejecutivo	15	15%
Madurez Emocional	15	15%
Participación de los usuarios	15	15%
Optimización	15	15%
Recursos especializados	10	10%
Arquitectura estándar	8	8%
Proceso ágil	7	7%
Ejecución modesta	6	6%
Experiencia en gestión de proyectos	5	5%
Objetivo de negocio claro	4	4%

Otra encuesta realizada por Lang y Fitzgerald revela que en más de 160 organizaciones que desarrollan software basado en la web revela que el 60% de las organizaciones siguen considerando que uno de los principales problemas en sus proyectos está relacionado con la claridad y la estabilidad de los requisitos.

Es por ello que la usabilidad pasa a ser un tema importante a destacar y considerar en el desarrollo de sistemas informáticos, junto con otras características de calidad, desarrollo basado en estándares, con funcionalidad y rendimiento acorde a las necesidades del usuario y exigencias del cliente son aspectos fundamentales para asegurar la aceptación, el éxito del producto software.



Según (Ormeño Y., Panach J., Pastor O, 2012), La usabilidad es considerada en las últimas etapas del proceso de desarrollo de software, cuando las interfaces han sido ya diseñadas. En esta etapa, la inclusión de características de usabilidad puede afectar a la arquitectura del sistema, con la participación tanto de la reanudación de procesos, actividades que ya han sido concluidas, más trabajo. Para minimizar este problema, es de importancia incluir características de usabilidad en la fase de obtención de requisitos.

La usabilidad es un factor crítico y determinante para la aceptación del producto software. Es por ello, que en el presente trabajo de investigación se realizó un esfuerzo por la captura de requisitos no funcionales, especialmente relacionados a la usabilidad, evaluando metodologías de desarrollo web, bajo el paradigma de Desarrollo Dirigido por Modelos (MDD), que tratan los requisitos no funcionales desde la etapa de captura de requisitos, con la integración de Usability Elicitation Patterns (USEPs) para la mejora de la identificación de requisitos de Usabilidad.

## 1.2. Formulación del problema

¿Cómo capturar requisitos de usabilidad en etapas tempranas del desarrollo de software para lograr la satisfacción del cliente?

## 1.3. Delimitación de la investigación.

- a) Esta investigación se desarrolla en las instalaciones del laboratorio de investigación y sistemas inteligentes y seguridad informática (LABSIS) con la ayuda de nuestro asesor especialista, la cual se llevó acabo en un periodo de 12 meses habiéndose iniciado en el mes de Enero del 2016.
- b) Los productos obtenidos (artefactos) por las metodologías UWE y NDT son desarrollados bajo los requerimientos solicitados por la Institución Educativa Privada "Santa María de la Paz", ubicada en la Calle Lambayeque cuadra 1 Pimentel, Lambayeque Perú, sobre la elaboración de un Sistema Web de Exámenes en Online. El cual se

desarrollará siguiendo los procedimientos y herramientas propuestas por las dos metodologías mencionadas anteriormente, con la integración de las directrices USEPs.

#### 1.4. Justificación e importancia

Con el presente proyecto se pretende atender un atributo de calidad que se presenta en la mayoría de clasificaciones, la usabilidad. La cual especifica el grado en que los usuarios pueden utilizar un producto de forma gratificante, logrando alcanzar aspectos como efectividad y eficiencia en el contexto de uso. La usabilidad es un aspecto crítico en la interactividad de sistemas software, por ello es un factor determinante en la aceptación de los proyectos software, implicando el éxito del mismo.

Moreno J. y Marciszack M. (2013) hace referencia que en el proceso de elicitación de requerimientos, los arquitectos del software dan una mayor atención a los requerimientos funcionales, dejando de lado los Requerimientos No funcionales. Los cuales no son considerados ni especificados durante el ciclo de vida de desarrollo de software, ya sea porque no existen herramientas, procedimientos, métodos y técnicas para la especificación y prueba para este tipo de requerimientos, particularmente en este caso, destinados al entorno Web. Uno de los requerimientos no funcionales que se analizó en esta investigación fue la Usabilidad, dado a que es un aspecto esencial de calidad que se debe tener en cuenta en los productos de software para que estos sean más usables y fáciles de aprender.

Por otro lado tenemos a Chian Wen Too & Sa'adah Hassan (2015), quienes manifiestan en su trabajo de investigación que la falta de atención a los requisitos de usabilidad, especialmente durante la primera etapa de proceso de desarrollo de software, aumentan el costo de la solución de problemas de usabilidad, y hace que los productos de software finales sean pobres en calidad.

A menudo la usabilidad en el ciclo de vida de software es tratada en la etapa de pruebas del desarrollo de software, implicando retraso y aumento de costo en el proyecto, y en muchos otros casos al cancelación del mismo así como la pérdida de patrocinadores (sponsors).

### 1.5. Limitaciones de la Investigación

- a) Las metodologías de desarrollo Web, NDT y UWE, son producto de investigaciones Doctorales, por ello la información encontrada sobre dichas metodologías son en su mayoría producto de investigaciones desarrolladas por otros autores. Lo que implica el reducido número de casos de aplicación reales, concisos, y en diversos contextos.
- b) Los requerimientos atendidos e implementados bajo las metodologías propuestas en esta investigación, fueron seleccionados, dentro de una gama de requerimientos solicitados por el usuario, por convicción propia de los investigadores, con el fin de que estas ayuden a lograr el objetivo propuesto.

### 1.6. Objetivos

#### Objetivo general

Evaluar metodologías de desarrollo Web bajo el paradigma de Desarrollo dirigido por modelos (MDD) con la integración de directrices para la captura de requisitos de usabilidad medido por la ISO/IEC 9126 para lograr la satisfacción del cliente.

#### Objetivos específicos

- a. Identificar los diferentes factores de calidad de la usabilidad.
- b. Identificar metodologías de desarrollo web, cuyo enfoque metodológico traten la usabilidad.
- c. Seleccionar las metodologías de desarrollo web, cuyo enfoque metodológico trata requerimientos de usabilidad.
- d. Aplicar las metodologías con la integración de las directrices para la captura de requerimientos de usabilidad en un caso de estudio.

## CAPÍTULO II: MARCO TEÓRICO

### 2.1. Antecedentes de la investigación

Wen Too y Hassan (2015- Malaysia) en su investigación “*The Adoptions of USEPs in Identifying Usability Requirements in SSM-Based Framework: A Case Study*”, identificación requerimientos de usabilidad en la etapa de captura de requisito de software mediante el uso del marco o Framework, basado en la Metodología de sistemas blandos, SSM-based.

La facilidad de uso, que es un tipo de requisito no funcional (NFR) es difícil de especificar y caracterizar en comparación con el requisito funcional (FR). NFR tiene un gran impacto en la calidad del producto software, y el no tratarlos debidamente en etapas tempranas conlleva a fallos, retrasos o aumentos significativos en el costo. Hoy en día la comunidad de investigación en ingeniería de software es consciente de cuán importante es prestar una atención significativa al tratamiento relacionado con NFR. Debido a esta razón el trabajo se centró con en la facilidad de uso en la primera etapa del proceso de desarrollo de software.

Para ello llevaron a cabo un caso de estudio donde aplicaron un modelo de obtención de usabilidad, (USEP'S), que es apto para ser adoptado por el marco propuesto y puede ayudar a mejorar las actividades de la fase de análisis para especificar e incorporar características de funcionalidad de uso para determinar una funcionalidad correcta. Además, se realizaron algunas mejoras y extensiones al marco basado en los resultados del estudio. Son necesarias algunas modificaciones para ampliar la información de los patrones existentes mediante la especificación de temas adicionales que se discutió durante el proceso de obtención de ciertos mecanismos. Las tareas a futuro mencionadas en esta investigación fueron el desarrollar una herramienta de software con el fin de apoyar el proceso de



obtención del requisito de usabilidad junto con el requisito funcional, junto con la adopción de los pasos en la herramienta para proporcionar orientación a las partes interesadas. Este trabajo de investigación de Wen Too y Hassan (2015- Malaysia) es un aporte para la mejorar calidad de productos software, en función de los requisitos de usabilidad y, al mismo tiempo, resolver el problema de la identificación tardía del requisito de usabilidad que es comúnmente sucede en el proceso de desarrollo de software.

El trabajo destaca por ilustrar un ejemplo de adopción de directrices para la captura de requisitos de usabilidad, USEP's, durante etapas tempranas de desarrollo de software, la cual sirve de referencia para esta investigación, dado a que podemos observar como las directrices tomadas para el desarrollo de esta investigación están siendo integradas con otra metodología de desarrollo web, Metodología de sistemas blandos, SSM-based.

Del Valle Rojo & Oliveros (2012,Buenos Aires) en su investigación "Requerimientos no funcionales para aplicaciones web". El objetivo principal es el estudio comparativo de los diferentes tipos de Requerimientos No Funcionales, técnicas utilizadas para su elicitación, especificación, validación y gestión en metodologías de Desarrollo de aplicaciones Web.

La investigación de Rojo está basada en el trabajo de Escalona y Koch, quien aporta una comparación de diez metodologías de desarrollo web, destacando entre otros aspectos, técnicas de ingeniería de requerimientos y tipos de requerimientos que manejan. En esta comparación extrajeron el análisis comparativo con sus diferentes tipos de requerimientos que son tratados por cada una de las metodologías existentes, dentro de cuales se identificaron las metodologías de desarrollo de software web que consideran a los requerimientos no funcionales, como lo fueron: Web Site Design Method (WSDM), Hypermedia Flexible Process Modeling(HFPM), UML-based Web

Engineering (UWE), Web Modeling Language (WebML), Navigational Development Techniques (NDT) y Design-Driven Requirements Elicitation (DDDP).

Una vez identificadas se analizaron los procesos de desarrollo de las metodologías WSDM, HFPM, UWE, WebML, NDT, DDDP, identificando los tipos de requerimientos no funcionales contemplados por cada uno, así como también las técnicas de ingeniería de requerimientos que proponen para el tratamiento y gestión de este tipo de requerimientos. Estableciéndose cual/es fase/s del ciclo de vida de desarrollo de software son tratados los Requerimientos No Funcionales.

Uno de los indicios encontrados en esta investigación es que la usabilidad es la característica del modelo ISO/IEC 9126 considerada por todos procesos, y que la usabilidad, performance y seguridad, son las características más relevantes, como lo podemos observar en la Tabla 2.

Tabla 2: Comparación de Metodologías de desarrollo Web, según Atributos de calidad. Fuente: Del Valle Rojo & Oliveros (2012, Buenos Aires)

	Atributos de calidad ISO/IEC 9126-1						Otros				
	Usabilidad	Performance (1)	Seguridad (2)	Confiablez	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Reutilización	Fiabilidad	Disponibilidad	Escalabilidad
WSDM	✓		✓		✓						
HFPM	✓	✓						✓			
UWE	✓		✓								
WebML	✓	✓	✓		✓					✓	✓
NDT	✓					✓			✓		
DDDP	✓	✓	✓				✓				

(1) Subcaracterística de la Eficiencia

(2) Subcaracterística de la Funcionalidad

(3) Subcaracterística de la Portabilidad



Otros puntos a destacar en esta investigación con respecto a los requisitos no funcionales como restricciones en las metodologías web fueron las siguientes:

- a. WebML marca un enfoque completo y a detalle que cubre todas las fases del ciclo de vida de desarrollo de un producto software web.
- b. WSDM, HFPM, UWE, DDDP presentan un proceso de desarrollo completo, sin embargo el nivel de definición de cada proceso es parcial.
- c. NDT marca un proceso de desarrollo parcial. Aunque el nivel de detalle de la definición del proceso es completa dado a que detalla las actividades y las técnicas que propone.

Tabla 3: Requerimientos No funcionales – Restricciones y las metodologías Web. Fuente: Del Valle Rojo & Oliveros (2012,Buenos Aires)

	Restricciones										
	Restricciones de proceso de desarrollo				Otras						
	Alcance del proceso		Detalle de proceso		Lenguajes		Técnicas				
	Parcial	Completo	Parcial	Completo	Lenguajes de modelado	Lenguaje de implementación	Representación interna de datos	Control de versiones,	Protocolos de interface	Comunicación	De ambiente de desarrollo
WSDM		✓	✓			✓					
HFPM		✓	✓		✓						
UWE		✓	✓		✓						
WebML		✓	✓		✓	✓					
NDT	✓			✓	✓					✓	✓
DDDP		✓	✓		✓	✓	✓	✓	✓		

Esta investigación nos otorga la evaluación de 10 metodologías de desarrollo, bajo una perspectiva comparativa de los tipos de Requerimientos No Funcionales, dentro de ellas encontramos la usabilidad; y las técnicas utilizadas para su elicitación, especificación, validación y gestión por las metodologías de desarrollo de aplicaciones Web. Los resultados obtenidos de la investigación de Del Valle Rojo & Oliveros (2012,Buenos Aires)sirvieron para analizar y seleccionar dentro



de las metodologías de desarrollo Web, las metodologías Navigational Development Techniques (NDT) y UML – Bases Web Engineering (UWE), con las que se trabajó en la presente investigación.

Según Del Valle Rojo y Oliveros (2013 - Argentina) en su investigación “*Elicitación y especificación de requerimientos no funcionales en aplicaciones web*”. El objetivo es desarrollar técnicas y lineamientos para los procesos de elicitación y especificación de la Ingeniería de Requerimientos, que sean aplicables al tratamiento de los requerimientos no funcionales en un marco de desarrollo de aplicaciones Web.

La cual presenta dos aportes significativos. En primer lugar provee procesos para la elicitación y la especificación de los requerimientos no funcionales de aplicaciones Web y un patrón para escribir requerimientos no funcionales. En segundo lugar provee todas las plantillas definidas como soporte de los procesos de elicitación y especificación construidas en formato Excel, con los lineamientos particulares para que puedan ser completadas por un Ingeniero de requerimientos, ya sea para llevar a cabo el proceso de elicitación de los RNF, donde se proveen un marco de referencia para la labor de un ingeniero de requerimientos, plantilla para llevar el control de autorización de acceso, plasmar las restricciones, especificación de los requerimientos, así como también una plantilla que permite dar seguimiento del estado de los RNF durante la ejecución de los procesos de desarrollo.

La tipificación de RNF de calidad y restricciones propuestas, permite analizar, obtener y documentar un modelo de RNF completo en la fase inicial del ciclo de vida de desarrollo.

El enfoque plasmado en esta investigación para la gestión de los requerimientos no funcionales de aplicaciones Web esta soportado por la propuesta de una serie de documentos, los cuales fueron aplicados en un caso real, el cual consistía en el desarrollo de un aplicativo Web para una empresa consultora, la cual definió todas las reglas del servicio comercial del aplicativo. La aplicación de la propuesta de Del Valle Rojo

y Oliveros afirma que con este proceso es posible la elicitación y especificación de los RNF de aplicaciones Web, mediante el uso de las plantillas demostró la utilidad para capturar conocimiento y como guía durante la búsqueda y adquisición del conocimiento relevante del dominio del problema relacionado a los RNF.

En esta investigación se presentan a las metodologías de desarrollo Web, una descripción general y las herramientas, actividades a realizar dentro de cada etapa de desarrollo. Así como también se puede observar que la gran mayoría de las metodologías que existen están centradas en el diseño de sistemas web en comparación con las que contemplan la especificación de requisitos.

Panach (2010 - España) es su investigación "*Incorporación de Mecanismos de Usabilidad en un Entorno de Producción de Software Dirigido por Modelos*". El objetivo principal fue obtener un método llamado MIMAT para incorporar Functional Usability Features (FUF) existentes dentro de un método de desarrollo Model-Driven Development (MDD).

Esta investigación está basada en los mecanismos de usabilidad definidos por (Juristo N., Moreno A., Sanchez M., 2007), definidos para ser incorporados en un proceso tradicional de desarrollo de software. Las propuestas por Juristo fueron incorporadas en un método propuesto por Panach llamado MIMAT, donde las guías de captura de requisitos son utilizadas para extraer qué propiedades de los mecanismos que se deben representar en los modelos conceptuales. Por otro lado, se hizo uso de patrones de diseño para la definición de reglas de transformación de modelo a código utilizadas para implementar la funcionalidad de los mecanismos de usabilidad.

El método MIMAT consta de cuatro pasos: definición de formas de uso, identificación de propiedades, definición de primitivas conceptuales, identificación de los cambios necesarios en el compilador de modelos. Cabe mencionar que los dos primeros pasos son válidos para cualquier Tecnologías de Transformación de Modelos (TTM), y los dos últimos mecanismos mencionados dependen de la tecnología de

transformación de modelos sobre la cual se aplique MIMAT, ya que cada tecnología de transformación de modelos tiene un modelo conceptual y compilador de modelos propios.

Como ejemplo de Tecnologías de Transformación de Modelos (TTM) para aplicar MIMAT, en la investigación de Panach, se eligió OO-Method, método de desarrollo software que soporta el estándar de Arquitectura Dirigida por Modelos (MDA), debido a que dispone de una herramienta industrial que la implementa (llamada OlivaNOVA).

El modelo conceptual de OO-Method es lo suficientemente abstracto como para incorporar nuevas primitivas con semántica relacionada con la usabilidad. Para cada uno de los mecanismos de usabilidad definidos por Juristo, se han extraído una serie de propiedades, las cuales provocaron cambios en los modelos conceptuales y en el compilador de modelos de OO-Method. Los cambios que se realizaron en esta investigación han sido documentados para luego sirvan de ejemplo, para poder aplicar MIMAT a otras TTMs.

En esta investigación se demostró que el método propuesto, MIMAT, puede ser aplicado a TTMs reales y el esfuerzo por mejorar la usabilidad en las aplicaciones se ha alcanzado, usando como ejemplo la metodología OO-Method.

El autor señala que la creación de prototipo ayudan a que el analista de OO-Method pueda añadir más propiedades de usabilidad de forma abstracta representados mediante primitivas conceptuales, así como también dan una idea de cómo se podrían añadir las propiedades de usabilidad a otras TTMs distintas de OO-Method

En este experimento participaron tanto usuarios que interactuaron previamente con aplicaciones generadas por OlivaNOVA como usuarios sin experiencia previa en este tipo de aplicaciones. En cada uno de estos grupos hubieron usuarios que interactuaron con una aplicación que incorporaba mecanismos de usabilidad y otros usuarios han interactuado con la misma aplicación pero sin incorporar dichos mecanismos. De esta manera, se ha podido comparar si la usabilidad

percibida por el grupo que interactuó con la aplicación que usaba mecanismos de usabilidad era mejor o no con respecto al grupo que interactuó con la aplicación que no usaba mecanismos.

La investigación Panach destaca por aplicar un método llamado MIMAT haciendo uso de guías de captura de requisitos y los patrones de diseño en una metodología de desarrollo web. Tomamos como referencia a esta investigación porque podemos observar el uso de los Functional Usability Features (FUF) dentro para la captura de requisitos provocando cambios en los modelos conceptuales y en el compilador de modelos de la metodología OO-Method.

Ignacio Panach, Juristo, Valverde, & Pastor (España, 2015) en su investigación "*A framework to identify primitives that represent usability within model-driven development*". El propósito de esta investigación fue el incluir características funcionales de la usabilidad, las cuales son recomendaciones de usabilidad fuertemente relacionadas con la funcionalidad del sistema, en el método de Desarrollo Dirigido por Modelos (MDD) a través de primitivas conceptuales.

En esta investigación se identificó que el Desarrollo bajo el paradigma Model-Driven (MDD) establece que todos los esfuerzos de los analistas deben ser reunidos en el modelo conceptual y el sistema a desarrollar se implementa por medio de reglas de transformación, las cuales pueden automatizarse, y estas generan la mayor cantidad de código en un lenguaje específico, todo a partir de los modelos conceptuales bien establecidos. Hoy en día, existen varias herramientas que implementan el paradigma MDD, como WebRatio, UWE, NDT y OO-Method, entre otros. Todas estas herramientas son muy poderosas para representar y generar la funcionalidad del sistema, el comportamiento y persistencia a través de modelos conceptuales. Sin embargo, en la mayoría de los métodos de MDD, hay una falta de expresividad para representar características de uso.

La problemática radica en que si estas características han de incluirse en los sistemas desarrollados por estos métodos MDD, el código generado, por herramientas que procesan los modelos

conceptuales a través de transformaciones, debe ser cambiado manualmente. Estos cambios manuales implican algunas desventajas como:

- a) Los cambios en el código pueden ser incompatibles con las características expresadas en el modelo conceptual.
- b) Cada vez que se regenera el código del modelo conceptual, los cambios manuales el código debe ser aplicada.
- c) Comprender el código para mejorar la facilidad de uso del sistema puede ser difícil para el analista

Con el fin de superar todos estos problemas, se propuso la inclusión de características de usabilidad en un modelo conceptual de manera similar a lo que se hace actualmente con la funcionalidad, el comportamiento y persistencia en la mayoría de los métodos de MDD. Esta propuesta es un paso adelante para incorporar características de los sistemas de software sin combinar, hasta la fecha, en los métodos de MDD.

En el pasado, muchos autores de Ingeniería de Software han considerado a la usabilidad como un requisito no funcional. Sin embargo recientemente algunos autores han identificado que varias características de usabilidad son fuertemente relacionadas con la funcionalidad. Es por ello que en esta investigación se centra en estas características, ya que no sólo afectan la interfaz, sino también la arquitectura, y son difíciles de tratar si no se consideran en las primeras etapas de desarrollo. La contribución de este trabajo es la definición de un proceso para representar características de uso funcionales en un modelo conceptual de tal manera que un modelo compilador puede generar automáticamente su código.

Los beneficios de la incorporación de características de usabilidad funcionales en un método MDD a través de primitivas conceptual son:

- a) Inequívocamente definida características de usabilidad funcionales, se podrá realizar los modelos conceptuales luego estos modelos podrán ser transformados a código.



- b) Reducción del esfuerzo de desarrollo, con respecto a la inclusión de características de uso a mano, en el código generado por el compilador.
- c) Si se requiere en cambio o la inclusión de requisitos de funcionalidad estos solamente necesita ser agregados por los modelos conceptuales ya no desde código, por lo tanto el modelo será capaz de evolucionar más fácilmente.

La propuesta para incluir características de uso es válida para cualquier método MDD. Sin embargo para fines de estudio los investigadores. Eligieron el método OO-Method, ya que se apoya en una herramienta comercial que se está utilizado regularmente para desarrollar sistemas reales de una empresa (Integranova).

Para ello se pretende identificar las características de usabilidad a través de directrices, estas características son representadas en un modelo conceptual, para que luego estas puedan servir como entradas para un compilador de modelos, que genera el código de acuerdo a las características expresadas en ellos. Para ello se llevó a cabo un estudio empírico con 66 personas quienes hicieron uso de una aplicación Web de alquiler de autos, con el fin de estudiar el efecto de incluir características de usabilidad funcionales con respecto a la satisfacción y el tiempo de los usuarios finales para completar las tareas. Por otra parte, se comparó la carga de trabajo de los dos analistas MDD, los cuales incluyeron características de uso a mano en el código generado frente a incluirlos a través de las primitivas conceptuales de acuerdo al enfoque propuesto en esta investigación.

Los resultados del estudio empírico, mostraron que después de incluir las características de usabilidad se logró la satisfacción de los usuarios finales, mientras que el tiempo de desarrollo no tuvo cambio significativo, con la inclusión de características funcionales de la usabilidad, en el proceso de desarrollo. Los resultados de la comparación muestran que la carga de trabajo necesaria para adaptar el

método MDD para apoyar la usabilidad través de las primitivas conceptuales es pesada.

Sin embargo se ha concluido que MDD soporta la integración de estas características de usabilidad, los analistas que trabajaron con el método MDD y la integración de características funcionales de la usabilidad, son más eficientes que los analistas MDD quienes trabajaron la integración de las características funcionales de forma manual, cuando el código ya había sido generado y se realizaron las configuraciones de forma manual.

El trabajo realizado por (Ignacio Panach, Juristo, Valverde, & Pastor, 2015) sirve como referencia para esta investigación, dado a que muestra un caso de estudio, la cual ha sido desarrollo bajo una metodología de desarrollo Web que es la OO-Method, que dirige el proceso de desarrollo de software bajo el paradigma MDD, propio de la familia de las metodologías seleccionadas para este trabajo. Las perspectivas de las características de usabilidad que se pretende capturar y procesar en los modelos conceptuales de esta investigación está basados bajo las recomendaciones FUF's, que son el conjunto de recomendaciones que hacen referencia a las características de usabilidad como funcionales, las cuales toma como referencia las directrices a utilizar en esta investigación, los patrones de elicitación de Usabilidad (USEPs).

En el contexto nacional encontramos investigaciones referentes a la usabilidad, como las siguientes:

Paz (2013 - Perú) en su investigación “Heurísticas de usabilidad para sitios web Transaccionales”, tiene como objetivo principal elaborar un conjunto de principios heurísticos que permitan realizar de forma efectiva evaluaciones de usabilidad de Sitios Web Transaccionales. Las heurísticas más reconocidas y utilizadas en la actualidad para evaluar el nivel de usabilidad de sitios Web transaccionales presentan falencias e inducen a errores en los resultados, motivo por el cual se determina que

las actuales heurísticas ya no son instrumentos apropiados y fiables para evaluar esta característica de calidad en las nuevas y emergentes categorías de aplicaciones de software.

Este hecho ha sido el móvil para plantear la siguiente interrogante como tema de investigación: ¿Qué elementos esenciales deberían considerarse en la elaboración de un nuevo conjunto de principios heurísticos de evaluación que permitan medir con mayor precisión el nivel de usabilidad de un sitio Web transaccional?

Para solucionar la problemática descrita en párrafos anteriores, se elaboró una propuesta metodológica basada en el análisis de datos de distintos casos de estudio que como resultado presentan un conjunto de principios heurísticos, los cuales han sido desarrollados considerando que pueden ser utilizados por evaluadores con distintos niveles de experticia en evaluaciones de usabilidad.

El objetivo de este nuevo conjunto de heurísticas de usabilidad propuesto estuvo orientado a la obtención de principios que sean entendibles, fáciles de utilizar y que a su vez permitan realizar de forma efectiva evaluaciones de usabilidad de Sitios Web Transaccionales. La metodología propuesta está enmarcada básicamente en la mejora del actual instrumento de evaluación propuesto por Jakob Nielsen y otros aspectos que han sido tomados de la literatura.

El alcance de la investigación se enmarca dentro del paradigma mixto debido a que la primera fase del trabajo se centró en demostrar la existencia de la problemática planteada a través de un análisis cualitativo de la evidencia obtenida en estudios de casos y en la elaboración de la propuesta heurística de evaluación que cubre aquellos aspectos que no están siendo considerados en la actualidad. Posteriormente, la realización de una evaluación heurística ha permitido validar mediante la aplicación en la práctica, la nueva propuesta desarrollada por el autor del trabajo de investigación; con lo que se consolida un nuevo conocimiento que aporta al bagaje cultural de la ciencia.

En esta investigación se aplicó métodos heurísticos para la evaluación de la usabilidad en sitios web transaccionales para determinar qué instrumentos son apropiados y fiables para evaluar la calidad, la cual sirve como referencia para esta investigación, ya que podemos observar cómo puede ser evaluado la usabilidad en sitios web transaccionales.

## 2.2. Estado del arte

(Ormeño y Panach 2013 - España), en su investigación “Mapping Study about Usability Requirements Elicitation” tiene como objetivo de revisar los estudios existentes relacionados con los requisitos de usabilidad tanto por la comunidad de Computer Interaction (HCI) y el campos de Ingeniería de Software (SE). Y ante ello manifiesta que el objetivo de desarrollar sistemas utilizables ha sido tratado por el Human Computer Interaction (HCI) y los campos de Ingeniería de Software (SE). En ambas comunidades, la facilidad de uso es generalmente considerado en las últimas etapas del proceso de desarrollo de software, cuando las interfaces ya han sido diseñados. Incluir características de usabilidad en estas últimas etapas podría afectar a la arquitectura del sistema. Para minimizar este problema, la facilidad de uso debe ser incluido en la etapa de obtención de requisitos. La comunidad SE tiene una amplia experiencia en los requisitos de obtención temprana y métodos para estos. Sin embargo, estos métodos se centran principalmente en los requisitos funcionales y requisitos no funcionales (NFR) históricamente han sido olvidados en esta primera etapa.

De acuerdo a muchos autores, el cumplimiento de los requisitos funcionales no es suficiente para crear un producto de calidad. La usabilidad es un factor clave en la obtención de buenos índices de aceptación. En este estudio, se propone identificar los métodos existentes para la captura de requisitos de usabilidad. Para ello, llevamos a cabo un estudio de mapeo (MS) en base a los trabajos realizados por Kitchenham. Un MS proporciona un procedimiento cuyo

objetivo es la identificación de la naturaleza y el alcance de la investigación que está disponible para responder a una pregunta en particular. Estos estudios también son útiles para identificar lagunas en la investigación actual y sugerir áreas para una mayor investigación. De todos los métodos de desarrollo de software, nos centramos en el Desarrollo Dirigido por Modelos (MDD). MDD tiene como objetivo desarrollar el software por medio de un modelo conceptual, que es la entrada para un compilador modelo que genera la implementación del código del sistema. La comunidad SE ha estado trabajando con este paradigma, y, hoy en día, hay métodos y herramientas (por ejemplo OO-Method, WebRatio, OOHDM).

Sin embargo, para conocimiento de los autores, ninguno de estos métodos de tratan con la facilidad de uso. En general, los métodos existentes MDD se ocupan de la facilidad de uso cuando se han definido los modelos que representan los requisitos funcionales y el código ha sido generado. En esta etapa, si el analista necesita mejorar la facilidad de uso del sistema, el código debe ser modificado manualmente. Por otra parte, algunos cambios requieren volver a trabajarse la arquitectura del software. Estas son las razones por las cuales se deben hacer más esfuerzos para incluir la facilidad de uso en los métodos de MDD, y esto MS pretende ser un paso adelante esta dirección.

El objetivo a largo plazo es doble:

1. Mejorar las prácticas actuales de captura de requisitos de usabilidad.
2. Mejorar los métodos existentes MDD para tener en consideración la captura de requisitos de usabilidad. La MS puede ayudarnos a identificar las ventajas y desventajas de cada método de captura existente, como paso previo para nuestro objetivo. Sin embargo, la MS no es exclusivo de MDD; se puede analizar en detalle cualquier método de desarrollo de software que incluya la captura o elicitación de requisitos de usabilidad.

Este MS combina aspectos de usabilidad tanto de la comunidad de Ingeniería de Software (SE) y la comunidad Human-Computer Interaction (HCI). Se exploró los métodos de desarrollo que tengan en cuenta la facilidad de uso como una exigencia de la comunidad SE. Se ha estudiado las directrices y la heurística de la comunidad de HCI que se utilizan para desarrollar aplicaciones utilizables. La MS tiene como objetivo revisar los estudios existentes relacionados con los requisitos de usabilidad en ambas comunidades, sobre todo en las propuestas por obtener los requisitos de usabilidad desde las primeras etapas del proceso de desarrollo de software.

La MS se ha realizado de acuerdo con la metodología Kitchenhams, centrándose en los últimos 12 años. Un total de 29 publicaciones fueron seleccionadas de un grupo inicial de 150 publicaciones devueltas por la cadena de búsqueda. Las evaluaciones de la calidad de las publicaciones fueron desarrolladas con el fin de contrastar la importancia de las publicaciones seleccionadas, donde el 97% se compone de buenas publicaciones y muy buenas publicaciones.

Utilizando los resultados de la MS, se puede concluir que existe una clara línea de investigación en el campo de los requisitos de usabilidad en entornos MDD. Por lo general, los métodos de MDD históricamente se han enfocado en el comportamiento modelado y persistencia, pero relegando interacción (y en particular la facilidad de uso) para una aplicación manual. Esta aplicación manual de clara contradicción con el paradigma MDD, que defiende que el analista debe trabajar con modelos conceptuales holísticos, donde cada característica del sistema (incluyendo características de uso) podría estar representada.

La estrategia aplicada para la extracción de datos consistió en clasificar las posibles respuestas para cada subpregunta. Las clasificaciones se definen para facilitar la respuesta a la pregunta de investigación. Estos son:

- a) SQ1 Métodos para provocar los requisitos de usabilidad. A) si B) no
- b) SQ2 Métodos que trate de obtener requisitos de interacción. A) si B) no
- c) SQ3 Directrices para la captura de requisitos de usabilidad. A) Existe b) Nuevos c) no existen.
- d) SQ4 Herramientas de apoyo a la obtención requisitos de usabilidad A) Diseño de interfaz (asistente para diseñar) B) el desarrollo de modelos C) no existe
- e) SQ5 Notaciones que trate de obtener los requisitos de usabilidad. A) UML B) Lenguaje Natural (sesiones de trabajo, listas de verificación, cuestionarios, heurística, de intercambio de ideas, o inter-visitas) C) i \* marcos D) CTT (Concur Task Trees) E) Formal. (Operadores lógicos o gramáticas) F) QOC (Criterios opción de la pregunta) G) BPMN H) no existe
- f) SQ6 Entorno de validación empírica. A) Industrial) B) Académico C) no existe

Se resumen los resultados más relevantes para cada subpregunta en la siguiente gráfica:

Tabla 4: Mapa de las publicaciones seleccionadas. Fuentes: (Ormeño y Panach 2013 - España),

SQ1		SQ2		SQ3			SQ4			SQ5					SQ6			Quality Score	ID			
A	B	A	B	A	B	C	A	B	C	A	B	C	D	E	F	G	H			A	B	C
X	X	X					X			X								X			5,00	[10]
X	X	X					X	X										X			3,67	[37]
X	X	X					X										X	X			7,00	[27]
X	X	X					X										X	X			1,00	[26]
X	X					X	X						X					X			-1,00	[6]
X	X	X					X			X	X	X						X			1,33	[21]
X		X				X	X	X										X			3,67	[24]
X	X	X					X			X								X			1,00	[4]
X	X	X					X	X	X									X			0,00	[51]
X	X	X					X	X	X									X			-0,33	[17]
X	X	X					X	X	X									X			-0,67	[35]
X	X					X	X	X	X	X					X			X			3,00	[45]
X		X	X				X	X	X	X								X			4,67	[14]
X	X	X				X									X			X			-0,33	[32]
X		X				X	X			X								X			-2,00	[3]
X	X	X					X	X										X			0,33	[42]
X	X	X					X											X			0,33	[44]
X	X	X					X	X										X			0,67	[8]
X	X	X	X				X									X	X				4,00	[22]
X	X	X					X									X	X				-2,67	[43]
X	X	X					X	X	X									X			2,67	[46]
X	X	X	X				X						X					X			5,00	[47]
X	X		X	X			X	X										X			2,67	[15]
X	X		X	X			X	X										X			0,33	[16]
X	X		X	X			X	X					X					X			4,00	[38]
X	X	X					X										X	X			1,33	[30]
X		X	X	X			X							X			X				7,67	[48]
X	X	X					X	X	X	X				X				X			6,67	[9]
X	X	X					X	X										X			4,00	[25]

SQ1: A) Yes 24.14% B) No 75.86%; SQ2: A) Yes 17.24% B) No 82.76%; SQ3: A) Existing 31.03% B) New 24.14% C) Not Exist 44.83%; SQ4: A) Interface Design 17.24% B) Model Development 24.14% C) Not Exist 58.62%; SQ5: A) UML 41.38% B) Natural Language 27.59% C) i\* 27.59% D) CTT 13.79% E) Formal 6.9% F) QOC 6.9% G) BPMN 3.45% J) Not Exist 17.24% SQ6: A) Industrial 10.34% B) Academic 58.62% C) Not Exist 31.03%.

Benigni G., Zabala S. (Venezuela - 2015). En su investigación “Diseño de interfaces en los medios visitados por los venezolanos ¿Los más usados son usables?”, en la cual se estudia y se determina si los sitios web de los medios de comunicación social más visitados por los internautas venezolanos cumplen con las Heurísticas de Nielsen y con la Guía para evaluación experta, de Márquez Correa, en cuanto a diseño de interfaces web de usuario. Para la selección de los medios de comunicación social con presencia en Internet que, de acuerdo a un ranking internacional de medición de uso de sitios web (Ranking de Alexa), están en los top 100 más visitados.





En este caso, se tomó el Ranking de Alexa ya que es una página de inteligencia y navegación en la Web, la cual recopila una extensa información sobre las páginas más visitadas por los internautas a través de su navegación en Internet, todo lo cual resulta en sus rankings ampliamente conocidos a nivel mundial.

Para realizar esta investigación se construyeron dos (2) instrumentos de evaluación con preguntas dicotómicas que responden a las Heurísticas de Nielsen y a la Guía para evaluación experta, de Márquez Correa. Luego, se creó una encuesta digital en Google Driver con la que los expertos evaluaron cada sitio web y respondieron a las preguntas del cuestionario. Finalmente, se tabularon todas las respuestas de los expertos para su análisis y para la generalización de los resultados.

La investigación concluyo que el uso garantiza un posicionamiento en la Web o en Internet, también es cierto que, si estos sitios web mejoraran sus aspectos de usabilidad, podrían también mejorar su posicionamiento en el Ranking de Alexa, o en otros rankings. Sin embargo, quizá esto contradiga la posición de, por ejemplo, el sitio web de “La Patilla”, medio de comunicación digital que ya ocupa el primer lugar del ranking, y que para nada podría estar interesado en reenfocar el diseño web de su espacio en Internet. No obstante, si lo que se persigue es ofrecer a los usuarios internautas lo mejor de los recursos informáticos, cada grupo de diseño web o web master debería estar considerando incorporar mejoras teóricas, prácticas y técnicamente comprobadas sobre los principios de usabilidad de aplicaciones informáticas, y, en este caso particular, de interfaces gráficas de sitios web para mejorar la experiencia de uso de los sitios y el disfrute de estos por parte del usuario internauta.

No acoplarse a los beneficios estudiados y recomendados por expertos para el diseño de interfaces gráficas de usuario puede hacer que el éxito y la vigencia de un sitio web sean efímeros en el tiempo. Por

tanto, se debe tener en cuenta para el diseño de páginas web que el uso no garantiza la usabilidad de las aplicaciones; sin embargo, la usabilidad sí garantiza el uso de las ellas. De modo que, si una empresa desea obtener ventajas competitivas, estar en línea con los avances tecnológicos y aprovechar todos los beneficios que estos pueden aportar al desenvolvimiento del negocio, es preciso comenzar a diseñar aplicaciones y sistemas de información que cumplan las Heurísticas de Nielsen y las partes de la Guía para evaluación experta, de Márquez Correa en cuanto al diseño de interfaces gráficas para la interacción humano-computador.

En el trabajo realizado por Hari Prasad N. y Ramesh G. (India – 2015), que lleva por título “Usability through Design and its Impact on Reusability”, tiene por objetivo determinar el impacto de la usabilidad en la reusabilidad cuando la aplicación de software ha sido desarrollada usando directrices de usabilidad para el desarrollo de la misma.

La usabilidad es " *el grado en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto específico de uso* ", hace referencia la ISO 9241 citado por Hari Prasad N. y Ramesh G. (India – 2015). Usabilidad generalmente conocido como " la calidad en uso ". Una aplicación de software, incluso con más servicios y funcionalidad es inútil, si poseen problemas de usabilidad.

En trabajos existentes definen directrices de usabilidad y artefactos de diseño para construir características específicas de usabilidad en las aplicaciones de software. La hipótesis de los trabajos existentes indican que la aplicación de las directrices de usabilidad, reduce el tiempo de desarrollo, reduce la dificultad percibida funcionalidad asociada a la facilidad de uso para los desarrolladores, y mejora la calidad de los diseños de software resultante. Sin embargo, estas investigaciones no tuvieron en cuenta el impacto de la utilización de las directrices de usabilidad a la reusabilidad. Es por ello que el

trabajo realizado por Hari Prasad N. y Ramesh G, se incrementa la facilidad de uso a través de las guías de diseño utilizando, así como evalúa su impacto en la capacidad de reutilización. Las directrices son aplicadas con dos modelos de directrices llamadas No Guidelines (NG) y Full Guidelines (FG), donde cada aplicación y comparar la facilidad de uso y calidad alcanzada en cada modo.

Los resultados obtenidos a partir de cada modo se evalúan y se comparan entre sí desde la perspectiva de la reutilización como una intención de encontrar impacto facilidad de uso en la reutilización de calcular el porcentaje de reutilización de código fuente de cada modo.

Este trabajo de investigación llegó a la conclusión de que la reutilización se reduce cuando aumenta la facilidad de uso, es decir no hay equilibrio entre las posibilidades de uso cuando las aplicaciones se desarrollan utilizando las directrices de usabilidad. Los ingenieros de software o académicos de investigación pueden encontrar el impacto de la usabilidad sobre otros atributos de calidad como el rendimiento, la seguridad, facilidad de mantenimiento, etc.

## **2.3. Bases teóricas científicas**

### **2.3.5. Ingeniería de software**

#### **2.3.1.1. Definición de software**

Según el autor(Pressman S., 2010, pág. 6), da como definición de software como programas aislados que procesan datos comerciales o técnicos, controlando sus funciones en tiempo real, facilitando las operaciones de negocio con el fin de resolver una necesidad específica del negocio.

(Loic Martinez, 2005, pág. 75), hace mención que un software normalmente se clasifica en programas de control, programas de procesos y programas de aplicación.

#### **2.3.1.2. Definición de Ingeniería de Software**

*“La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software*

*desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza” (Sommerville I. , 2005, pág. 22)*

Incluyendo actividades de ingeniería de requisitos, modelos de procesos y técnicas de estimación. (Daniel Ramos, 2015, pág. 77)

La ingeniería del software es *“la aplicación de herramientas, métodos, disciplinas para la producción y mantenimiento de soluciones automáticas a problemas del mundo real”*. (Loic Martinez, 2005, pág. 75)

### 2.3.1.3. Proceso de desarrollo de software

Un proceso es el control y gestión de un proyecto de software en donde se define que se está haciendo, quien lo está realizando, cuando se va a realizar y como lo va a realizar para poder alcanzar con los objetivos planteados. Un proceso sirve como guía para todos los involucrados durante todo el desarrollo de software, que son los usuarios, clientes y desarrolladores, etc. (Ivar, Grady, & James, 2000).

Jacobson nos menciona que un *“proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requerimientos del cliente”* (Ivar, Grady, & James, 2000, pág. 30).

(Laínez Fuentes, Noriega Martínez, Ramos, & Durango, 2015, pág. 13), hace mención que *“un proceso de desarrollo de software es una estructura utilizada para el desarrollo de un producto software”*.

Según Sommerville existen 4 actividades fundamentales de procesos que son comunes para todos los procesos de software (Sommerville I. , 2005, pág. 81):

- a. **Especificación del software:** Aquí el cliente y el desarrollador definen las funcionalidades y restricciones operacionales que el software debe cumplir.

- b. **Desarrollo del software:** Se diseña y construye el software de acuerdo a las especificaciones realizadas en la etapa anterior.
- c. **Validación del software:** Aquí se realiza una comparación entre las especificaciones realizadas en la primera etapa con el producto final con el fin de validar que el software cumpla con dichas especificaciones realizadas por el cliente.
- d. **Evolución del software:** Donde el software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado.

#### 2.3.1.4. Ciclo de vida de software

Según (Macconnel , 1996, pág. 145) define al ciclo de vida de software como *“un modelo prescriptivo de lo que pasaría entre la primera chispa y el último aliento”*.

Un modelo de ciclo de vida establece el orden en la cual se realizarán los prototipos, se diseñará, implementará, revisará, se probará y se la ejecución de otras actividades en un proyecto. “Establece el paso a paso de una actividad a otra”.

El definir el ciclo de vida en un proyecto es un punto crucial e importante a establecer en el plan del proyecto, dado a que la buena elección de este ayuda a la consecución del objetivo así como la definición del éxito del proyecto como cualquier otro punto relacionado a la planificación.

Según (SOMMERVILLE, 2005, pág. 62)el término Ciclo de vida del software implica *“la colección de actividades de trabajo, acciones y tareas que se realizan cuando va a crearse algún producto terminado. Cada una de las actividades, acciones y tareas se encuentra dentro de una estructura o modelo que define su relación tanto con el proceso como entre sí.”* Y la estructura de este o las actividades determinadas para la Ingeniería de software, son cinco actividades estructurales: comunicación, planeación, modelado, construcción y despliegue.

(Sánchez, Sicilia, & Rodriguez, 2012)Hace referencia que el ciclo de vida del software tiene como punto de partida la concepción del

producto software y se dice que el ciclo de vida concluye cuando este deja de usarse.

#### 2.3.1.4.1. Modelos de ciclo de vida

Los modelos del ciclo de vida especifican las actividades o fases que se deben desarrollar, así como el orden de ejecución de las mismas. Según (Macconel , 1996, pág. 146) la selección ineficaz de un modelo de ciclo de vida puede provocar retrasos en el flujo de trabajo, realizar trabajo repetitivo, innecesario y frustrante. Los mismos efectos se pueden producir si no se elige un modelo de ciclo de vida para llevar a cabo un proyecto de elaboración de un producto software. A Continuación veremos distintos modelos de ciclo de vida existentes:

##### 2.3.1.4.1. Modelo cascada

Uno de los primeros modelos de ciclo de vida. En este tipo de modelo el proyecto progresa a través de una secuencia ordenada de pasos partiendo del concepto inicial del software hasta la prueba del sistema. Bajo este modelo de desarrollo es indispensable la realización de una revisión final en cada etapa, con el fin de identificar deficiencias o generar observaciones, las cuales impiden el pasar a la siguiente etapa.

El modelo en cascada ayuda a localizar errores en las primeras etapas del proyecto, *“funciona correctamente cuando los requerimientos de calidad dominan sobre los requerimientos de coste y de planificación”*. Indica (Macconel , 1996, pág. 149).

Según (Sánchez, Sicilia, & Rodriguez, 2012) las ventajas del modelo en cascada son las siguientes:

- a) Asume la estabilidad de los requisitos durante el desarrollo, lo que en la mayoría de los casos resulta poco realista.
- b) Los límites entre las fases son demasiados rígidos, y esto hace que la revisión de los entregables e hitos entre las fases sea difícil, a menudo estas decisiones son tan complejas que obliga a paralizar el desarrollo.

c) Los errores de cada fase son percibidos en la fase de pruebas, cuando el código ya se ha realizado, lo cual para la subsanación de las mismas requiere mucho esfuerzo.

(Macconnel , 1996, pág. 150) Hace referencia que la desventaja de este modelo es la dificultad que se tiene para especificar claramente los requerimientos al iniciar con el proyecto, antes de que se realicen ningún trabajo de diseño y antes de escribir ningún código.

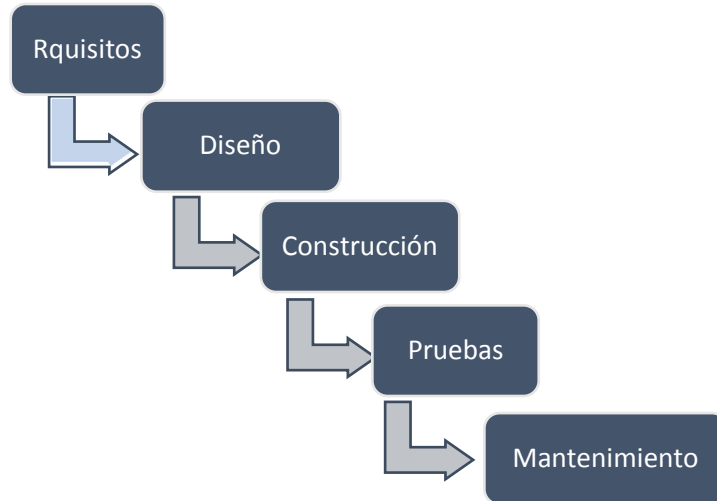


Figura 2: Modelo de ciclo de vida en cascada. Fuente: Ingeniería del software un enfoque desde la guía Swebok (2012), página 40.

**2.3.1.4.1. Modelo de ciclo de vida en “V”**

(Sánchez, Sicilia, & Rodriguez, 2012) Hace referencia que este modelo es una evolutivo del ciclo de vida en cascada, donde se hace énfasis en la verificación y validación.

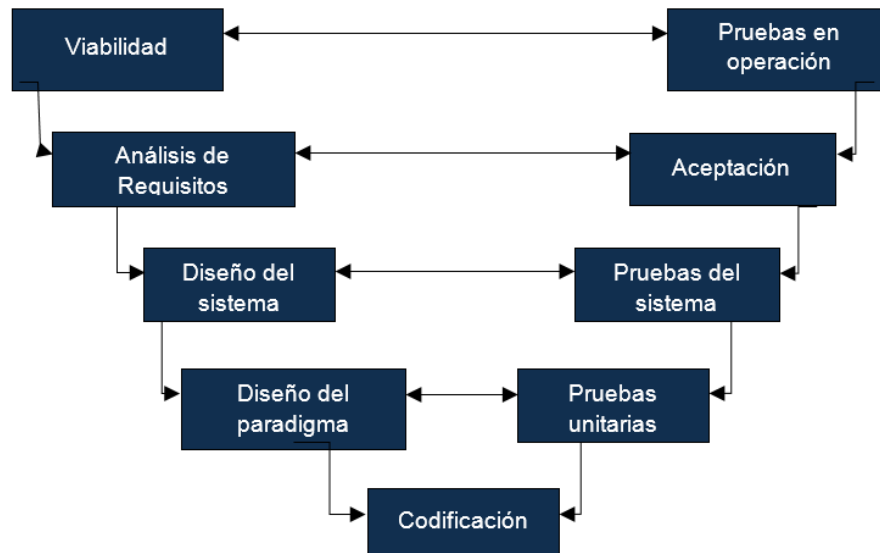


Figura 3: Modelo en “V”. Fuente: (Sánchez, Sicilia, & Rodriguez, 2012, pág. 41)

(Pressman S., 2010, pág. 34) Indica que en este modelo se puede apreciar la *“relación entre las acciones para el aseguramiento de la calidad y aquellas asociadas con la comunicación, modelado y construcción. A medida que el equipo de software avanza hacia abajo desde el lado izquierdo de la V, los requerimientos básicos del problema mejoran hacia representaciones técnicas cada vez más detalladas del problema y se su solución”*.

Subiendo a la ejecución de las actividades del lado derecho nos encontramos con las pruebas unitarias la cual corresponde al diseño detallado o de “bajo nivel”, aunque en otros casos las pruebas unitarias son consideradas como la prueba de la codificación. Continuando con las actividades del lado derecho indica la ejecución de las pruebas del sistema en la cual se pretende evaluar el diseño a alto nivel, las pruebas de aceptación, la cual evalúa que los requisitos que se capturaron y se establecieron están reflejadas fielmente de acuerdo a las necesidades indicadas por el usuario, (Sánchez, Sicilia, & Rodríguez, 2012).

Este modelo ha sido elegido como ciclo de vida complementaria a las metodologías propuestas para el desarrollo de los casos de estudio de este proyecto de investigación, como un método complementario de la metodología UWE, dado a que este modelo permite el seguimiento de nuestro proyecto caso de estudio bajo la perspectiva constante de la validación, en actividades de control y verificación, del lado del desarrollo.

#### **2.3.1.4.1. Modelo espiral**

(SOMMERVILLE, 2005, pág. 68) Indica que este modelo fue propuesto por Boehm en 1988. Representa el proceso del software como una espiral. Cada ciclo en la espiral representa una fase del proceso del software, donde el ciclo más interno podría referirse a la viabilidad del sistema, el siguiente en la definición de requerimientos, en el diseño, etc, La representación gráfica de este modelo esta partido por cuatro cuadrantes, las cuales especifican una actividad diferente:



- a. Definición de objetivos: Se definen los objetivos específicos del proyecto, así como también se identifican las restricciones del proceso y el producto, se lleva a cabo la elaboración de un plan de gestión, así como también se identifican los riesgos y se elaboran planes hacer frente a estas contingencias.
- b. Evaluación y reducción de riesgos: en este punto se realiza el análisis detallado de los riesgos del proyecto identificado, para poder definir los pasos a seguir para la reducción de estos riesgos.
- c. Desarrollo y validación. Para lo cual es necesaria la construcción de prototipos evolutivos.
- d. Planificación. Revisar el estado de desarrollo del proyecto, para así poder tomar la decisión de continuar con el siguiente ciclo en espiral. Si se decide continuar, se desarrollan los planes para hacer frente a la siguiente fase del proyecto.

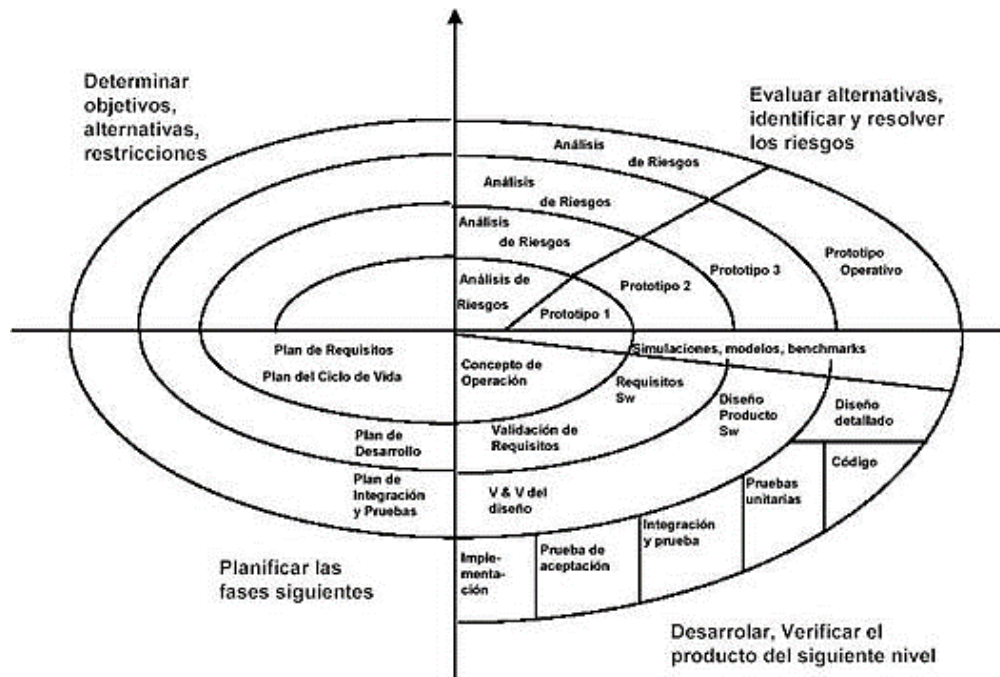


Figura 4: Modelo en espiral de Boehn para el proceso del software. Fuente: (Pressman S., 2010).

En este modelo la gestión de los riesgos es una tarea muy primordial.

(Sánchez, Sicilia, & Rodríguez, 2012), hace mención que este modelo reúne las mejores características del uso de prototipos y que el número de vueltas de la espiral no queda determinado por el modelo. Uno de los puntos fuertes es que la aplicación se desarrolla de forma progresiva, con la colaboración de los usuarios, y por lo tanto la aceptación del software es progresiva.

#### **2.3.1.4.1. Modelo de proceso incremental**

(Pressman S., 2010, pág. 35), hace referencia que recomendable manejar procesos incrementales cuando se requiera entregar rápidamente cierta funcionalidad del software al usuario y aumentar las funcionalidades conforme se haga entrega de nuevas versiones del software. Ante la situación descrita con anterioridad recomendable elegir un modelo de procesos que está diseñado para producir el software de forma incremental.

Cuando se opta por este modelo es necesario definir el proceso principal a automatizar con el producto software, dado que mientras el usuario pone a prueba el producto fundamental, se encontraran algunas observaciones que requieran hacer mejoras, y subsanadas en el siguiente incremento y presentación del producto, es fundamental también realizar un plan para los incrementos siguientes, teniendo en consideración el levantamiento de las incidencias encontradas, y también el añadir nueva funcionalidad al sistema en base a los requerimientos solicitados por el usuario. Cada incremento tiene que tener como producto final un entregable operable. El proceso descrito anteriormente se repite hasta terminar el producto final.

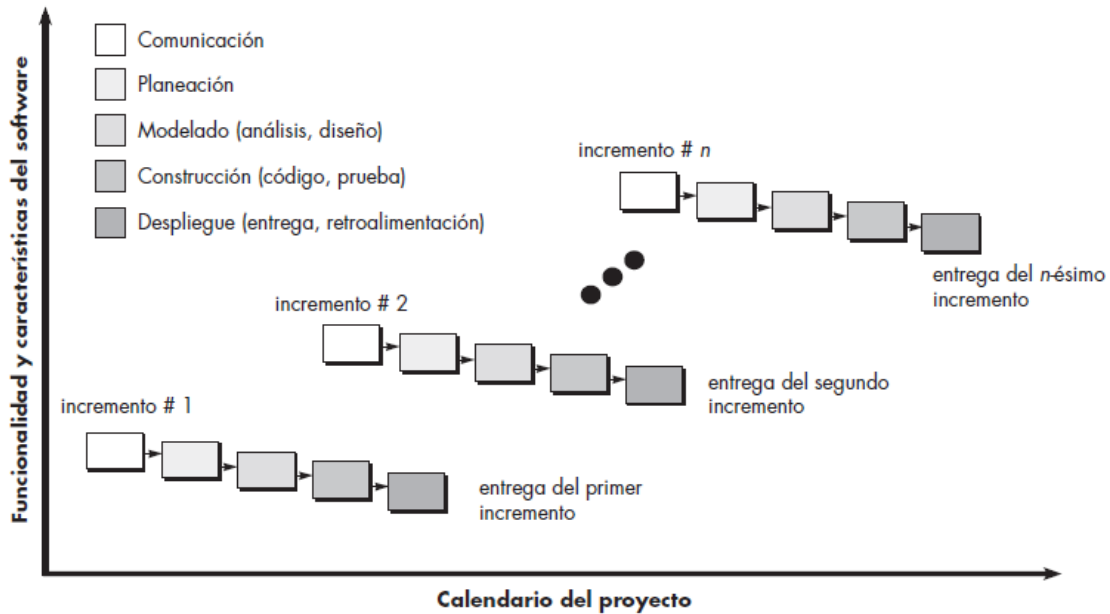


Figura 5: Modelo incremental. Fuente: (Pressman S., 2010).

Según (SOMMERVILLE, 2005, pág. 8) “la mayor parte de los modelos de procesos del desarrollo de software se basan en uno de los tres modelos generales o paradigmas desarrollo de software” como los que veremos a continuación, pero para ello es necesario definir y entender que es un paradigma.

## 2.3.5. Paradigma de desarrollo de software

### 2.3.2.1. Definición

Según ( H. Kaisler, 2005, pág. 2) los paradigmas son el resultado emergente de un proceso social en donde las personas desarrolla nuevas ideas para crear principios y prácticas, estas están incorporadas dentro de esas ideas.

“Laingeniería de software está compuesta por una serie de modelos que abarcan los métodos, las herramientas y los procedimientos. Estos modelos se denominan frecuentemente paradigmas de la ingeniería del software y la elección de un paradigma se realiza básicamente de acuerdo al tipo del proyecto y de la aplicación, los controles y las entregas a realiza.”

## 2.3.2.2. Tipos de Paradigmas

### 2.3.2.2.1. Lineal y secuencial

(Garrido Carrillo, 2005, pág. 348) Hace mención que este paradigma lleva acabo las actividades de desarrollo de la solución software, análisis, diseño, implementación y prueba, de forma secuencial, cerrando cada una de ellas una antes de comenzar la siguiente. En la práctica, es muy común retroceder a la actividad anterior, aun que esta debe ser evitada dado a que causa un elevado coste.

### 2.3.2.2.2. Evolutivos

En este paradigma se mezclan las distintas actividades, de forma que se obtengan versiones incrementales del sistema. Las distintas versiones generadas se van refinando hasta obtener el resultado deseado. (Garrido Carrillo, 2005, pág. 348)

(Cortés Morales, 2006, pág. 23) , menciona que el proceso de construcción del software es evolutivo, cuya actividad de partida es la planificación, posteriormente se realiza la evaluación de los riesgos existentes. Si el riesgo de continuación del proyecto es muy grande, este se detendrá. Si no se sigue con la actividad de ingeniería, que construirá un producto sujeto a la posterior evaluación del cliente. De las observaciones de estos últimos se hará una nueva planificación, se estudiarán los nuevos riesgos, y se realizara el siguiente producto. Una vez que el cliente de por aprobado el producto, se planificará la puesta en producción, se evaluarán sus riesgos y se concluirá, con la puesta en producción de la aplicación. Este paradigma es muy utilizado en aquellos proyectos en los cuales la experiencia en la aplicación y la tecnología resulta muy nueva para el equipo de trabajo, y en aquellos proyectos donde la complejidad es considerable.

*“Este es un paradigma que requiere una gran habilidad para la identificación de riesgos. Si un riesgo no se detecta, el conjunto del proyecto puede verse afectado.”* Y puede ser difícil de convencer a los

clientes del enfoque evolutivo dado a que en este tipo de paradigma no es muy claro el visualizar la finalización del proyecto.

### **2.3.2.2.3. Prototipado**

Este tipo de paradigma hace énfasis en la etapa de requisitos para poder definir y establecer el alcance del producto software a construir, para así poder realizar los modelos de las interfaces de aplicación que serán presentadas al cliente de tal forma que se puedan ir refinando las ideas. Las Observaciones deben ser captadas por el programador y modificar el modelo de acuerdo a las observaciones. El proceso se repite hasta alcanzar a cubrir todos los requerimientos del producto software, pasando luego a construir la aplicación, aprovechando muchas veces la programación efectuada en los prototipos. (Cortés Morales, 2006, pág. 23)

### **2.3.2.2.4. Ensamblaje de componentes**

Considera la existencia de algunas partes necesarias para el sistema, incluso la compra a otras empresas, de forma que solo es necesaria su integración en el sistema que se desarrolla, mientras que otras partes también se desarrollan como componentes, que se podrán reusar en otros sistemas, como lo indica (Garrido Carrillo, 2005, pág. 349)

## **2.3.5. Desarrollo de software dirigido por modelos (MDD)**

### **2.3.3.1. Introducción**

(Pons, Giandini, & Pérez, 2010, págs. 24, 25), hace referencia que “DeMarco(1979), Structured Analysis and SystemSpecification” destaco que la construcción de un sistema debe ser precedida por la construcción de un modelo, y este modelo del sistema debe conceptualizar el dominio del problema y la solución que se ha planteado. Este modelo debe representar al mundo real: identificándolo, clasificándolo y abstrayendo elementos que componen el problema y estos estar organizados.

La abstracción es una técnica que nos permite ocultar lo que irrelevante, con el fin de convertir o minimizar las características de un sistema completo a algo que sea comprensible y manejable. Cuando se trata de software, es sumamente útil abstraerse de los detalles tecnológicos de implementación y tratar con los conceptos del dominio de la forma más directa posible, dado a que el modelo que se realice servirá como herramienta de comunicación y negociación entre usuarios, analistas y desarrolladores, es por ello que el modelo debe ocultar o minimizar los aspectos relacionados con la tecnología de implementación, para que pueda ser entendidos por todos los involucrados.

*“Actualmente casi todos los métodos de desarrollo de software utilizan modelos. Lo que varía de un método a otro es la clase de modelos que deben construirse, la forma de representarlos y manipularlos”* hace referencia (Pons, Giandini, & Pérez, 2010, págs. 24, 25), Existen dos tendencias principales de los modelos:

- a) Modelos matemáticos: estos modelos nos plasmar el problema y dar solución a este de una forma más rigurosa y precisa, utilizando un lenguaje basado en las matemáticas. La ventaja de esta forma de modelar lleva a generar sistemas robustos y consistentes, en los cuales es posible tanto la validación de la especificación por parte del usuario (detección y corrección temprana de defectos), como la verificación del software. No tiene mucha aplicación en la industria de desarrollo de software dado a que es una forma muy compleja, para su entendimiento ambas partes deben ser expertos en el lenguaje matemático.
- b) Modelos diagramáticos: son una propuesta de plasmar la realidad problemática y la solución a esta a partir de modelos basados en graficas las cuales son más fáciles de utilizar y comprender que los métodos formales. Un ejemplo a representativo y a utilizar en es UML, que transmite un significado intuitivo, este tipo de

lenguaje diagramáticos son los más aceptados por los desarrolladores de software.

### 2.3.3.2. Paradigma MDD

Según (Pons, Giandini, & Pérez, 2010, págs. 28-20), el MDD (Model Driven software Development) es un paradigma de desarrollo software, que provee mejoras en el proceso de construcción de un software basándose en un proceso guiado por modelos y soportado por potentes herramientas. *“El adjetivo dirigido (driven) en MDD, a diferencia de “basado” (based), enfatiza que este paradigma asigna a los modelos un rol central y activo: son al menos tan importantes como el código fuente”*. Los modelos se van generando y se van haciendo más concretos a través de múltiples transformaciones y/o refinamientos, hasta que esto lleguen por medio de una transformación lleguen a estar representados por códigos. A continuación mencionaremos algunos puntos claves de la iniciativa MDD:

- a. Hacer uso de una mayor abstracción en la especificación tanto del problema como la solución a la realidad encontrada. Para ello se implementan modelos del dominio, que conceptualizan el problema y la solución lo más entendible posible, ocultando o minimizando aspectos relacionados con la tecnología de implementación, de tal forma que sean estos modelos resulten lo más amigable para el usuario y sean fáciles de transmitir. El modelo permite reducir el impacto que la de evolución tecnología que se impone sobre el desarrollo de aplicaciones, permitiendo que el mismo modelo abstracto se materialice en múltiples plataformas de software.
- b. La automatización del computador para soportar el análisis, el diseño y la ejecución, aumenta la productividad, calidad, y confianza en los mismos. En este paradigma de desarrollo de software es fundamental el utilizar a las computadoras para transformar modelos expresados mediante conceptos de alto

nivel, a programas informáticos que puedan ser desplegados sobre una plataforma tecnológica específica.

- c. El uso de estándares industriales es un buen punto clave dado a que facilita la comunicación, la interacción entre diferentes herramientas complementarias tecnológicas.

Con este paradigma los modelos pasan de ser herramientas contemplativas, de solo interpretación, para convertirse en entidades productivas de las cuales se deriva la implementación automática.

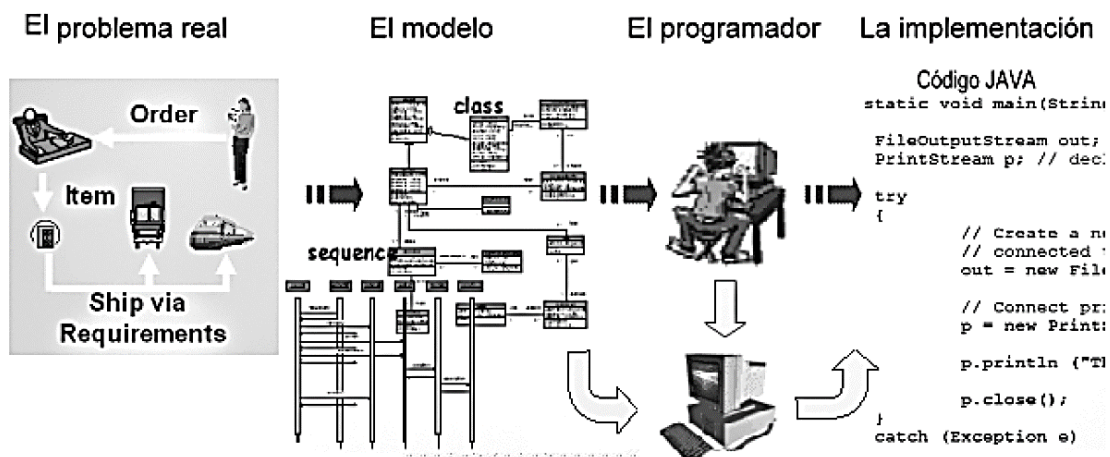


Figura 6: Desarrollo de software dirigido por modelos. Fuente: (Pons, Giandini, & Pérez, 2010).

### 2.3.3.3. El ciclo de vida dirigido por modelos

(Pons, Giandini, & Pérez, 2010, pág. 26) Manifiesta que el ciclo de vida dirigido por modelos, considera un proceso que incluye seis fases:

- a. La conceptualización la determinación de los requisitos del usuario.
- b. Análisis y descripción funcional
- c. Diseño
- d. Codificación
- e. Testeo
- f. Emplazamiento

A diferencia de otros modelos de ciclo de vida este se diferencia por el tipo de artefactos que se crean durante el proceso de desarrollo.



Los artefactos son modelos formales, que pueden ser comprendidos por una computadora.

En la figura Nº 7 se observan las actividades a realizar para el desarrollo de software y los artefactos que se consideran realizar y son entradas para las siguientes actividades.

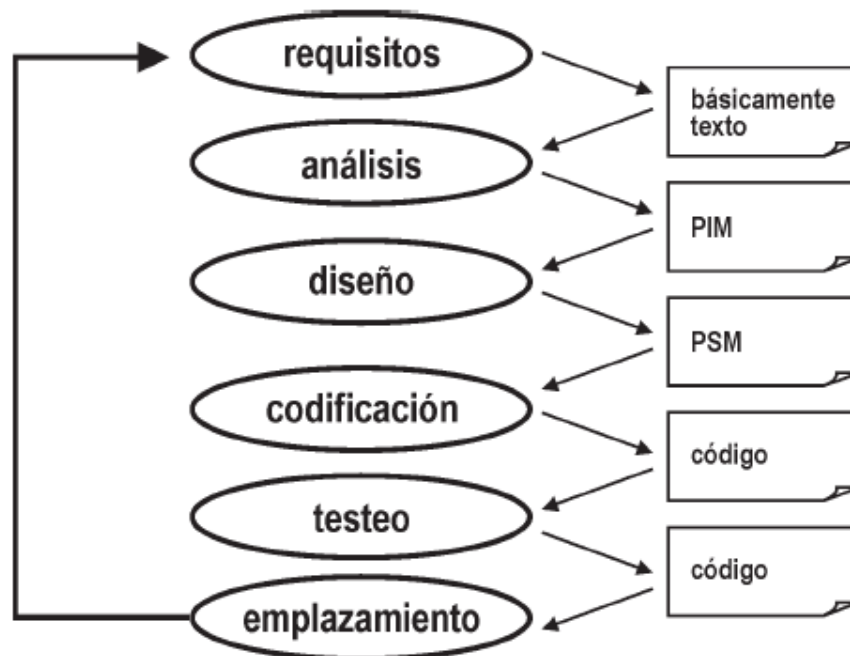


Figura 7: Ciclo de vida del desarrollo de software dirigido por modelos. Fuente: (Pons, Giandini, & Pérez, 2010, pág. 31)

El paradigma MDD identifica los siguientes tipos de modelos:

- a. Modelos con alto nivel de abstracción independientes de cualquier metodología computacional, llamados CIMs (Computational Independent Model) o modelo del dominio y en su construcción se utiliza un vocabulario que resulta familiar para los expertos en el dominio en cuestión, para lo cual no es necesario de tener un conocimiento técnico para entender estos modelos. El CIM juega un papel muy importante en reducir la brecha entre los expertos en el dominio y sus requisitos por un lado, y los expertos en diseñar y construir artefactos desoftware por el otro.
- b. Modelos independientes de cualquier tecnología de implementación llamados PIMs (Platform Independent Model).

Cuando se realiza el modelo del sistema bajo este paradigma se debe de hacer pensando como este soportará mejor al negocio, dejando de lado aspectos relacionados a la tecnología que se va a hacer uso (lenguaje de programación, hardware, sistema operativo, etc.), de tal manera que un PIM pueda ser implementado sobre diferentes plataformas.

- c. Modelos que especifican el sistema en términos de construcciones de implementación disponibles en alguna tecnología específica, conocidos como PSMs (Platform Specific Model), Cada PSM representa la proyección del PIM en una plataforma específica. Un PIM puede generar múltiples PSMs, cada uno para una tecnología en particular. Generalmente, los PSMs deben colaborar entre sí para una solución completa y consistente. Por ejemplo, un PSM para Java contiene términos como clase, interfaz, etc. Un PSM para una base de datos relacional contiene términos como tabla, columna, clave foránea, etc.
- d. Modelos que representan el código fuente en sí mismo, identificados como PIMs (Implementation Model). El paso final en el desarrollo es la transformación de cada PSM a código fuente. Ya que el PSM está orientado al dominio tecnológico específico.

En el proceso de desarrollo de software tradicional, las transformaciones de modelo a modelo, o de modelo a código son hechas mayormente con intervención humana. Muchas herramientas pueden generar código a partir de modelos, pero generalmente no van más allá de la generación de algún esqueleto de código que luego se debe completar manualmente. En el paradigma MDD las transformaciones del PSM a código son siempre ejecutas por herramientas que automatizan este proceso.

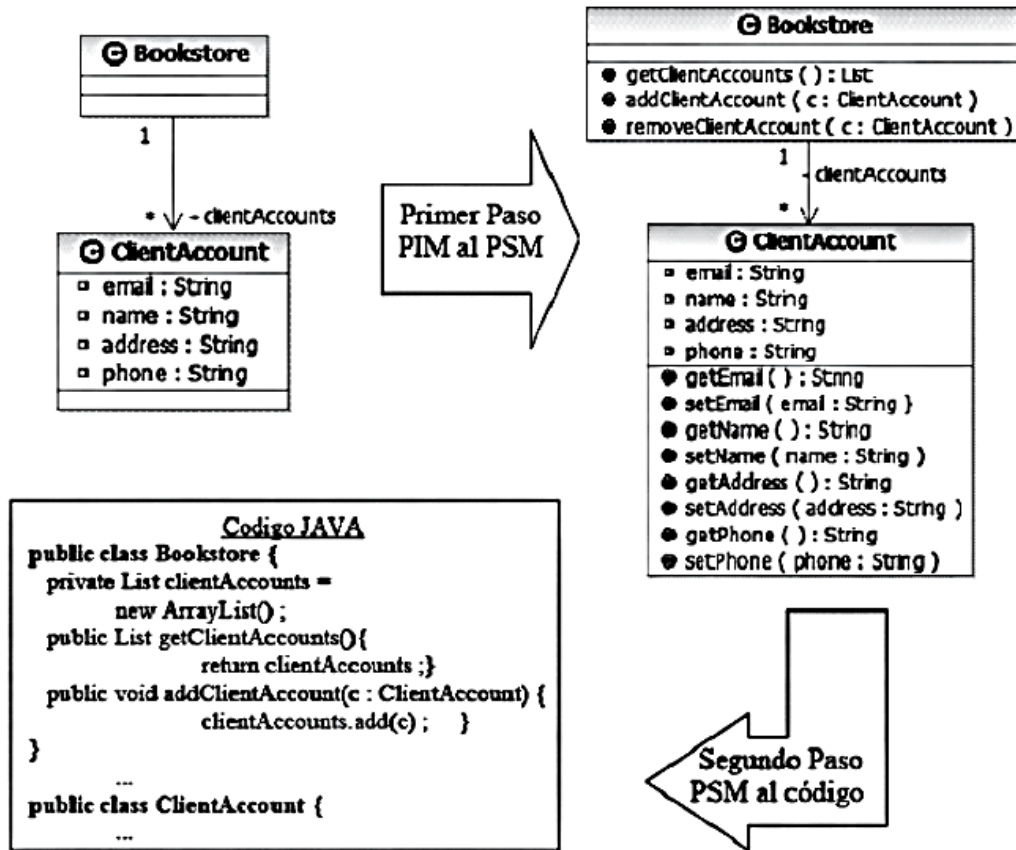


Figura 8: Ejemplo de transformación de PIM a PSM y de PSM a código. Fuente:(Pons, Giandini, & Pérez, 2010, pág. 54)

#### 2.3.3.4. Beneficios del ciclo de vida por modelos

Para (Pons, Giandini, & Pérez, 2010, págs. 32,33) MDD es la evolución natural de la ingeniería de software basada en modelos la cual es complementada con las transformaciones automáticas entre modelos. A continuación veremos algunas de las ventajas del uso de este paradigma:

- Incrementos en la productividad: reduce el costo de desarrollo de software mediante la generación automática de código y otros artefactos a partir de los modelos, lo cual incrementa la productividad de los desarrolladores. El costo de realizar estas transformaciones se justifica mediante el re-uso de dichas transformaciones.
- Adaptación a los cambios tecnológicos: MDD propone una arquitectura fácil de mantener donde los cambios se implementan rápida y consistentemente, habilitando una migración eficiente de los componentes hacia las nuevas tecnologías. Realizar modelos

de alto nivel donde se tome en consideración los detalles de implementación, facilitan el uso de estas para implementar la solución en otras plataformas tecnológicas. Los cambios se realizan solo modificando la transformación del PIM al PSM.

- c. Adaptación a los cambios en los requisitos: Adaptarse a los cambios de requerimientos es un punto esencial, es por ello que cuando usamos un proceso MDD, agregar, modificar una funcionalidad de negocios ya no es una tarea complicada, dado a que necesitamos desarrollar el modelo especificado que soporta a la nueva solución. El resto de la información necesaria para generar los artefactos de implementación ya han sido capturadas en las transformaciones y pueden ser reutilizadas.
- d. Consistencia: la codificación y el diseño son tareas propensas a errores, los cuales son disminuidos con la automatización MDD, dado que favorece a la generación consistente de los artefactos.
- e. Re-uso: en MDD se invierte en el desarrollo de modelos y transformaciones. El re-uso de artefactos ya probados incrementa la confianza en el desarrollo de nuevas funcionalidades y reduce los riesgos ya que los temas técnicos han sido previamente resueltos.
- f. Mejoras en la comunicación con los usuarios: los modelos omiten detalles de implementación que no son relevantes para entender el comportamiento lógico del sistema, permitiendo una comunicación más fácil con el usuario y el lenguaje en el cual se expresa la solución.
- g. Mejoras en la comunicación entre los desarrolladores: los modelos facilitan el entendimiento del sistema por parte de los distintos desarrolladores, lo que conlleva a tener un intercambio de opiniones más productivas. Además el hecho que los modelos no sean parte solo de la documentación sino en si del sistema, hacen los modelos siempre permanezca actualizados y confiables.
- h. Los modelos son productos de larga duración: Los modelos de alto nivel son independientes de la plataforma, lo que hace que estos

sean de perdurables en el tiempo y la tecnología, solo sufren cambios cuando se añaden nuevos requerimientos del negocio.

Este paradigma de desarrollo de software ofrece la posibilidad de transformar modelos escritos en distintos lenguajes, como es el caso en que modelemos en UML y se requiera hacer una transformación a otro modelo con notación Entidad- Relación. Generalmente UMNL es el lenguaje elegido para definir los modelos en MDD.

### 2.3.3.5. El rol de UML en MDD

(Schmuller, 2001, pág. 20) El lenguaje unificado de Modelado (UML) es una herramienta que sirve para capturar la idea de un sistema para luego comunicarla a todos los involucrados del proyecto, este es entendido por usuarios especialistas como usuarios no tan especialistas dado a que UML se basa en un conjunto de símbolos y diagramas. Y cada uno de estos diagramas tiene fines distintos dentro del proceso de desarrollo.

(Pons, Giandini, & Pérez, 2010, pág. 50), UML es un estándar abierto y es el estándar de facto para el modelado de software. UML es un lenguaje de propósito general que puede ser aplicado a diversos casos de desarrollo de aplicaciones.

El uso de UML como lenguaje en MDD tiene los siguientes beneficios:

- a. El uso de perfiles de UML para MDD nos permite aprovechar la experiencia que se ha ganado en el desarrollo de este lenguaje. Esto significa que podemos proveer un ambiente de modelado personalizado sin afrontar el costo de diseñarlo e implementarlo desde cero.
- b. UML es un estándar abierto y el estándar de facto para el modelado de software en la industria.
- c. UML ha demostrado ser durable, su primera versión apareció en 1995.
- d. El éxito de UML ha propiciado la disponibilidad de muchos libros y cursos de muy buena calidad.

- e. La mayoría de los estudiantes de las carreras relacionadas con la ingeniería de software han aprendido algo de UML en la universidad.
- f. Existen muchas herramientas maduras que soportan UML.
- g. Muchas organizaciones necesitan desarrollar varios tipos diferentes de software. Si podemos usar una solución de modelado en común, configurada de forma apropiada para describir cada uno de esos dominios de software, entonces será más fácil lidiar con su posterior integración.



Figura 99: Transformación de un modelo PIM escrito en UML a un modelo de implementación escrito en Java.(Pons, Giandini, & Pérez, 2010)

Los paradigmas están inmersos en las metodologías de desarrollo, y el paradigma Desarrollo dirigido por Modelos no es ajeno a esto, dado a existen metodologías de desarrollo web basado en este. Según (Moreno, Marciszack, 2014, pág. 5), como ejemplo se pueden citar: OO-Method, WebRatio, OOHDM, UWE. Así como también la metodología de desarrollo web NDT, que según (Maria J. Escalona ; University of Seville, Seville ; Gustavo Aragón, 2008, pág. 377), es una metodología web basada en el paradigma MDD, que está siendo aplicado en proyectos reales.

### 2.3.5. Metodologías de desarrollo de aplicaciones Web

#### 2.3.4.1. Aplicaciones Web

En los primeros días de la World Wide Web, los sitios web consistían en un conjunto de archivos de Hipertexto vinculados que presentaban información utilizando textos y gráficos limitados. A medida que pasaba el tiempo Hypertext Markup Language (HTML) fue ampliado

por las herramientas y tecnologías como XML (Lenguaje de marcas Extensible) y java, que permitió a los Ingenieros Web poder proporcionar del lado del Cliente y del lado del servidor, capacidad computacional junto con el contenido, naciendo los sistemas y aplicaciones basados en la Web. Hoy en día, WebApps han convertido en herramientas informáticas sofisticadas que no sólo proporcionan una funcionalidad independiente para el usuario final, sino que también se han integrado con las bases de datos y aplicaciones corporativas y gubernamentales, (Presssman & Lowe, 2009, pág. 2)

Según (Brandon, 2008, pág. 1) las aplicaciones y sistemas basados en la Web ofrecen un complejo conjunto de funcionalidades a un gran y diverso número de grupos de usuarios, y en la actualidad se ha creado una gran dependencia de la web, es por ello que el rendimiento, la fiabilidad y la calidad se ha convertido en características vitales para los sistemas y aplicaciones Web. Como resultado, el desarrollo de aplicaciones Web se ha vuelto más complejo que el desarrollo de software tradicional. Es por ello que para construir con éxito y mantener sistemas grandes y complejos basados en la Web y las aplicaciones, los desarrolladores Web necesitan adoptar un proceso de desarrollo disciplinado y una sólida metodología, para el éxito del desarrollo web.

#### 2.3.4.2. Atributos de las aplicaciones basadas en la Web

(Pressman S., 2010, pág. 522) Hace referencia que los sistemas y las aplicaciones basadas en la web son muy diferentes de las categorías de software informático, dado a que los sistemas basados en la web *“implican una mezcla de publicaciones impresa y desarrollo de software, marketing e informática, de comunicaciones internas y relaciones externas, y de arte y tecnología”*. A continuación mencionaremos atributos que en su mayoría son encontrados en las WebApps.

- a. Intensivas de Red: las WebApps residen en una red y debe dar servicio a las necesidades de una comunidad diversa de clientes.

- b. Controlada por el contenido: En muchos casos, la función primaria de una WebApp es utilizar hipermedia para presentar al usuario el contenido de textos, gráficos, sonido y vídeo.
- c. Evolución continua: A diferencia del software de aplicaciones convencional, que evoluciona con una serie de versiones planificadas y cronológicamente espaciadas, las aplicaciones Web están en constante evolución. No es inusual que algunas WebApps se actualicen cada hora, en especial su contenido.
- d. Inmediatez: Las aplicaciones basadas en Web pueden ser comercializadas en cuestión de días o semanas. Los métodos de planificación, análisis, diseños, implementación y comprobación seleccionada por los desarrolladores deben tener la inmediatez para crear el producto en un corto tiempo.
- e. Seguridad: Dado que las WebApps están disponibles a través del acceso por red, llegando a miles y miles de usuarios es por ello que se deberá de implementar fuertes medidas de seguridad para proteger el contenido confidencial u proporcionar formas seguras de transmisión de datos.
- f. Estética: el atractivo de una WebApp, es su apariencia e interacción, es por ello que la estética define mucho el éxito del producto software.

#### 2.3.4.3. Atributos de la calidad

Según (Pressman S., 2010, pág. 523), hace mención que al igual que los software tradiciones en los sistemas y aplicaciones Web, la precepción de “lo bueno” por parte del usuario, es muy importante en la aceptación o no aceptación del resultante de estos sistemas basados en la web, y esta percepción podría ser más importante que cualquier discusión técnica sobre la calidad de este.

Todas las características generales de la calidad de software se aplican también a las aplicaciones y sistemas basados en la Web. Sin embargo las características más relevantes para este tipo de sistemas y aplicaciones son la usabilidad, fiabilidad, eficiencia, capacidad de



entendimiento, estas proporcionan una base muy útil para para evaluar la calidad de los sistemas basados en Web.

Ólsina (1999) citado en (Pressman S., 2010, pág. 524) hace mención que se ha preparado un Árbol de requisitos de calidad, que identifican un conjunto de atributos que conducen a los sistemas y aplicaciones basados en la web de alta calidad.



Figura 10: Árbol de requisitos de calidad Fuente: (Pressman S., 2010).

#### 2.3.4.4. Metodologías de desarrollo Web

##### 2.3.4.4.1. NDT

Según (Del Valle Rojo, S. 2012), la Metodología Navigational Development Techniques, fue propuesta por Escalona, Torres, Mejias, como sus autores lo indican, es una metodología centrada en la primera fase del ciclo de vida, su proceso de desarrollo se enfoca en la ingeniería de requerimientos, para lo cual presenta actividades y técnicas para elicitar, analizar, especificar y validar los requerimientos de las aplicaciones Web.

##### 2.3.4.4.1. Características

1. El proceso no contempla las fases de implementación o mantenimiento del ciclo de vida de desarrollo de aplicaciones Web, está centrado en la fase de definición de requerimientos.
2. El proceso de definición de requerimientos se basa los diferentes roles de usuarios que pueden aparecen en el sistema.
3. Es un enfoque orientado al proceso debido a que para cada actividad describe concretamente: los pasos a seguir para obtener los requerimientos y en base a ellos generar los modelos.
4. Es un enfoque orientado a la técnica, debido a que describe las técnicas que se deben aplicar, la forma de aplicarlas y los resultados que debe ser obtenido.
5. Es un enfoque orientado al resultado, puesto que NDT describe los modelos de estructura y nomenclatura para los resultados obtenidos tras la aplicación de cada técnica y los resultados finales del proceso que son: la especificación de requerimientos, la especificación de modelos de análisis y los prototipos de navegación.

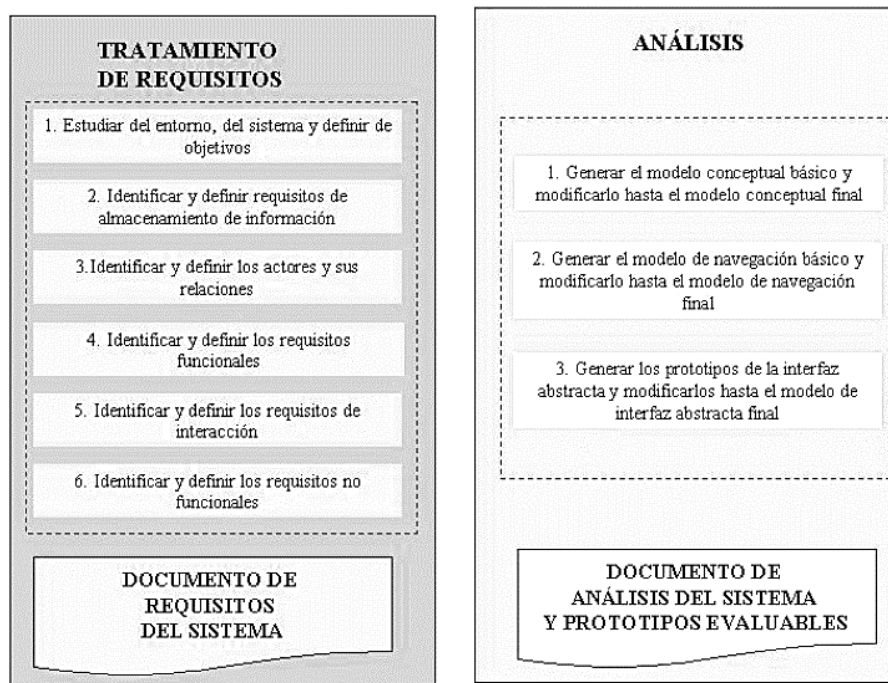


Figura 11: Diagrama con el proceso de desarrollo de NDT. (Del Valle Rojo, S. 2012),

6. Propone la validación de las especificaciones y modelos conceptuales obtenidos.
7. Ofrece una guía sistemática para el tratamiento de la navegación.

8. NDT no propone lenguajes de modelado, sino que utiliza lenguajes de modelado estándares como UML y UWE definido en otro enfoque metodológico.
9. Define un proceso iterativo para generar la especificación de requerimientos y especificación de modelos de análisis, dado que estos modelos son estándares, pueden ser tomados por otra metodología basada en esos estándares y continuar el ciclo de vida de desarrollo.
10. Propone 2 fases: Captura de Requerimientos la cual se divide en 7 subfases y por otro lado la fase de Análisis que comprende 3 subfases.
11. Es soportado por una herramienta CASE NDT-Suite.

#### 2.3.4.4.1. Fases del proceso NDT

- I. *Fase de Captura de requerimientos: Según* (Del Valle Rojo, S. 2012) el objetivo de esta fase es obtener como resultado el Documento de especificación de requerimientos. LA NDT clasifica a los requerimientos en:
  - a. Requisitos de almacenamiento de información, estos definen qué información se va a manejar en el sistema y cómo se relacionan entre sí. NDT permite también definir nuevas naturalezas de datos que se vayan a utilizar en el sistema.
  - b. Requisitos de actores, define quienes interactuarán con el sistema y las relaciones que se pueden producir entre ellos.
  - c. Requisitos funcionales, define las funcionalidades que se incluirán en el sistema.
  - d. Requisitos de interacción, define como se muestra la información, como se podrá navegar en el sistema y los criterios de recuperación que se ofrece.
  - e. Requisitos no funcionales

Según Del Valle Rojo, S. (2012), el proceso comienza definiendo los objetivos del sistema para lo cual plantea las siguientes actividades:

- a. Obtener información sobre el entorno y definir objetivos, las actividades principales consisten en: Obtener información sobre el

dominio de problema. Para ello es importante preparar y realizar reuniones y entrevistas. Identificar y definir los objetivos.

- b. Identificar y definir los requerimientos de almacenamiento de información: se propone realizar las actividades de identificación y definición de los requerimientos de almacenamiento de información e Identificar y definir las nuevas naturalezas.
- c. Identificar y definir los actores. Para ello propone un patrón para definición de actores básicos, diagramas para la representación de los actores generalizados, matriz para la definición de los mismos y derivados.
- d. Identificar y definir los requerimientos Funcionales, para ello se propone diseñar los diagramas de casos de uso y describir los casos de uso.
- e. Identificar y definir los requerimientos de interacción, para ello es necesario realizar la identificación y definición de las fases, así como también identificar y definir los prototipos de visualización. Mediante las técnicas como Bounded natural lenguaje, un patrón para definir las fases y un patrón para la definición de prototipos de visualización.
- f. Identificar y definir los Requerimientos No funcionales, en esta actividad se trata de identificar y describir a los requerimientos de comunicación, fiabilidad, de entorno de desarrollo, de portabilidad.
- g. Validar los requerimientos, para ello se propone técnicas como son: revisiones, auditorias, glosarios, tesauros y ontologías y matriz de trazabilidad. Si en esta actividad se detentan errores, se volverá a la actividad en la que se hayan detectado los mismos para corregirlos.
- h. Generar el documento de requerimientos del sistema: NDT propone una estructura que debe de tener este documento, el cual es la base para la fase de análisis.

Cabe mencionar que esta metodología no especifica alguna técnica para Elicitación/captura de requerimientos.

- II. *Fase de captura de Análisis*: El objetivo de esta fase es obtener como resultado el documento de Análisis del sistema y los prototipos evaluables, por ende se plantean las siguientes actividades:
- a. Realizar el Modelo Conceptual, para ello se propone técnicas como son los diagramas de clases, patrón para la definición de clases asociaciones, paquetes, proceso de generación y revisión de modelo básico.
  - b. Realizar el Modelo de Navegación, actividad que consiste en dos tareas Realizar el Modelo de Navegación Básico y la Realizar el Modelo de Navegación Final. Las técnicas propuestas son diagrama de navegación de UWE, patrón para la definición de nodos, enlaces, menús, índices, queries, proceso de generación de modelo básico, proceso de revisión de modelo básico algoritmos de Kruskal, y algoritmo de Warshall.
  - c. Realizar y validar el conjunto de prototipos, para ello es necesario la generación de los mismos, revisión y auditorías.
  - d. Generar el documento de análisis del sistema: NDT propone la estructura que debe tener este documento; la cual es similar a la del documento de requerimientos. En este documento se incluyen y describen los modelos generados durante la fase de análisis mencionados anteriormente.

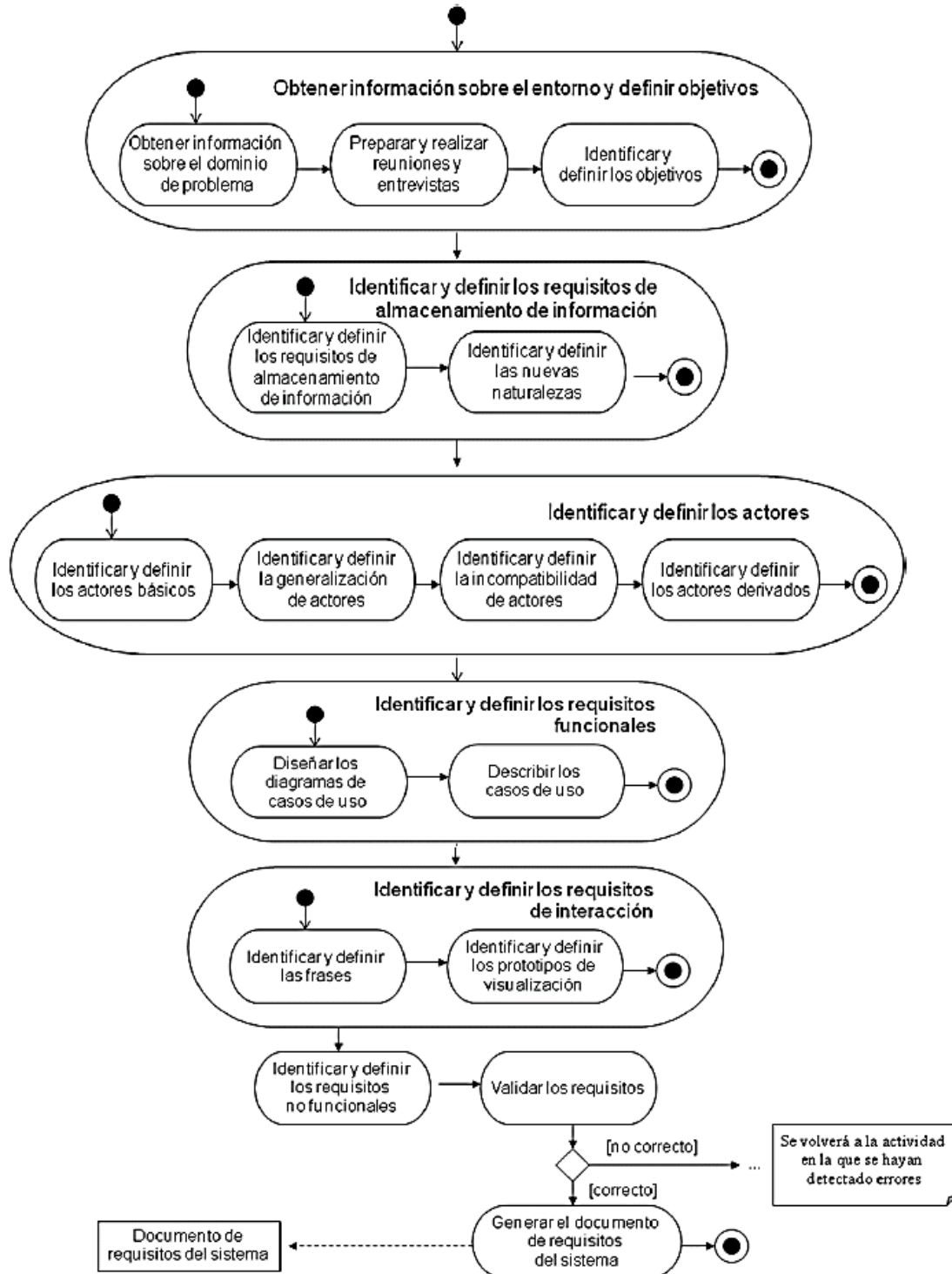


Figura 12 Diagrama de Proceso de NDT. Fuente: Del Valle Rojo, Silvana (2012)

### 2.3.4.4.2. UWE

(LMU – Ludwig-Maximilians-Universität München, Institute for Informatics., 2015) , indica que UWE (UML – Bases Web Engineering) es una propuesta basada en UML y en el proceso unificado para

modelar aplicaciones web, donde producir sistemas adaptativos y la sistematización son aspectos importantes en la UWE.

Esta metodología está considerada como una extensión del estándar UML, pero también se basa en otros estándares como lo son el XMI, como modelo de intercambio de formato, MOF para el meta modelado, los principios del modelado de MDA, el modelo de transformación del lenguaje QVT y XML

El modelo que propone UWE está compuesto 6 sub-modelos:

- a. Modelo de Casos de Uso: plasma los requisitos del sistema.
- b. Modelo de Contenido: es un modelo conceptual para el desarrollo del contenido.
- c. Modelo de Usuario: es modelo de navegación, en el cual se incluyen modelos estáticos y modelos dinámicos.
- d. Modelo de estructura: en el cual se encuentra la presentación del sistema y el modelo de flujo.
- e. Modelo Abstracto: incluye el modelo a de interfaz de usuario y el modelo de ciclo de vida del objeto.
- f. Modelo de Adaptación.

En cuanto a los requisitos, UWE los clasifica dependiendo del carácter de cada uno. Además distingue entre las fases de captura, definición y validación de requisitos.

#### **2.3.4.4.2. Característica**

En el trabajo de investigación realizado por Según Del Valle Rojo, S. (2012), hace mención de las siguientes características de la metodología UWE, las cuales son:

- a. El proceso de UWE alcanza todo el ciclo de vida de desarrollo de aplicaciones Web.
- b. Está compuesto por 3 fases o etapas.
- c. Se centra en el usuario en la etapa de la captura de requerimientos.
- d. Centra su atención en aplicaciones Web personalizadas o adaptativas.

- e. En la fase de captura de requerimientos, hace la separación entre actividades de elicitación, especificación y validación de requerimientos.
- f. Utiliza UML extendido denominado UML Profile, para generar los modelos propuestos en el proceso.
- g. El proceso está más enfocado a actividades de análisis y de diseño utiliza el paradigma de orientación a objetos.
- h. Los modelos son refinados en las iteraciones que plantea el proceso de desarrollo.
- i. Las herramienta Case que soporta UWE, son MagicUWE, tidyDiagram, ArgoUWE, UWEet (paleta), UWE4JSF, UWE2FACPL, MagicSNP

#### 2.3.4.4.2. Etapas de la UWE

Según Del Valle Rojo, S. (2012), menciona y describe las etapas de la UWE, las cuales son:

- I. Capturar requerimientos: Esta etapa tiene por objetivo encontrar los requerimientos que la aplicación Web debe satisfacer y representarlos a través de un modelo de casos de uso, dado que es una técnica centrada en el usuario que permite definir quiénes serán los actores que van a interactuar con la aplicación y representa la funcionalidad que la aplicación Web deberá cumplir para cada actor.

Como técnicas de captura, especificación y validación de requerimientos UWE propone: para captura entrevistas, cuestionarios y checklists; para la especificación glosarios, escenarios y casos de usos, para validación Walk-Through, auditorias y prototipos.

UWE identifica requerimientos funcionales (contenido, presentación, navegación, adaptación) y no funcionales, si bien no establece lineamientos para representar los Requerimientos No Funcionales, los mismos serán considerados en las etapas siguientes.



II. *Analizar y diseñar*: Esta etapa toma como entrada el modelo de casos de uso a fin de generar:

El modelo conceptual del dominio: Debe tratar de no tomar en cuenta los aspectos de navegación, presentación y adaptación. El resultado es un modelo de clases UML que contiene los atributos, operaciones de cada clase, además muestra gráficamente las relaciones de asociación, dependencia, jerarquía entre las clases del modelo; identifica interfaces y restricciones.

El modelo de navegación: Este se diseña a partir del modelo conceptual, las decisiones de diseño están basadas en modelo conceptual, modelo de casos y los requerimientos de navegación. El resultado es un modelo que muestra un diagrama de clases UML que representa las clases de navegación y las asociaciones entre estas.

El modelo de estructura de navegación: Describe como es la navegación a través del espacio de navegación. Dicho en otras palabras muestra como el usuario alcanzará las clases definidas en el modelo de navegación. Este modelo sirve de guía para construir el modelo de Presentación.

El modelo de presentación: Se enfoca en visualizar como es la organización estructural no en la apariencia física (formato, color etc.). Hay muchas posibilidades de construir un Modelo de presentación en base a Modelo de estructura de navegación dado. Los autores proponen dividir el modelo en dos. Por un lado presentar el usuario el árbol de navegación y por otro el contenido. La profundidad del árbol de navegación en esta etapa de modelado debe ser limitada. El resultado es un Diagrama UML de clases construidas con estereotipos UML para elementos de presentación.

III. *Realizar la implementación*: Esta consiste en implementar los modelos validados e integrarlos. Se implementan desde los prototipos de interfaz de usuario hasta el código.

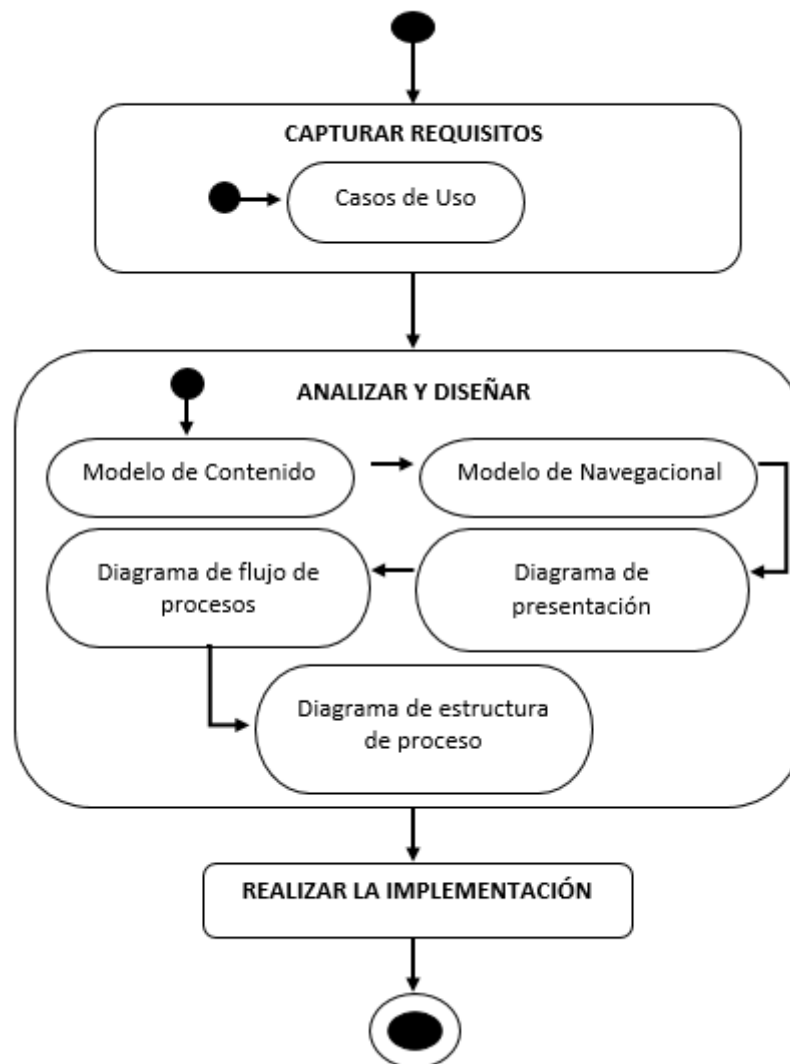


Figura 13 Diagrama de Proceso de UWE. Fuente: Elaboración propia, basado en Del Valle Rojo, Silvana (2012)

### 2.3.4.5. Ingeniería de Requerimiento

#### 2.3.4.5.1. Definición de requerimientos:

“Un requerimiento se define como una propiedad que debe exhibir el software para resolver algún problema del mundo real.” (Swebook, 2012, pág. 24)

“Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”. (Std 610.12-1900, IEEE: 62).

Según (Sommerville I. , 2005, pág. 83) en el proceso de ingeniería de requisitos se clasifican en dos niveles:

- a. Requerimientos de usuarios que es lo que es usuario necesita y lo que el usuario espera del sistema.
- b. Requerimientos del sistema: Aquí se habla de una manera más detalla de lo que el sistema va a realizar, como, cuáles serán sus funciones y que restricciones va a tener el sistema.

#### 2.3.4.5.2. Características de un requisito

Según (Gomez Fuentes, 2011) las características que debe tener un requisito son:

- a. Deben ser correctos. Es por ello que se recomienda ser revisado tanto por el cliente como el desarrollador para así asegurar la inexistencia de errores.
- b. Deben ser consistentes. Los requerimientos son inconsistentes cuando es imposible satisfacerlos simultáneamente.
- c. Deben estar completos. El conjunto de requerimientos está completo si todos los estados posibles, cambios de estado, entradas, productos y restricciones están descritos en alguno de los requerimientos.
- d. Deben ser realistas. Todos los requerimientos deben ser revisados para asegurar que son posibles.

#### 2.3.4.5.3. Captura de requisitos

La captura de los requerimientos tiene como objetivo principal la comprensión de las funcionalidades del sistema esperada por los clientes y los usuarios. Se refiere al “cómo” se van a recolectar los requisitos por parte del ingeniero de software. Es importante en este punto entender la problemática a solucionar, para ello es importante que la primera tarea a realizar es la identificación de los stakeholders. “*La clave del éxito de la captura de requisitos es la comunicación entre los usuarios del software y los ingenieros*” (Swebok, 2012, pg. 34).

Existen una variedad de técnicas para la captura de los requisitos:

- a. Entrevistas
- b. Prototipos
- c. Escenarios
- d. Observación
- e. Tormenta de ideas ( Brainstorming

#### 2.3.4.5.4. Análisis de requisitos:

Según el autor Eric J. Braude para poder analizar los requisitos los clasifica en dos niveles: (Braude, 2003, pág. 135)

- a. En el primer nivel se encarga de documentar todos los deseos y necesidades del cliente, la cual son expresados y capturados a través del lenguaje natural, a este nivel se le conoce como requisitos del cliente o como requisitos “C”.
- b. En el segundo nivel es donde se encarga de documentar todos los requisitos capturados en el nivel anterior de una manera específica y estructurada, a este nivel se le conoce como requisitos del desarrollador o requisitos “D”

#### 2.3.4.5.5. Clasificación de requisitos

Los requisitos se clasifican de dos maneras:

- I. **Requisitos funcionales:** Aquí se detalla todo lo que el sistema quiere que haga (sus funcionalidades) (Braude, 2003)  
Se especifica los servicios que el sistema debe de ofrecer, en como reaccionaria el sistema en entradas particulares, como debería reaccionar ante situaciones específicas y muchas veces también se especifica lo que el sistema no va a realizar. (Sommerville I. , 2005, pág. 84)
- II. **Requisitos no funcionales:** *“Son limitaciones sobre servicios o funciones que ofrece el sistema que incluye restricciones tanto de temporización y del proceso de desarrollo”* (Sommerville I. , 2005, pág. 85).

Los requisitos nos funcionales los clasifica de la siguiente manera, hace referencia (Braude, 2003):

- a. Requisito de rendimiento: *“Aquí se especifica las restricciones de tiempo que debe observar la aplicación”*. Se especifica el tiempo de demora que tardara la aplicación en realizar una acción.
- b. Requisito de confiabilidad y disponibilidad: *“Especifican la confiabilidad en términos cuantificados”*
- c. Manejo de errores: Aquí se especifica como la aplicación debe reaccionar ante un eventual error durante su ejecución, ya sea un error propio de la aplicación o un error en el manejo de ella.
- d. Requisitos de interfaz. *“Se describen el formato con el que la aplicación se comunica con su entorno”*
- e. Restricciones: *“Las restricciones de diseño e implementación describen los límites o condiciones para diseñar o implementar la aplicación”*
- f. Requisito inverso: Aquí se especifican lo que el software no hace.

Según Sommerville clasifica los requerimientos no funcionales de la siguiente manera (Sommerville I. , 2005):

- a. Requerimientos del producto: *“Especifica o restringen el comportamiento del software”*
- b. Requerimientos de la organización: *“Son requerimientos de sistemas amplios derivados de políticas y procedimientos en la organización y del desarrollador”*
- c. Requerimientos externos: *“Este término cubre todos los requerimientos derivados de factores externos del sistema y su procesos de desarrollo”*

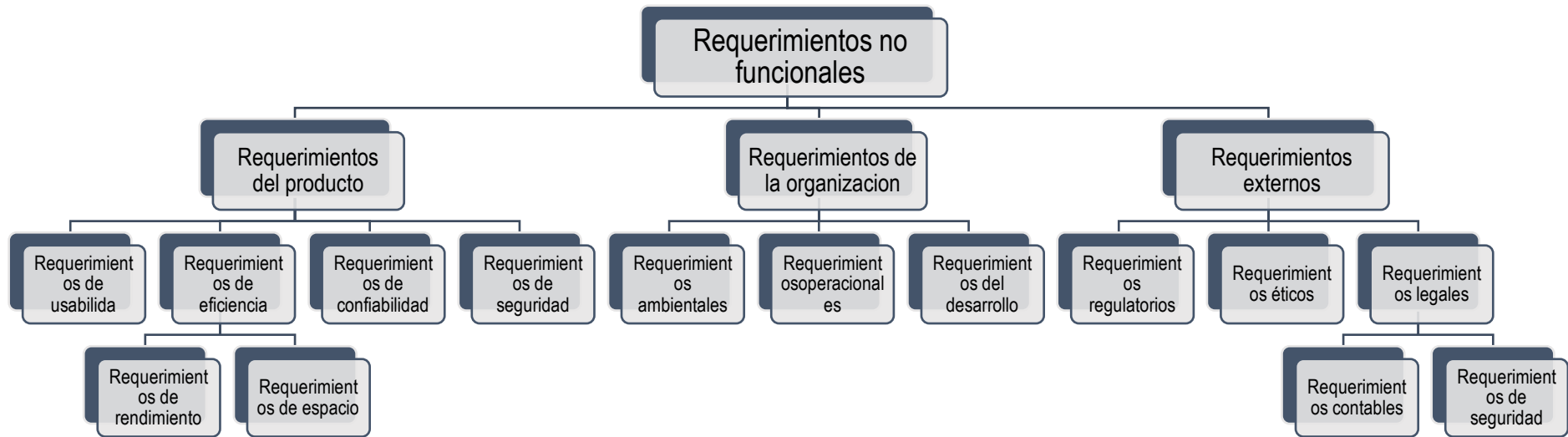


Figura 14: Requerimientos no funcionales. Fuente: (Sommerville I. , 2005)



#### 2.3.4.5.6. Especificación de requisitos

Los requisitos son plasmados en un documento, para poder facilitar la verificación por parte de los actores del proyecto. Es usual que se hagan tres documentos:

- a. Definición del sistema o documento de exigencias
- b. Requisitos de sistema
- c. Requisitos de software

#### 2.3.4.5.7. Validación de requisitos

En este punto de un proyecto software todos los documentos realizados deben ser comprensibles por cualquier usuario, y deben estar aptos a una auditoria final por todos los involucrados en el proyecto, para asegurarse que lo plasmado allí es lo que fue solicitado:

- a. Revisión de los requisitos: llevado a cabo por un comité revisor, que incluye a un representante del cliente, con el fin de evaluar los documentos ya mencionados.
- b. Prototipado: Es un medio que utiliza el ingeniero de software para manifestar lo que entendió y para facilitar la corrección, eliminación o adición de requisitos. Las ventajas de aplicar esta estrategia para plasmar lo solicitado puede ser una gran ventaja, pero a la vez muy costoso
- c. Pruebas de Aceptación: aplicada a cada requisito para determinar si el producto satisface las necesidades plasmadas del cliente. Estas pruebas deben hacerse con planeación y de manera cuantitativa.

### 2.3.5. Calidad del Software

#### 2.3.5.1. ¿Qué es la calidad?

Según (Calidad Integral, 2017) Menciona que calidad es el *“Grado en el que un conjunto de características inherentes cumple con los requisitos”*.

#### 2.3.5.2. Calidad de Software

Según (Pressman S., 2010) citado por (Lopez Echeverry, Cabrera, & Valencia Ayala, 2008) define a la calidad de Software como

la “*Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario*”.

### 2.3.5.3. Tipo de calidad en el desarrollo de un proyecto

(Sommerville I. , 2005, pág. 592) Clasifica a la calidad en:

#### a) Calidad del proceso

Cuando hablamos de calidad del proceso nos hace referencia a la calidad de un producto software, ya que este depende por la calidad que tenga en cada uno de sus procesos.

#### b) La calidad del producto

Es el conjunto de propiedades o características que tiene un producto para satisfacer las necesidades del cliente.

### 2.3.5. ¿Qué es la ISO?

Según (Iso.org, 2015) La Organización Internacional Independiente (ISO), se encuentra integrada por 153 países, teniendo como objetivo de promover la estandarización de los diferentes productos, servicios y bienes para poder garantizar un producto o servicio de calidad.

#### 2.3.5.1. ¿Qué son normas o estándares?

Una definición según (Iso.org, 2015) es un conjunto de requisitos y especificaciones que se toma como referencia para así poder asegurar que los procesos, servicios o productos son de calidad.

La importancia que tiene los estándares en el desarrollo de software según (Sommerville I. , 2005, pág. 587) son:

- a. Una norma o estándar ya han sido probadas y están certificadas para poderse aplicar en una empresa



- b. Ofrecen una guía para que se puedan aplicar de manera correcta y así poder garantizar la calidad del producto o servicio.
- c. Como ofrece un marco de trabajo permite que todos trabajen bajo los mismos estándares reduciendo así el esfuerzo de aprendizaje.

### 2.3.5.2. ISO 9126

(Moraga, Piattini, & Calero, 2010, págs. 55-58) *“El modelo ISO para la calidad del producto software, IO/IEC 9126 se publicó por primera vez en 1991 definiendo la calidad del software como el grado en el que el sistema satisface las necesidades implícitas y explícitas de los diferentes usuarios”.*

Las características que debe tener un producto software según el estándar son:

- I. Funcionalidad: Es la capacidad de poder cumplir las necesidades del usuario a través de las funcionalidades del software, cuando este sea utilizado.
  - a. Precisión
  - b. Adecuación
  - c. Interoperabilidad
  - d. Seguridad
  - e. Conformidad
- II. Eficiencia: Es la capacidad del software de desempeñarse utilizando lo más mínimo de recursos posibles en el menor tiempo para lograr su objetivo.
  - a. Comportamiento temporal
  - b. Utilización
- III. Mantenibilidad: Un software es mantenible cuando el esfuerzo para modificar, mejorar es bajo.
  - a. Facilidad de análisis.
  - b. Estabilidad
  - c. Facilidad de prueba

- IV. **Fiabilidad:** Capacidad del software cumpla con las actividades requeridas por el usuario manteniendo el rendimiento y durante un tiempo establecido.
  - a. Madurez
  - b. Tolerancia a fallo
  - c. Recuperabilidad
- V. **Usabilidad:** Capacidad que tiene un producto software para ser utilizado de una manera fácil y sencilla.
  - a. Compresibilidad
  - b. Facilidad de aprendizaje
  - c. Operabilidad
  - d. Atractividad
- VI. **Portabilidad:** Todo software es portable cuando este puede ser trasladado de un entorno a otro sin problemas.
  - a. Adaptabilidad.
  - b. Facilidad de instalación
  - c. Co-existencia
  - d. Reemplazabilidad

#### **2.3.5.2.1. Modelo ISO para la calidad del producto software**

Según (Largo Garcia & Marin, 2005) La ISO/IEC 9126 es un marco de trabajo para la evaluación de un producto software durante todo el ciclo de vida del software.

La calidad de un producto software según la norma ISO/IEC 9126 se presenta de dos formas:

- I. **Calidad interna y externa:** Para la calidad interna y externa tenemos las siguientes características:



Figura 15: Calidad Externa e Interna. Fuente:(Fernández, 2014)

II. Calidad en uso: Para la calidad en uso tenemos las siguientes características:



Figura 16: Calidad en Uso. Fuente: (Fernández, 2014)

### 2.3.5.3. Usabilidad

(Fernández, 2014), hace referencia que algunos autores de la comunidad de la Ingeniería del Software proponen la incorporación de la usabilidad incluso antes del diseño de la arquitectura del sistema, es decir, en la fase de captura de requisitos. Donde el trato explícito de la usabilidad en la especificación de los requisitos es importante para

proporcionar una visibilidad de las características de la usabilidad tanto para los desarrolladores como para los testadores.

La palabra usabilidad no existe en el diccionario, sino que es un anglicismo que en castellano se puede traducir literalmente como “facilidad de uso”. Se refiere a la facilidad con la que se puede utilizar una herramienta, objeto o interfaz web.

Aunque ya existía y se aplicaba desde mucho antes para el diseño de herramientas e interfaces, en otros ámbitos se empezó a hablar del termino usabilidad con la proliferación de la interacción entre personas y ordenadores.

Jakob Nielsen está considerado como el padre de la usabilidad y define como:

- a. La usabilidad es una cualidad de la que el usuario no se da cuenta si está presente, porque puede interactuar con facilidad y rapidez. Pero en cambio sí se nota mucho su ausencia, por que el usuario no sabe cómo interactuar, se produce errores y el usuario se queja.

La organización internacional de estandarización (ISO) define la usabilidad en varios documentos donde según ISO/IEC 9241:

- b. ISO/IEC 9241 citado por (Fernández, 2014, pág. 90) menciona que la *“Usabilidad es el grado en que un producto puede ser usado por usuarios específicos, para alcanzar unos objetivos determinados con eficacia, eficiencia y satisfacción, en un contexto de uso específico”*.

Por otro lado, según la norma ISO/IEC 9126 citado por (Fernández, 2014, pág. 90) define a la usabilidad como:

- c. *“La capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario en condiciones específicas de uso”*.

Es importante señalar que estas definiciones hablan de usabilidad para usuarios, objetivos y contextos específicos.

La usabilidad, desde el punto de vista de la satisfacción del usuario, es un concepto abstracto difícil de medir, que está relacionada con lo útil que puede ser, lo eficiente, cómodo y fácil de usar. Para Jakob Nielsen, la usabilidad está definida en función de cinco componentes que influyen en el diseño, hace referencia (Fernández, 2014, pág. 91) las cuales son:

- a. Facilidad de aprendizaje
- b. Eficiencia: La velocidad con la que pueden realizarse las tareas
- c. Errores: Cuantos se comenten y si es fácil de solucionarlos
- d. Satisfacción: La valoración subjetiva del usuario

### 2.3.5.3.1. Importancia de la Usabilidad

(Técnicas Avanzadas de Diseño Web, 2009, pág. 121) *“La usabilidad es importante debido a que un sitio web usable requiere menos esfuerzo para su aprendizaje y reduce los errores cometidos por los usuarios”*, lo cual permite que las tareas que se realicen en nuestro sistema Web sean más eficientes y efectivas, para así lograr la

**Usabilidad = Simplicidad = Satisfacción del usuario = Beneficios**

satisfacción del usuario.

Entonces se puede decir que la usabilidad es importante por qué:

- a. Aumenta la satisfacción de los usuarios del sitio web.
- b. Aumenta el número de usuarios que se convierten en clientes.
- c. Mejora de la imagen, el prestigio del sitio y de la empresa.
- d. Reduce costos de aprendizaje por parte de los usuarios.
- e. Reduce de costos de mantenimiento y rediseño.

### 2.3.5.3.2. Ventajas y dificultades en la implantación de sitios web usables

Según (García, 2015, pág. 47) La usabilidad de un sitio web mejora la experiencia de usuario y tiene beneficios económicos tangibles:

- f. Mayor productividad: En muchos casos, como podría ser una aplicación web de gestión un sitio web es una herramienta de trabajo. Si está bien diseñada los usuarios serán capaces de realizar sus tareas más rápido.
- g. Menores costes en formación y soporte: Una de las características de la usabilidad es la facilidad de aprendizaje. Una página usable requerirá una inversión menor en formación y soporte de los empleados.
- h. Más visitas y ventas: Si un web es agradable y fácil de usar es más probable que los usuarios vuelvan a visitarla. Además, si se trata de una página.

(Panach Navarrete, 2010, pág. 11) Hace referencia que históricamente, la ingeniería de software ha tendido a separar el diseño de la interfaz del resto de etapas del proceso de desarrollo software, argumentando que las interfaces se podrían diseñar en las últimas etapas del desarrollo independiente de la funcionalidad del sistema. Al dejar el diseño de la interfaces en las últimas etapas del proceso de desarrollo de software evitan que las características de usabilidad que se incluyan en el sistema sean atendidas en tempranas etapas. Por otro lado, han incrementados los trabajos en la comunidad IPO que están relacionados con el diseño de interfaces y guías de usabilidad que deben seguir las interfaces, esta comunidad ha aportado una serie de recomendaciones relacionados a la usabilidad que debe ser incluida en los sistemas software.

### 2.3.5.3.3. Mecanismo de usabilidad

(Juristo N., Moreno A., Sanchez M. 2007), definieron los patrones de elicitación de la usabilidad (Usability Elicitation Patterns, USEPs). Esta guía fue realizada bajo las características Funcional Usability Features (FUFs), que según hace mención (Panach J., 2010) estas características fueron propuestas por Persona-Ordenador (IPO), que

tienen como objetivo tratar la usabilidad desde las primeras etapas del proceso de desarrollo de software. La principal ventaja de esta aproximación es que la usabilidad se tiene en cuenta en la fase de construcción de la arquitectura y se evitan modificaciones de la misma una vez se haya representado toda la funcionalidad de la lógica de negocio.

En el siguiente cuadro se realiza la descripción de las características funcionales de usabilidad (FUFs):

Tabla 5 Descripción de las características funcionales de usabilidad (FUFs). Basado en el trabajo de (Panach Navarrete, 2010)

Característica funcional de usabilidad	Descripción	Mecanismo de la usabilidad	Objetivo
<b>Feedback</b>	El objetivo de este FUF es el mantener informado al usuario las acciones o procesos que se están ejecutando en el sistema.	System Status Feedback	Informar al usuario de cambios relevantes que se producen en el sistema o cuando se produce un fallo. Antes, durante y después de la ejecución de un servicio.
		Interaction Feedback	Informar al usuario de sus peticiones e interacciones que ha realizado.
		Progress Feedback	Mostrar el tiempo restante para la finalización de un servicio lanzado por el usuario. Útil para aquellos servicios cuya ejecución requiere más de dos segundos.
		Warning	Confirmar al usuario en caso de que el servicio solicitado tenga consecuencias irreversibles, evitando que el usuario cometa errores. El sistema informa al usuario las consecuencias de ejecución de la acción.



Undo/Cancel	Permite al usuario cancelar una acción o volver a un estado anterior al cual se encuentra.	Global undo	Incorpora en los sistemas la posibilidad de deshacer las últimas acciones que haya realizado el usuario.
		Abort Operation	Permite al usuario cancelar una acción antes de que esta finalice su ejecución.
User input errors prevention/Correction	Facilitar la entrada de datos, evitando que el usuario cometa errores en digitalizar la información solicitada en campos con formato específico de entrada de datos.	Structured Text Entry	Especifica elementos del interfaz que facilitan la entrada de datos con un formato específico.
Wizard	Ayuda a los usuarios a ejecutar acciones que requieren más de un paso.	Step by Step	Se utiliza para ayudar al usuario en tareas que son complejas y requieren la inserción de gran cantidad de datos.
User profile	Permite al usuario adaptar el sistema a sus preferencias.	Preferences	Se utiliza este patrón para especificar qué aspectos visuales se pueden personalizar por el usuario.
		Personal Object Space	Se incorpora en la aplicación la funcionalidad de que el usuario pueda modificar la posición de los elementos de la interfaz según prefiera.
		Favourites	Se habilita en el usuario la funcionalidad de crear accesos rápidos, permitiendo al usuario el ingreso rápido a aquellas funcionalidades que sean de interés primordial.





<b>Help</b>	Proporciona ayuda al usuario sobre cómo llevar a cabo las acciones del sistema.	Multilevel help	Introduce la funcionalidad para ayudar al usuario sobre el funcionamiento del sistema.
<b>Command aggregation</b>	Ayuda al usuario a lanzar la ejecución de más de una acción mediante el uso de comandos.		
<b>Shortcuts</b>	Permite a los usuarios lanzar una acción de forma rápida.		
<b>Reuse information</b>	Permite mover fácilmente información de una parte del sistema a otra parte.		

La comunidad IPO considera que a partir de este nivel de descripción de los FUFs, el analista debe ser capaz de seleccionar las funcionalidades a implementar en el sistema de acuerdo a los objetivos del proyecto y usuarios.

### 2.3.6. Herramientas de Desarrollo

#### 2.3.6.1. PHP

(The PHP Group, 2016) Indica que Hypertext Preprocessor(PHP) es un lenguaje de código abierto para el desarrollo Web y que puede ser incrustado en HTML.

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, Microsoft Windows, Mac OS X, RISC OS, etc, que admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros.

PHP es un lenguaje de programación que puede hacer frente a utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

#### 2.3.6.2. Postgres

Según (Martinez, 2013)Postgres es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Postgres utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.



A continuación teneis algunas de las características más importantes y soportadas por PostgreSQL:

- a. Es una base de datos 100% ACID
- b. Integridad referencial
- c. Tablespace
- d. Nested transactions (savepoints)
- e. Replicación asincrónica/sincrónica / Streaming replication - Hot Standby
- f. Two-phase commit
- g. PITR - point in time recovery
- h. Copias de seguridad en caliente (Online/hot backups)
- i. Unicode
- j. Juegos de caracteres internacionales
- k. Regionalización por columna
- l. Multi-Version Concurrency Control (MVCC)
- m. Múltiples métodos de autenticación
- n. Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

#### 2.3.6.3. XAMP

XAMP es un servidor web Apache, esta incluye un paquete de herramientas de desarrollo, como son:

- a. Apache, el famoso servidor Web
- b. MySQL, una excelente, libre, datos de código abierto
- c. PHP, el lenguaje de programación (en las versiones 4.3.8 y 5.0.1 en el momento de escribir este documento)
- d. Perl, el lenguaje de programación
- e. ProFTPD, un servidor FTP
- f. OpenSSL, por secure sockets soporte de la capa

#### 2.3.6.4. NDTSuite

Según (IWT2, 2016), NDT-Suite es un conjunto de herramientas para aplicar la metodología NDT. NDT ha sido una metodología muy aplicada pero la única forma de que pueda ser una realidad empresarial es por ello que requiere de una herramienta que de soporte al desarrollo.

La NDT-Suite aborda el ciclo de vida de desarrollo de software propuesto por la metodología NDT. Esta suite está compuesta por las siguientes herramientas:

- a) NDT-Profile: implementa un perfil definido sobre la herramienta Enterprise Architect, donde se define cada uno de los metamodelos y artefactos a utilizar con la NDT.
- b) NDT-Quality esta herramienta analiza un proyecto realizado en base a la NDT-Profile y comprueba que la trazabilidad y las normas de NDT se cumplan, y en caso que no sean así te da los puntos a corregir.
- c) NDT- Driver: es una herramienta que permite llevar acabo las transformaciones entre los distintos modelos NDT definidos en un proyecto desarrollado con NDT-Profile.
- d) NDT-Prototypes: esta herramienta genera de forma automática un conjunto de prototipos XHTML a partir de los modelos de navegación descritos en la fase de análisis del ciclo de vida de un proyecto desarrollado con NDT-Profile.
- e) NDT-Glossary: Da soporte a la tarea de Elicitación de Requisitos, englobada dentro de la fase de Ingeniería de Requisitos del ciclo de vida de desarrollo de un producto software.
- f) NDT- Checker: herramienta que contiene un conjunto de plantillas, diferentes por cada elemento de NDT.
- g) NDT-Counter: esta herramienta da soporte en la etapa de gestión de proyectos, específicamente en la tarea de estimación de costes del proyecto a realizar.
- h) NDT-Report: Herramienta que genera la documentación necesaria del proyecto de acuerdo a la fase del ciclo de vida propuestos por la NDT (desde la etapa de captura de requisitos, análisis, diseño, pruebas, mantenimiento).

### 2.3.6.5. MagicUWE

Según (LMU – Ludwig-Maximilians-Universität München, Institute for Informatics., 2015) MagicUWE es un herramienta CASE que ha sido desarrollada para el diseño asistido por ordenador de las aplicaciones Web, utilizando la metodología UML-based Web Engineering (UWE). MagicUWE está construido como un plugin para MagicDraw 18.0 .

El objetivo de MagicUWE es hacer el diseño de aplicaciones web con UWE y MagicDraw muy fácil, por lo tanto, una interfaz de usuario intuitiva es esencial. Con MagicDraw podemos crear todos los tipos de diagramas UWE fuera del menú contextual de MagicDraw "Nuevo Esquema" y no sólo utilizar la barra de herramientas familiarizado con su última funcionalidad utilizada, sino también seleccionar rápidamente el artículo que requiera, reconocible por su símbolo estereotipo.

## 2.4. Definición de la terminología

- a) Método: Consecución de acciones para lograr un propósito, estos determinan el orden y la forma como estos se realizarán.
- b) Metodología:(Condor, 2013) la define como un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado.
- c) Proceso de software: conjunto de políticas, estructuras organizativas, tecnologías, procedimientos y artefactos que se necesitan para concebir, desarrollar, implantar y mantener un producto software, (Ruiz, 2016)

- d) Modelo: es una representación de un objeto, sistema o idea, de forma diferente al de la entidad misma, cuyo propósito es ayudar a explicar, entender o mejorar un sistema,(MARTÍNEZ PÉREZ , 2012).
- e) Elicitación: es el proceso que consiste en adquirir todo el conocimiento relevante, necesario para producir un modelo de requerimientos de un dominio de problema, (Thomas,P., 2006, pág. 10).
- f) Evaluación: acción realizada para señalar el valor de algo. Estimar, apreciar, calcular el valor de algo, (Lozada , 2013).
- g) Satisfacción del cliente: percepción del cliente sobre el grado en que se han cumplido sus requerimientos.(iso9001calidad.com, 2013)
- h) Requisitos: Necesidad o expectativa establecida, generalmente implícita u obligatoria, <sup>1</sup>(iso9001calidad.com, 2013)
- i) Medir: Comparar una cantidad con su respectiva unidad, con el fin de averiguar cuantas veces la segunda está contenida en la primera, (Real Academia Española, 2014)

## CAPITULO III: MARCO METODOLÓGICO

### 3.1. Tipo y diseño de la investigación

#### 3.1.1. Tipo de investigación

Primero debemos saber que las investigaciones aplicadas son las repuestas fundamentales a un problema detectado, la cual se concentra en poner en práctica las teorías generales para resolver los problemas y necesidades que se plantea en un corto, mediano o largo plazo. (Hernández & Ander, 2011)

Por otro lado una investigación comparativa busca descifrar la relación entre dos o más variables documentando las diferencias o similitudes observadas entre dos o más sujetos. Por consiguiente, la comparación implica una observación en un marco más "natural," no sujeto a límites experimentales y, de esta manera, evoca similitudes con la descripción. (Carpi & Anne E., 2008)

En este proyecto de tesis el tipo de investigación es aplicada comparativa, dado a que en esta investigación se realizar la comparación de dos metodologías integrando a cada una de estas metodologías unas ciertas directrices, con el fin de evaluar si el producto final medido a través de una ISO cumple con las perspectivas de los stakeholders (usuario final).

#### 3.1.2. Diseño de la investigación

El diseño de la investigación marca el plan o la estrategia que se desarrollará para obtener la información que se requiere en una investigación, y con ella intentamos obtener respuestas a preguntas como: Contar, medir y describir.

El diseño de esta investigación es cuasi-experimental, dado a que en este tipo de diseño de investigación no se utiliza ningún tipo de selección aleatoria o proceso de pre-selección para escoger los grupos en las que se prueba una variable.

Los diseños cuasi-experimentales manipulan deliberadamente, al menos, una variable independiente para observar su efecto y relación con una o más variables dependientes, (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2010, págs. 120,148).

### 3.2. Población y muestra

#### 3.2.1. Población

La población de la presente investigación son 10 metodologías, basados en la investigación de (Escalona., M. y Koch N, 2004) se realizó un estudio comparativo de 10 metodologías de desarrollo en Web, que muestran las tendencias en el uso de técnicas para la captura, especificación y validación de requerimientos Web. Investigación es realiza dado a que la mayoría de las metodologías que se han propuesto para el desarrollo de aplicaciones Web se centran en el diseño prestando menos atención a la Ingeniería de requisitos.

Tabla 6: Metodologías de Desarrollo Web. Fuente: Elaboración propia

Metodologías de desarrollo Web	WDSM (Web Site Design Method)
	SOHDM (Scenario-based Object-oriented Hypermedia Design Methodology)
	RNA (Relationship Navigational Analysis )
	HFPM (Hypermedia Flexible Process Modeling)
	OOHDM (Object Oriented Hypermedia Design Methodology)
	UWE (UML- based Web Engineering)
	W200
	WebML (Web Modeling Language)
	NDT (Navigational Development Thechniques)
	DDDP (Design-Driven Requirements Elicitation)



### 3.2.2. Muestra

La muestra seleccionada para esta investigación son dos metodologías de desarrollo de Software Web, UWE y NDT. Metodologías cuyo proceso de desarrollo fueron analizadas en la investigación de Rojo (2012 - Argentina), y se identificaron los tipos de requerimientos No funcionales, contemplados por cada uno, así como la técnica de ingeniería de requerimientos propuestas por estos para su tratamiento y gestión. Según el estudio realizado se puede visualizar que la usabilidad es la característica del modelo ISO/IEC 9126 considerada por todos procesos y más relevante.

### 3.3. Hipótesis

La integración de directrices para la captura de requisitos de Usabilidad, en metodologías de desarrollo web bajo el paradigma dirigido por modelos, permitirá capturar requisitos de usabilidad en etapas tempranas de desarrollo de software logrando la satisfacción del cliente.

### 3.4. Variables

#### 3.4.1. Variables dependientes

Usabilidad en etapas tempranas de desarrollo de software Web.

#### 3.4.2. Variables independientes

La integración de directrices para la captura de requisitos de Usabilidad, en metodologías de desarrollo web bajo el paradigma dirigido por modelos.



### 3.5. Operacionalización

Variable independiente	Dimensiones	Indicador	Descripción	Fórmula	Técnicas e Instrumentos de Recolección de datos
La integración de directrices para la captura de requisitos de Usabilidad, en metodologías bajo el paradigma dirigido por modelos.	Factibilidad	Tiempo	Tiempo de desarrollo.	$X = T_F - T_i$	Análisis estadístico de pruebas – Guías de Observación.
		Total de presentaciones	Número de presentaciones para ser aceptado por el usuario.	$X = T_1 + T_1 + \dots$	Análisis estadístico de pruebas – Guías de Observación.
		Total de artefactos	Cantidad de artefactos que incluyen a la usabilidad.	$X = A_i + A_i + \dots$	Análisis estadístico de pruebas – Guías de Observación.

Variable Dependiente	Dimensiones	Indicador	Descripción	Fórmula	Técnicas e Instrumentos de Recolección de datos
Usabilidad en etapas tempranas de desarrollo Web	Facilidad de Aprendizaje	de Tiempo aprendizaje	T= Tiempo medio que tarda el usuario aprender una determinada función.	$X = \frac{(t_i + \dots + t_n)}{t_n}$	Análisis estadístico de pruebas – Guías de Observación.
		Valorización aprender	A= Según desde la perspectiva del usuario que tan fácil le resulto manejar el software.	$X = \frac{(A_i + A_i + \dots + A_n)}{A_n}$	



		Tiempo de operación	T=Tiempo en que el usuario llevar a cabo la tarea especificada.	$X = \frac{(t_1 + t_2 + \dots + t_n)}{t_n}$	
Comprensibilidad	Comprensión de las funcionalidades.	A=Contar el número de funciones que se entienden de manera adecuada. B=número total de las funciones comprendidas en el producto.	A=Contar el número de funciones que se entienden de manera adecuada. B=número total de las funciones comprendidas en el producto.	$X = \frac{(A_1 + A_2 + \dots + A_n)}{B_T}$	Análisis estadístico de pruebas – Guías de Observación.
	Salida y entrada de elementos	A = Número de elementos de entrada de datos y salida que el usuario entienda con éxito B = Número de elementos de datos de entrada y salida disponible en la interfaz	A = Número de elementos de entrada de datos y salida que el usuario entienda con éxito B = Número de elementos de datos de entrada y salida disponible en la interfaz	$X = \frac{(A_1 + A_2 + \dots + A_n)}{B_T}$	Software de monitorio
Operabilidad	Coherencia de uso	A = Número de mensajes o funciones que encontró el usuario inaceptable e inconsistente con la expectativa del usuario B = Número de mensajes o funciones	A = Número de mensajes o funciones que encontró el usuario inaceptable e inconsistente con la expectativa del usuario B = Número de mensajes o funciones	$X = 1 - \frac{A}{B}$	Análisis estadístico de pruebas – Guías de Observación.



		Corrección de errores	<p>Tc = Número de errores de entrada que el usuario corrige con éxito</p> <p>Ts = Número total de errores que obtuvo el usuario</p>	$X = \frac{T_c}{T_s}$	
		Capacidad de Operabilidad	<p>Tn= Mensajes de alertas que se le presenta al usuario.</p> <p>Ts=Mensajes totales existentes en la funcionalidad.</p>	$X = \frac{T_c + T_{c+1} + T_{c+n}}{T_s}$	
Atractividad		Interacción atractiva	<p>Que tan atractivo es la interfaz para el usuario.</p> <p>Ai= Valoración del usuario según la escala de LIKERT (1-5)</p> <p>An= cantidad total de encuestados.</p>	$X = \frac{(A_i + A_i + \dots + A_n)}{A_n}$	<p>Análisis estadístico de pruebas – Guías de Observación.</p>

	Cumplimiento de la usabilidad	Cantidad de métricas	<p>Y= Contar el número de elementos que cumplen las especificaciones.</p> <p>w= número de artículos que requieren el cumplimiento de la especificación.</p>	$X=1-\frac{w}{Y}$	<p>Análisis estadístico de pruebas – Guías de Observación.</p>
--	-------------------------------	----------------------	---	-------------------	--

### 3.6. Abordaje metodológico, técnicas e instrumentos de recolección de datos

#### 3.6.1. Abordaje metodológico

El presente trabajo de investigación utilizó como método de recolección de datos la Observación, la cual sirvió como instrumento de análisis, La observación científica *"tiene la capacidad de describir y explicar el comportamiento, al haber obtenido datos adecuados y fiables correspondientes a conductas, eventos y /o situaciones perfectamente identificadas e insertas en un contexto teórico"* (Travé Gonzales, 1996).

#### 3.6.2. Técnicas de recolección de datos

Esta investigación hizo uso de instrumentos, como lo son:

##### Entrevistas

Es una técnica que se aplica con frecuencia. Por medio de las entrevistas se logró comprender el problema y obtener información acerca de los objetivos de la aplicación a desarrollar. Este proceso abarcó cuatro etapas, las cuales fueron: identificación de las partes interesadas para la entrevista, elaboración de la entrevista, la propia entrevista y la documentación de los resultados de carácter formal.

##### Cuestionarios



Cuestionario, técnica que consiste en preparar un documento con preguntas para las cuales sólo respuestas breves y concretos o incluso con una selección limitada de respuestas (lista de comprobación) es posible. El cuestionario puede ser completado durante una entrevista o que puede ser utilizado para obtener información de manera independiente de una entrevista. Para la aplicación de esta técnica el analista necesita cierto conocimiento sobre el dominio del problema y la aplicación que se construirá en para preparar el cuestionario

### **Software de Monitoreo de Pcs**

Herramienta software utilizada para supervisar y controlar las actividad del PC y del uso de Internet. Técnica aplicada para supervisar las actividades realizadas por los docentes con el uso de los aplicativos finales. Con esta herramienta se pudo dar seguimiento automático a las interacciones de los usuarios evaluadores del prototipo funcional elaborado producto de esta investigación..

### **3.6.3. Instrumentos de recolección de datos**

#### **Formato de entrevista**

Instrumento que estandariza el proceso de las entrevistas, cuyo aporte principal para esta investigación es el poder llevar de forma ordenada y dinámica las reuniones realizadas con el sponsor. Ayuda a alcanzar los objetivos de la reunión y recolectar la información necesaria.

#### **Formato de cuestionario**

Instrumento utilizado para recolectar datos con respecto a una variable de la investigación a medir.

#### **Bitácora de incidencias**

Instrumento utilizado para documentar un problema durante un tiempo de tal forma que mantengamos una clara y concisa visión de lo que está ocurriendo.

#### **Artefactos de ingeniería de software**

Un artefacto es un producto resultante del proceso de desarrollo de software. Estos fueron utilizados para estandarizar la comunicación

entre los interesados en el proyecto, y así poder comunicar de forma adecuada las expectativas, y diseño del sistema. Algunos artefactos en esta investigación fueron: casos de uso, diagramas de clases y otros modelos UML que ayudan a la descripción de la problemática y solución, arquitectura o diseño del software. Otros se enfocan en el proceso de desarrollo como planes de proyecto, casos de negocio o enfoques de riesgos.

### 3.7. Procedimiento para la recolección de dato

#### Entrevista:

Para la realización de la entrevista se hizo uso de un instrumento que fue la ficha de entrevista la cual fue utilizada con la intención de poder plantear las preguntas de una manera precisa y concisa, con el fin de poder captar de manera eficiente la necesidad del usuario. Para ello se realizó de la siguiente manera:

1. Primero conocimos como es el flujo del proceso de las actividades diarias que ellos realizan para determinados procesos.
2. Teniendo el conocimiento del flujo de sus procesos, se planteó preguntas específicas que sirvieron para poder capturar información requerida para el desarrollo del proyecto.
3. Elaboradas las preguntas se procedió a hacer uso de la ficha de entrevista para establecer el orden en que se le pregunto al usuario final.
4. Realizadas las preguntas, se llegó a un acuerdo con el usuario para poder realizar la entrevista, para ello se tomó en cuenta lo siguiente:
  - a. Identificar todos los tipos posibles de usuarios que pueden existir en la empresa u organización.
  - b. Una vez identificado a los usuarios, se contactó a cada uno de los participantes para ver su disponibilidad de tiempo y así fijar una fecha exacta para llevar a cabo la entrevista.
5. Una vez realizada la encuesta se procedió a lo siguiente:
  - a. Clasificación de las respuestas de los usuarios, según el tipo de pregunta, ya sean preguntas abiertas o preguntas cerradas.

- b. Selección de las respuestas dada por los usuarios, quedándonos con solo aquellas que sirven para lograr el objetivo requerido.
6. Finalmente se obtuvo una matriz de respuesta realizada en Excel de los requerimientos que el usuario desee.

### **Cuestionarios:**

Para la realización de los cuestionarios se hizo uso de una Ficha de Cuestionario, en cual se planteó una lista de preguntas dirigidas al usuario con el fin de captar de manera más eficiente las necesidades de los mismos, para ello se procedió a realizar lo siguiente:

1. Formular las preguntas a realizar al usuario, para esto fue necesario:
  - a. Pensar en que datos necesitamos y como utilizaremos dicha información para así poder plantar las preguntas del cuestionario.
  - b. Planteamos preguntas cerradas para saber datos específicos y también realizar preguntas abiertas para saber cuál es la opinión del usuario.
  - c. Revisar las preguntas planteadas anteriormente y verificarlas, para así observar cuales preguntas nos ayudan a conseguir la información que deseamos.
  - d. Una vez realizadas las preguntas serán plasmadas en la ficha de cuestionario para llevar un orden y obtener la información de una manera precisa.
2. Pensar en el contexto de aplicación del cuestionario, para la cual se decidió que esta fuese de manera personal, para así tener un contacto directo con el usuario, en este punto se tuvo en cuenta lo siguiente:
  - a. Identificar todos los tipos posibles de usuarios que pueden existir en la empresa u organización.
  - b. Verificar la disponibilidad de los usuarios y fijar una fecha exacta en la que se aplicara el cuestionario.
3. Una vez realizada el cuestionario los datos recogidos a través de los cuestionarios, estos se plasmaron en un Excel.

### Software de Monitoreo de Pcs

El uso del Software de Monitoreo de Pcs fue utilizado de la siguiente manera:

Primero fue instalada en cada una de las maquinas, en la cual los usuarios interactuaron con el prototipo funcional. Realizada cada una de las actividades que se les indicó a los usuarios, el software instalado informó sobre las acciones realizadas por los mismos, como:

- a) El registro de las teclas presionadas.
- b) Capturas de pantallas.
- c) El tiempo en que se tardó en realizar determinada acción.
- d) Registro de los clics realizados y elementos seleccionados, como botones, campos de entradas de texto, consultas de ayuda, etc.

Con la información obtenida, se realizó un análisis de que tan factible fue la interacción, uso, del sistema para los usuarios.

### 3.8. Análisis estadístico e interpretación de los datos

En este proyecto de investigación se utilizó la herramienta estadística para el análisis de datos como:

1. Sumatoria también llamada como notación sigma, la cual es una operación matemática que se utiliza para calcular la sumatoria de muchos sumandos, esta operación se expresa de la siguiente manera con la letra griega sigma en mayúsculas  $\Sigma$  y se representa así:

Sea:

$i$ =es el valor inicial, llamado límite inferior

$n$ =es el valor final, llamado límite superior

Pero necesariamente debe cumplirse que:  $i \leq n$

$$\sum_{i=1}^n X_i$$



Expresión que se lee: "sumatoria de Xi, donde i toma los valores desde 1 hasta n".

- El promedio aritmético: Esta herramienta es usada para sumar todos los valores que toman la variable en el conjunto y dividir esta cantidad entre el número de elementos del conjunto. Por definición cada conjunto tiene solo un promedio, (Moya de Madrigal, 2005)

Sea:

X= variable en estudio.

N= número de elementos del conjunto.

$\mu_x$  = promedio aritmético de la variable en el conjunto total.

$\Sigma$  = símbolo de sumatoria.

$X_i = X_1, X_2, \dots, X_N$  = los valores de la variable en los "N" elementos del conjunto.

$$\mu_x = \frac{X_1 + X_2 + \dots + X_N}{N}$$

### VARIABLE INDEPENDIENTE

Factibilidad:

- Tiempo:

#### FORMULA

$$X = T_F - T_i$$

Variable	¿Qué es?	¿Cómo se obtiene?
$T_F$	Es la fecha en la que se dio por culminado el proyecto.	Se obtuvo la fecha en que se culminó el proyecto.
$T_i$	Es la fecha en que se dio por iniciado el proyecto.	Se obtuvo la fecha en que se inició el proyecto.
X	Es el tiempo en que se tomó en desarrollar el caso de estudio aplicando metodología.	Se obtuvo con la resta del tiempo final ( $T_F$ ) – el tiempo inicial ( $T_i$ ) expresado en una días.



Tabla 7: Análisis Estadístico de Tiempo  
Fuente: Elaboración propia.

2. Total de presentaciones:

FORMULA		
$X=T_1 + T_2 \dots + T_n$		
Variable	¿Qué es?	¿Cómo se obtiene?
$T_I$ :	Reunión llevada a cabo con el sponsor, para la presentación de avances del proyecto.	Se obtuvo contabilizando cada documento que se tuvo en el control de reuniones.
X	Es el número total de reuniones presentadas por cada metodología.	Se obtuvo sumando el número de documentos de control de reuniones firmados ( $T_I$ ) por el sponsor.

Tabla 8: Análisis Estadístico de Total de Presentaciones  
Fuente: Elaboración Propia

3. Total de artefactos:

FORMULA		
$X= A_i + A_i + \dots$		
Variable	¿Qué es?	¿Cómo se obtiene?
$A_I$	Es el total de artefactos que utiliza una metodología por cada etapa del ciclo de vida.	Se obtuvo sumando los artefactos realizados por una metodología en cada una de sus etapas.
X	Es el total de artefactos utilizada por una metodología.	Se obtuvo sumando todos sus artefactos de una metodología.

Tabla 9: Análisis Estadístico del Total de Artefactos  
Fuente: Elaboración Propia

**VARIABLE DEPENDIENTE**

Facilidad de Aprendizaje

1. Tiempo de Aprendizaje: Promedio del tiempo de los usuarios para aprender a utilizar una determinada función.



FORMULA

$$X = \frac{(t_1 + t_2 + \dots + t_n)}{t_n}$$

Variable	¿Qué es?	¿Cómo se obtiene?
$T_1$	Es el tiempo promedio por cada usuario para aprender una determinada función.	Se obtuvo tomando el tiempo promedio en que el usuario tardó en aprender una función.
$T_n$	Es número de personas que han sido evaluadas.	Se obtuvo contabilizando cada persona que fueron evaluada ( $t_i$ ).
$X$	Es el tiempo promedio que se tardó en aprender una determinada función.	Se obtuvo sumando todos los tiempos alcanzados por cada persona ( $t_i$ ) para realizar una función entre el número de personas evaluadas ( $n$ ).

Tabla 10: Análisis Estadístico del Tiempo de Aprendizaje  
Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

$$0 < X$$

- Mientras que  $X$  sea más cercano a 0, resulta factible.

2. Valorización aprender: Es la percepción de que tan fácil de aprender a utilizar el sistema para el usuario.

FORMULA

$$X = \frac{(A_1 + A_2 + \dots + A_n)}{A_n}$$

Variable	¿Qué es?	¿Cómo se obtiene?
$A_i$	Es la valoración del usuario bajo la escala de LIKERT.	Se obtendrá por el promedio obtenido por cada usuario bajo la escala de LIKERT.
$A_n$	Es el número de personas que han sido evaluadas bajo la escala de LIKERT.	Se obtiene sumando cada persona que participo en la evaluación ( $A_i$ ).
$X$	Es la valorización promedio de la percepción que tiene el usuario para con el sistema si es fácil de aprender.	Se obtiene sumando los promedios de cada persona ( $A_i$ ) entre el número de personas evaluadas ( $A_i$ ).

Tabla 11: Análisis Estadístico de la Valoración de Aprender  
Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

- Mientras que  $x$  se más cercano a 5 es más factible.

$$5 > X$$



3. Tiempo de operación: Promedio del tiempo de los usuarios para realizar una determinada función.

FORMULA

$$X = \frac{(t_1 + t_2 + \dots + t_n)}{t_n}$$

Variable	¿Qué es?	¿Cómo se obtiene?
$T_1$	Es el tiempo promedio por cada usuario para realizar una determinada función.	Se obtuvo tomando el tiempo promedio en que el usuario tardó en realizar una función.
$T_n$	Es número de personas que han sido evaluadas.	Se obtuvo sumando cada persona fue evaluada ( $t_i$ ).
$X$	Es el tiempo promedio que se tardó en realizar una determinada función.	Se obtiene sumando todos los tiempos alcanzados por cada persona ( $t_i$ ) para realizar una función entre el número de personas evaluadas ( $n$ ).

Tabla 12: Análisis Estadístico del Tiempo de Operación  
Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

$$0 < X$$

- Mientras que  $X$  sea más cercano a 0, resulta factible.

Comprensibilidad:

1. Comprensión de las funcionalidades.

FORMULA

$$X = \frac{(A_1 + A_{i+1} + \dots + A_{i+n})}{B_T}$$

Variable	¿Qué es?	¿Cómo se obtiene?
$A_i$	Es el total de tareas comprendidas por el usuario	Se obtuvo sumando cada tarea realizada por el usuario sin necesidad de ayuda.
$B_T$	Es el número total de tareas asignadas a los usuarios.	Se obtuvo sumando la cantidad de tareas asignadas a los usuarios.



X	Es promedio de tareas comprendidas por los usuarios	Se obtuvo el sumando las tareas comprendidas por los usuarios $A_i$ sobre el número total de tareas asignadas a los usuarios.  $B_T$
---	---	--

Tabla 13: Análisis Estadístico del Comprensión de Funcionalidades  
Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

## 2. Salida y entrada de elementos

FORMULA		
$X = \frac{(A_i + A_{i+1} + \dots + A_{i+n})}{B_T}$		
Variable	¿Qué es?	¿Cómo se obtiene?
$A_i$	Número total de elementos de entrada de datos y salida que el usuario entienda con éxito	Se obtuvo sumando cada elemento de entrada que usuario realizo correctamente.
$B_T$	Número de elementos de datos de entrada y salida disponible en la interfaz	Se obtuvo sumando los elementos total disponible en una interfaz.
X	Es promedio de elementos de entras y salidas comprendidas por los usuarios.	Se obtuvo el sumando $(A_i + A_{i+1} + \dots + A_{i+n})$ sobre $B_T$

Tabla 14: Análisis Estadístico del Salida y Entrada de Elementos  
Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

Operabilidad

Coherencia de uso

FORMULA		
$X = 1 - \frac{A}{B}$		
Variable	¿Qué es?	¿Cómo se obtiene?
$A_i$	Número total de mensaje o funciones que son inaceptables o inconsistentes por el usuario.	Se obtuvo sumando el número de observaciones o inconformidades por el usuario.
$B_T$	Número total de mensajes o funciones.	Se obtuvo el total de mensajes o funciones presentadas al usuario.
X	Que tan consistentes son los componentes	Se obtuvo de la siguiente manera:



de la interfaz de usuario.

$$X = 1 - \frac{A}{B}$$

Tabla 15: Análisis Estadístico de Coherencia en Uso  
Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

1. Corrección de errores

FORMULA		
$X = \frac{T_C}{T_S}$		
Variable	¿Qué es?	¿Cómo se obtiene?
$T_C$	Número total de errores corregidos satisfactoriamente por el usuario	Se obtuvo sumando el número de errores que corrigió el usuario.
$T_S$	Número total de errores presentados al usuario.	Se obtuvo sumando el total de errores que se le presento al usuario.
$X$	Es el promedio de la cantidad de errores corregidos por el usuario, mientras más cerca al 0 es mejor.	Se obtuvo de la siguiente manera: $X = \frac{T_C}{T_S}$

Tabla 16: Análisis Estadístico de Corrección de errores  
Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

2. Capacidad de entender los mensajes de alerta

FORMULA		
$X = \frac{T_C + T_{C+1} + T_{C+n}}{T_S}$		
Variable	¿Qué es?	¿Cómo se obtiene?
$T_C$	Número total mensajes de alertas que se le presenta al usuario	Se obtuvo sumando el número de mensajes de alerta que se le presento al usuario
$T_S$	Número total de alertas existentes en el sistema	Se obtuvo sumando el total de mensajes de alertas existentes en el sistema



X	Es el promedio de la cantidad de mensajes de alerta que se le presento al usuario.	Se obtuvo de la siguiente manera: $X = \frac{T_C + T_{C+1} + T_{C+n}}{T_S}$
---	--	--

Tabla 17: Análisis Estadístico de la Capacidad de Entender los mensajes de alerta  
Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

Atractividad

1. Interacción atractiva

FORMULA		
$X = \frac{(A_i + A_i + \dots + A_n)}{A_n}$		
Variable	¿Qué es?	¿Cómo se obtiene?
$A_i$	Es la valoración asignada por el usuario bajo la escala de LIKERT, para saber de qué tan atractivo es la interfaz del usuario.	Promedio obtenido por cada usuario bajo la escala de LIKERT.
$A_n$	Es el número de personas que han sido evaluadas bajo la escala de LIKERT.	Se obtiene contabilizando cada persona que fue evaluada ( $A_i$ ).
X	Es la valorización promedio de la percepción que tiene el usuario para con el sistema si es fácil de aprender.	Se obtiene sumando los promedios de cada persona ( $A_i$ ) entre el número de personas evaluadas ( $A_i$ ).

Tabla 18: Análisis Estadístico de Interacción Atractiva. Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

Cumplimiento de la usabilidad

1. Cantidad de ítems de especificaciones.

FORMULA		
$X = 1 - \frac{W}{Y}$		
Variable	¿Qué es?	¿Cómo se obtiene?
W	Es el número de factores de calidad que se han tomado en cuenta para evaluar la usabilidad externa en el producto software.	Se obtuvo sumando los ítems que se han considerado en este proyecto.



Y	Es el número de factores de calidad existentes para evaluar la usabilidad externa en el producto software.	Se obtuvo sumando el número total de ítems existentes para medir la usabilidad.
X	Que tanto ha completado el software bajo los estándares de usabilidad.	Se obtuvo de la siguiente manera: $X=1-\frac{W}{Y}$

Tabla 19: Análisis Estadístico de Cumplimiento de la Usabilidad Fuente: Elaboración Propia, basado en la ISO/IEC 9126-2

### 3.9. Principios éticos

Los criterios éticos que están presentes en esta investigación, va acordes a 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”, (Colegio de Ingenieros del Perú, 2012) 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.

En el presente proyecto de investigación se expresará lo más claro y conciso su contenido con el fin de generar un aporte al desarrollo de software, al estar disponible para el público en general los resultados obtenidos en esta investigación.

Criterios Éticos	
<b>PRINCIPIOS GENERALES</b>	Los ingenieros están al servicio de la sociedad. Por ello se ven en la obligación de contribuir al bienestar humano, dando importancia a la seguridad y adecuada utilización de los recursos en el desempeño de sus tareas profesionales
<b>DE LA RELACIÓN CON EL PÚBLICO</b>	Los ingenieros serán objetivos y veraces en sus informes, declaraciones o testimonios profesionales
	Los ingenieros, al explicar su trabajo lo realizaran de manera seria y modestamente, tratando de no promover sus propios intereses.





	Los ingenieros no participaran en la disseminación de conceptos falsos, injustos o exagerados acerca de la ingeniería.
<b>DEL EJERCICIO PROFESIONAL</b>	Los ingenieros podrán hacer promoción de sus servicios profesionales sólo cuando ella no contenga lenguaje jactancioso o engañoso o en cualquier forma denigrante para la dignidad de la profesión
	Los ingenieros podrán describir y hacer conocer su experiencia, mediante publicaciones realistas.

### 3.10. Criterios de rigor científico

Criterios		Acciones o Estrategias
<b>VALIDEZ</b>	En la Operacionalización de las variables y la definición de las dimensiones están escritas de una manera clara y concisa, teniendo en cuenta el objetivo general de la investigación. Estas dimensiones se evaluarán según los indicadores establecidos y las técnicas de recolección de datos definidas, de esa manera poder validar los resultados obtenidos.	
<b>REPLICABILIDAD</b>	Los resultados hallados en la presente investigación no se pueden contradecir si es que se vuelve a repetir, porque los métodos a aplicar están establecidos, por los autores de los artículos de investigación consultados, lo que puede diferenciar en los resultados si se toma un enfoque diferente.	



## CAPÍTULO IV: RESULTADOS DE LA INVESTIGACIÓN

En este apartado se mostrará el análisis realizado de la integración de las directrices para la captura de requerimientos de usabilidad USEP's con las metodologías de desarrollo web NDT y UWE, con el fin de seleccionar la metodología más adecuada, para luego ser desarrollada y así poder evaluar el producto software bajo los estándares de la ISO/IEC 9126-2, para medir la usabilidad funcional percibida por el usuario.

### 4.1 Resultados de la Evaluación:

Primero evaluaremos la integración de las directrices USEP's con las metodologías propuestas bajo los siguientes indicadores: cantidad de artefactos generados que incluye a la usabilidad, tiempo que tomo desarrollar cada metodología de acuerdo a su alcance, así como también el número de presentaciones que se realizó por cada metodología estudiada.

#### 4.1.1 Factibilidad:

##### 4.1.1.1 Tiempo: Tiempo de desarrollo

Se procedió a evaluar las metodologías NDT y UWE bajo el tiempo que tomo el realizar el modelado del sistema con la integración de las directrices, expresado en horas, teniendo en consideración que para el desarrollo de esta investigación se trabajó 6 horas diarias.

	Etapa	Fecha inicio	Fecha fin	Días	Horas trabajadas	Total de horas trabajadas
NDT	Captura de Requerimientos	15/08/2016	09/09/2016	25	6	150
	Definición de Requerimientos	10/09/2016	30/09/2016	20	6	120
	Análisis	01/10/2016	11/10/2016	10	6	60
	Diseño	12/10/2016	18/10/2016	6	6	36

Tabla 20 Tiempo de desarrollo de la metodología NDT.



UWE	Etapa	Fecha inicio	Fecha fin	Días	Horas trabajadas	Total de horas trabajadas
	Captura de Requerimientos	15/08/2016	09/09/2016	25	6	150
	Definición de Requerimientos	10/09/2016	19/09/2016	9	6	54
	Análisis	20/09/2016	04/10/2016	14	6	84
	Diseño	05/10/2016	20/10/2016	15	6	90

Tabla 21 Tiempo de desarrollo de la metodología UWE

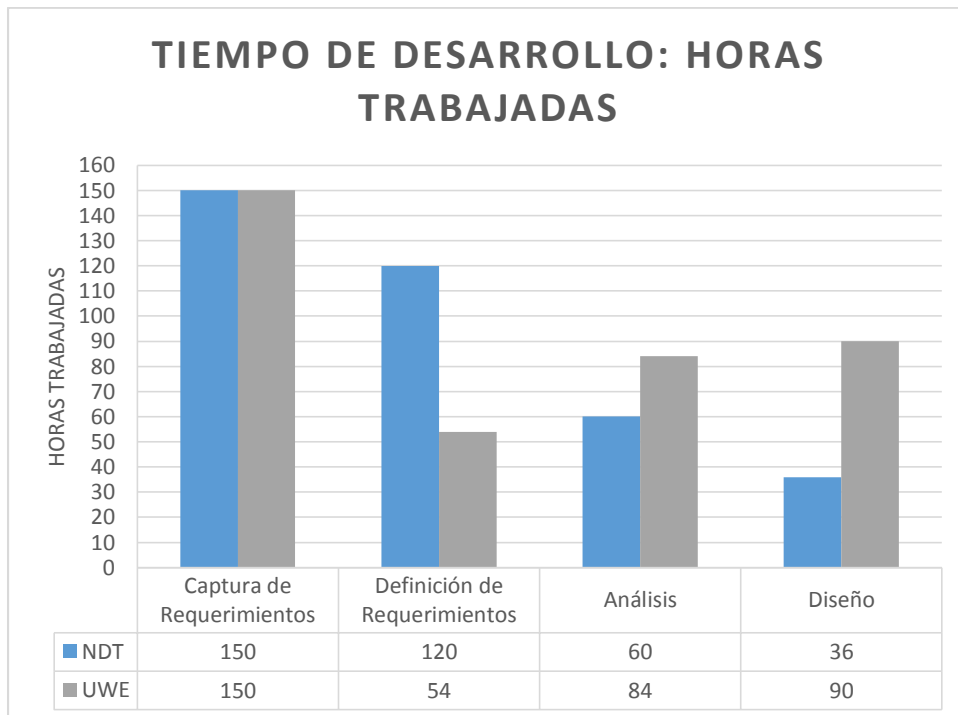


Gráfico 1 Comparación del tiempo de desarrollo de las metodologías.

Según el análisis realizado en el Gráfico 1 Comparación del tiempo de desarrollo de las metodologías. Se obtuvo:

- a) La NDT invierte más tiempo en la fase de definición de requerimientos
- b) La UWE invierte más tiempo en la fase de análisis y diseño dado que la NDT le hace más énfasis en la etapa de requerimientos y realiza las siguientes fases a través de transformaciones facilitado el modelado del sistema en sus siguientes fases.



c) La NDT es la metodología que requirió de menos días para terminar el proceso de captura de requerimientos, definición de requerimientos, análisis y diseño alcanzando un total de 61 días, que evaluado en horas hace referencia a un total de 366 horas frente al UWE, para la cual se invirtió un total de 63 días, haciendo referencia de 378 horas.

4.1.1.2 Total de presentaciones: Número de presentaciones para ser aceptado por el usuario.

Esta evaluación se realizó bajo la perspectiva de contabilizar el número de reuniones retro-alimentadoras pactadas con los principales Stakeholder donde el principal motivo de la reunión era mostrar los avances del sistema.

Fases del proyecto	Cantidad de Reuniones	
	NDT	UWE
Captura de Requerimientos	3	3
Definición de Requerimientos	3	2
Análisis	1	1
Diseño	2	2
<b>Total</b>	<b>9</b>	<b>8</b>

Tabla 22 Total de presentaciones realizadas por cada metodología.

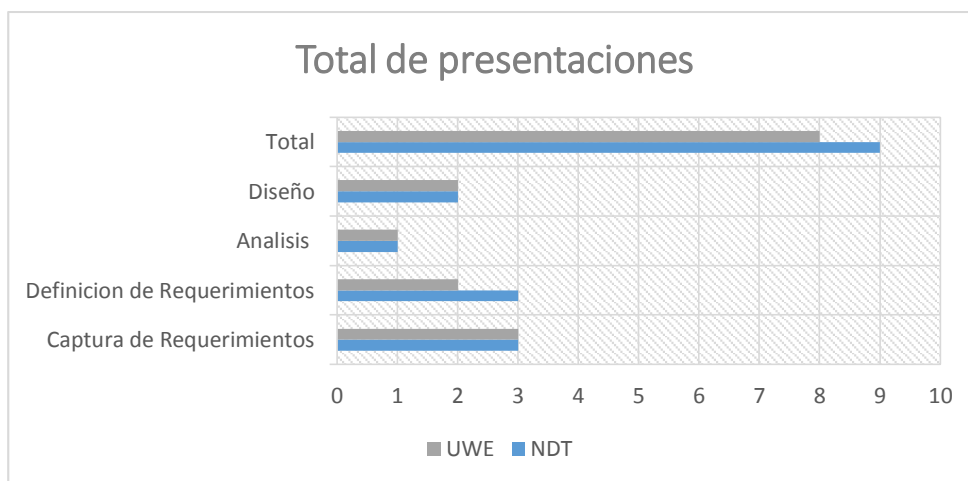


Gráfico 2 Total de presentaciones realizadas por cada metodología.



Según el análisis realizado en el Gráfico 2 Total de presentaciones realizadas por cada metodología. Se obtuvo:

- a) La NDT realizó 9 presentaciones con el stakeholders para que sus modelos realizados obtuvieran el visto bueno, mientras que la UWE requirió de 8 presentaciones para que sus modelos sean aceptados.
- b) La NDT requirió de una entrevista más con el Stakeholders en la etapa de definición de requerimientos.

En el Grafico 3 se hace énfasis en el tiempo de duración de las reuniones convocadas para el desarrollo de la solución del caso de estudio por cada metodología de desarrollo Web, obteniéndose un total de 14 h 30 min en tiempo de duración de las reuniones realizadas bajo en desarrollo con la metodología NDT a comparación de la UWE que acumuló un total de 14h 15 min.

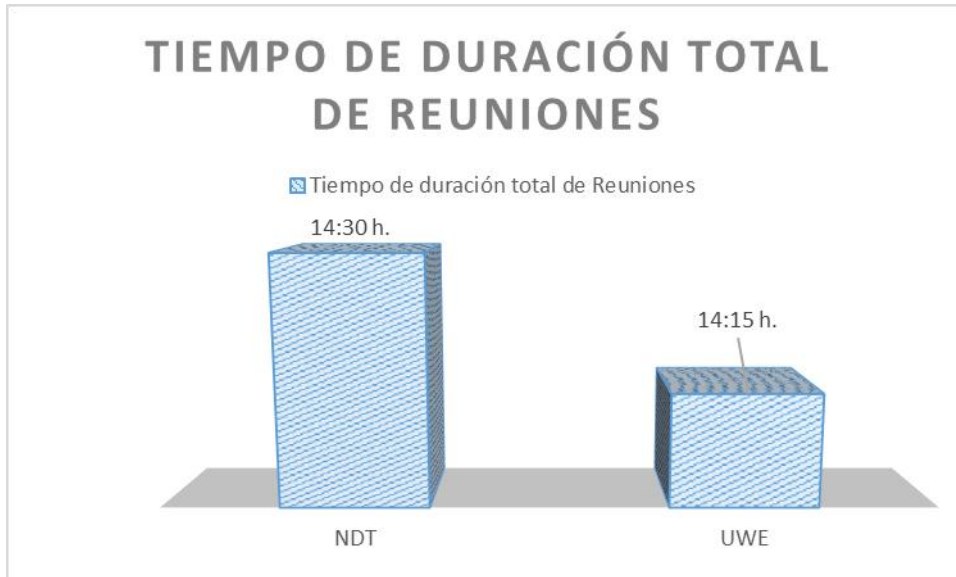


Gráfico 3 Tiempo de duración total de reuniones convocadas por cada una de las metodologías.

#### 4.1.1.3 Total de artefactos: Cantidad de artefactos que incluyen a la usabilidad.

Se evaluó la cantidad de artefactos generados por la metodología NDT y UWE, así como también se contabilizó el número de artefactos por cada metodología que plasma a un requerimiento funcional de usabilidad.

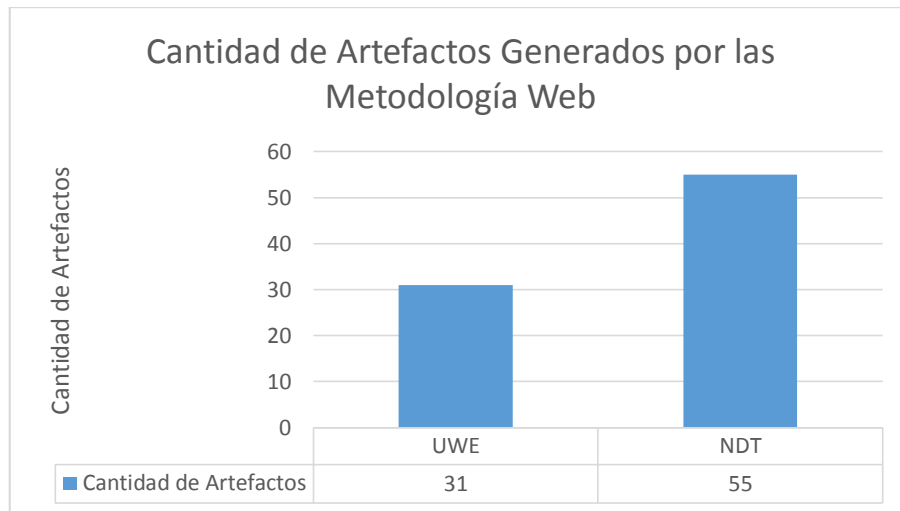


Gráfico 4 Cantidad de artefactos generados por cada una de las metodologías Web.

El Gráfico 4 muestra la cantidad de artefactos generados por cada una de las metodologías de desarrollo Web, en la cual se obtuvo que metodología con más artefactos generados es la NDT con una cantidad de 55 artefactos frente a 31 artefactos generados por la UWE.

El siguiente gráfico muestra la cantidad de artefactos por cada metodología que plasma un requisito funcional de usabilidad.

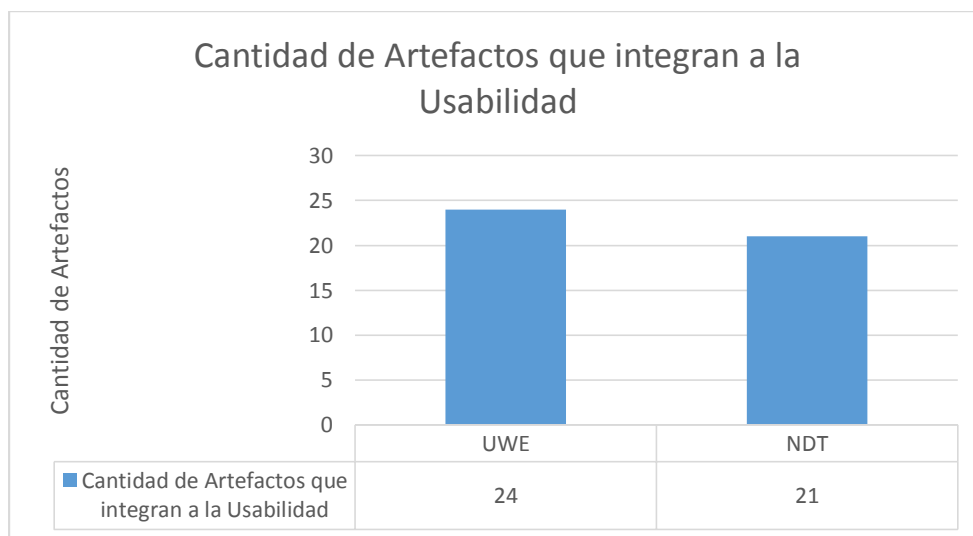


Gráfico 5 Cantidad de Artefactos que integran a la Usabilidad.

Una vez realizado el análisis de la cantidad de artefactos generado por cada metodología, se procedió a analizar, cuanto de estos artefactos generados incluye a un mecanismo de usabilidad. Obteniendo como resultado que el 77,43% de los artefactos generados por la UWE plasman al menos un requisito funcional de usabilidad mientras que el 38,18% de los artefactos generados por la NDT plasma al menos un requisito funcional de usabilidad.

Consecuente al análisis anterior se procedió a analizar, que cantidad de mecanismo de usabilidad son tratados o integrados dentro del modelado realizado por cada metodología.

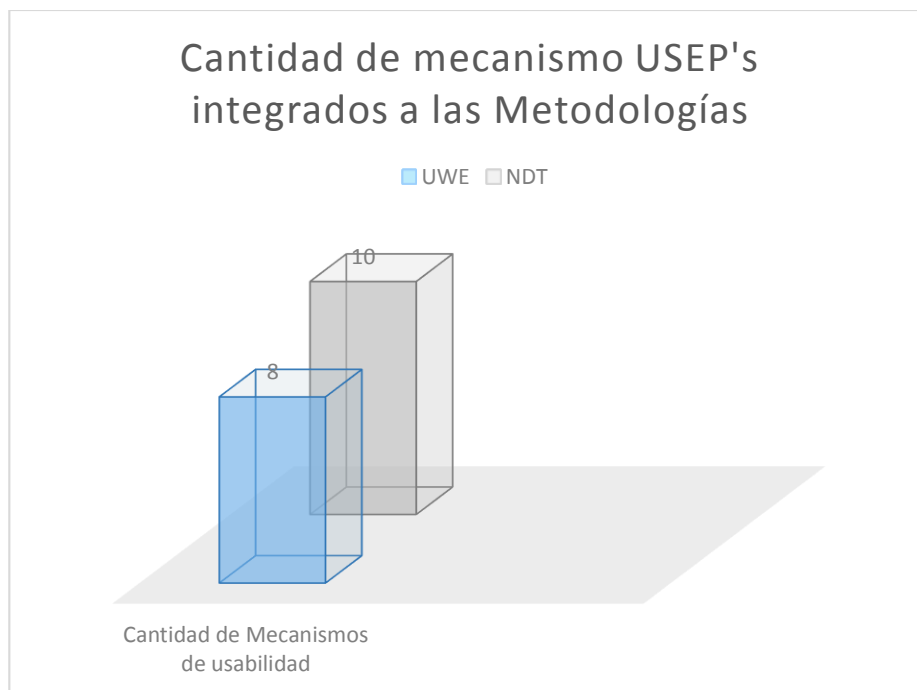


Gráfico 6 Cantidad de mecanismo USEP's integrados a las Metodologías.

Teniendo como resultado que la NDT a pesar de tener menos modelos que incluyen requerimientos funcionales de usabilidad, plasma 10/10 mecanismos de usabilidad en todo su proceso de modelado del sistema, mientras que la UWE cuenta una gran cantidad de modelos que incluyen al requerimiento funcional de usabilidad, pero incluye 8 /10 en todo su proceso de modelado del sistema.

MECANISMOS DE USABILIDAD	Captura de Requerimientos		Definición de Requerimientos		Análisis del Sistema		Diseño del Sistema	
	UWE	NDT	UWE	NDT	UWE	NDT	UWE	NDT
System status Feedback	x	x	x	x		x		
Interaction Feedback	x	x	x			x	x	
Progress Feedback	x	x				x		
Warning	x	x	x	x	x	x	x	x
Step by step	x	x				x		
Structured text entry	x	x		x	x	x		x
Undo y cancel: abort operation	x	x	x	x		x		
Help	x	x		x	x	x	x	x
User Profile	x	x		x	x	x	x	x

Tabla 23 Análisis de las características funcional de Usabilidad, en cada una de las etapas de modelado del sistema de las metodologías en estudio.

En la tabla 23 se puede observar a más detalle las características de usabilidad, tratadas en cada etapa del modelado del sistema por cada una de las metodologías.

Una vez desarrollado un prototipo funcional con la metodología de desarrollo web seleccionada, NDT (Navigational Development Techniques), la cual podemos observar en el Capítulo V se procedió a evaluar la calidad de prototipo funcional en cuanto a la usabilidad de acuerdo a la ISO/IEC 9126-2. La evaluación se realizó con la participación de 10 docentes y 10 alumnos de la institución educativa particular “Santa María de la Paz”, a quienes también se les aplicó la encuesta SUS (Escala de Usabilidad del Sistema), conformada por diez ítems a evaluar, que otorga una visión global subjetiva de usabilidad (Finstad, 4 Agosto 2016).

#### 4.1.2 Facilidad de Aprendizaje

##### 4.1.2.1 Tiempo de Aprendizaje:

En esta investigación se evaluó el tiempo que tarda un determinado usuario en aprender la funcionalidad del prototipo funcional desarrollado como parte del caso de estudio propuesto. La evaluación





se desarrolló promediando el tiempo que tardo cada usuario en entender el funcionamiento del prototipo funcional como podemos observar el Gráfico 7.

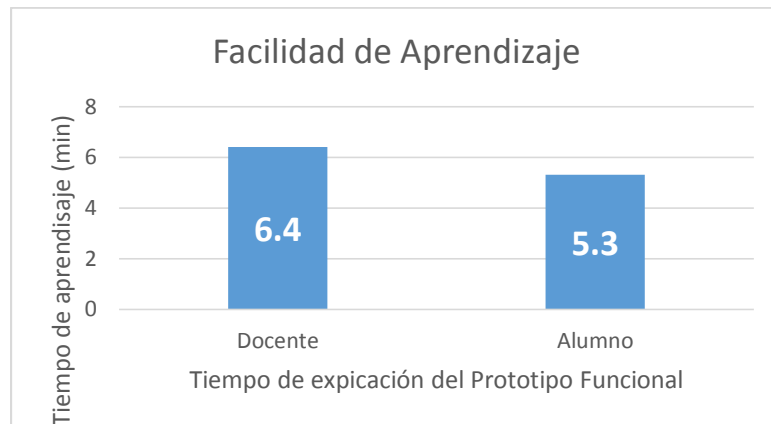


Gráfico 7 Tiempo de Aprendizaje

De acuerdo a los datos obtenidos en la investigación con respecto a la facilidad de aprendizaje comparado con ISO/IEC 9126-2 indica que mientras que el tiempo obtenido sea más cercano a 0 el software es fácil de aprender.

Para ello se evaluó el tiempo promedio que les llevo aprender a los docentes a crear un examen teniendo como resultado 6.4 min, por otro lado se evaluó el tiempo promedio de los alumnos en aprender a realizar un examen online obteniendo un promedio de 5.3 min.

#### 4.1.2.2 Valorización de aprender:

Para poder tener la valorización subjetiva con respecto a la facilidad de aprendizaje que tiene el usuario acerca de sistema se tomó como referencia algunas de las preguntas realizadas del cuestionario SUS la cual podemos observar en la Tabla 24.

Acrónimo	Pregunta
PR3	Pienso que el sistema es fácil de usar.
PR6	He encontrado demasiada inconsistencia en este sistema.
PR9	Me he sentido muy seguro haciendo uso del sistema.

Tabla 24 Preguntas para la valorización de aprendizaje



Estas preguntas fueron respondidas por ambos usuarios la cual el resultado se obtuvo promediando el puntaje obtenido por cada uno de los encuestados como podemos observar en la Tabla 25.

Numero	PR3	PR6	PR9	Puntuación
1	3	3	5	3.67
2	4	3	4	3.67
3	3	3	4	3.33
4	3	3	4	3.33
5	5	4	5	4.67
6	5	4	5	4.67
7	5	3	5	4.33
8	5	3	5	4.33
9	5	3	5	4.33
10	5	3	5	4.33
11	5	3	5	4.33
12	5	3	5	4.33
13	5	3	5	4.33
14	4	4	5	4.33
15	4	3	4	3.67
16	4	2	4	3.33
17	5	2	4	3.67
18	3	3	4	3.33
19	5	3	5	4.33
20	5	4	4	4.33
<b>Promedio</b>				<b>4.03</b>

Tabla 25 Valorización de la facilidad de aprendizaje del sistema.

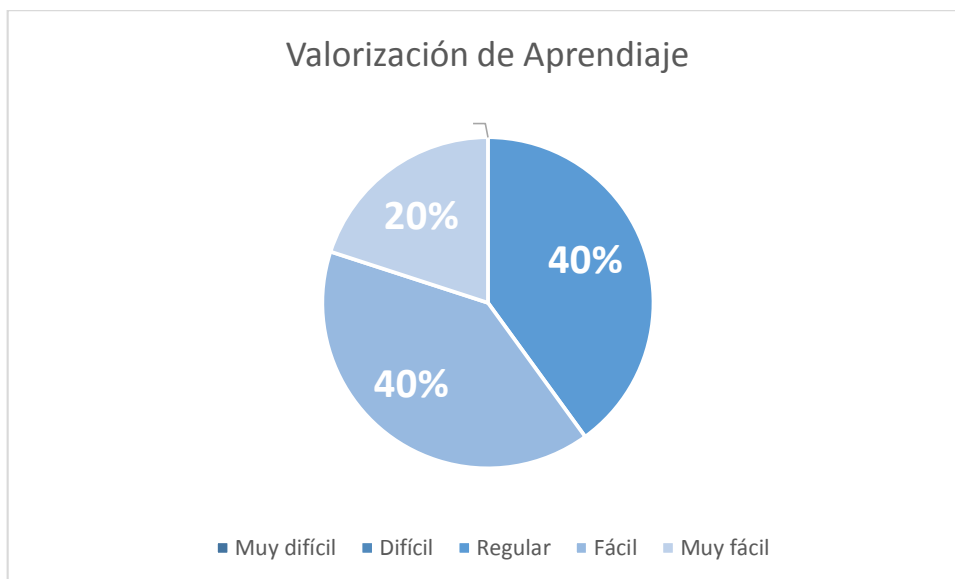


Gráfico 8 Análisis de la valorización de la facilidad de aprendizaje del sistema.



Como podemos observaren el Gráfico 8 de acuerdo a los parámetros establecidos según la escala de Likert se obtuvo que el 20% de los encuestados manifestaron que el sistema es muy fácil de aprender, 40% indico que el sistema es fácil de aprender mientras que otro grupo que representa 40% de los encuestados concuerdan que el sistema es de un aprendizaje regular.

#### 4.1.2.3 Tiempo de operación:

En esta investigación se evaluó el tiempo de operación que tarda un determinado usuario en concretar determinadas tareas asignadas en el prototipo funcional desarrollado como parte del caso de estudio propuesto.

La evaluación se desarrolló promediando el tiempo que tardó cada usuario en desarrollar cada actividad asignada.

Para el usuario docente se le asigno las actividades de Logeo, crear examen, listar alumnos para editar su contraseña y personalizar su espacio en la aplicación se obtuvo el siguiente resultado que se puede observar en el Grafico 9:

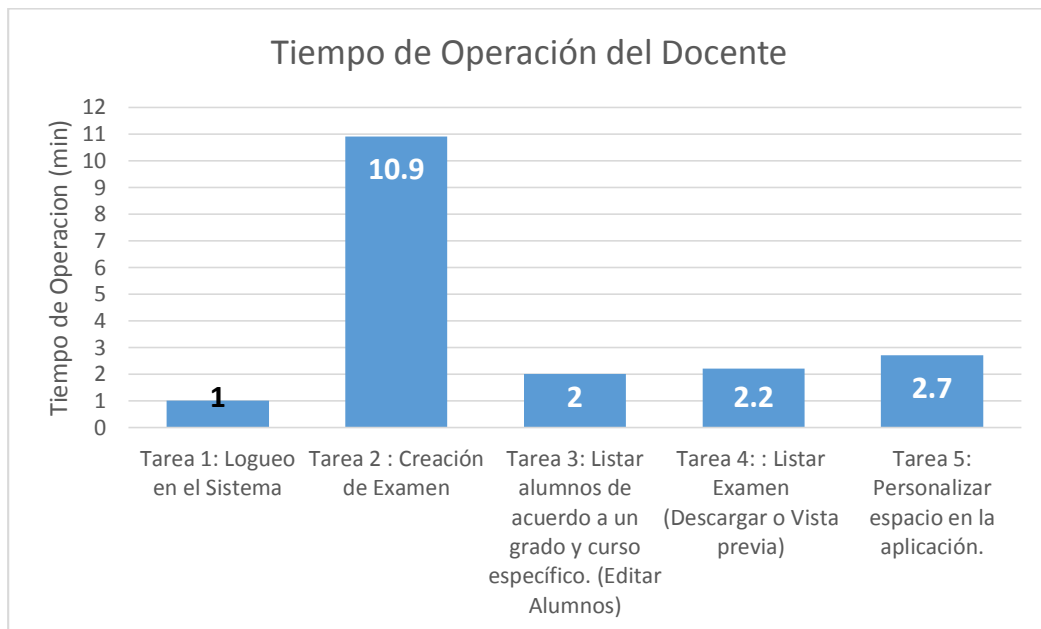


Gráfico 9 Tiempo de Operación de las actividades asignadas al docente.



Para el análisis del Grafico 9 cabe resaltar que para fines de esta evaluación se les indico a los usuarios docentes que para realizar la tarea de creación de exámenes debía registrar 5 preguntas con 4 alternativas por cada una de ella. Los resultados que se obtuvo fue que el tiempo promedio que tarda un docente en Logearse es de 1 minuto, la creación de examen tomando en cuenta el punto anterior es de 10.9 minutos, para el listado de alumnos de acuerdo a un determinado grado y curso es de 2 min, para el listado de exámenes es de un promedio de 2.2 minutos y para la personalización del espacio en la aplicación es de 2.7 minutos.

Para el usuario alumno se le asigno realizar las actividades de logeo, listar exámenes de acuerdo a un determinado curso, realizar un reporte de notas, realizar un determinado examen y personalizar el espacio de la aplicación.

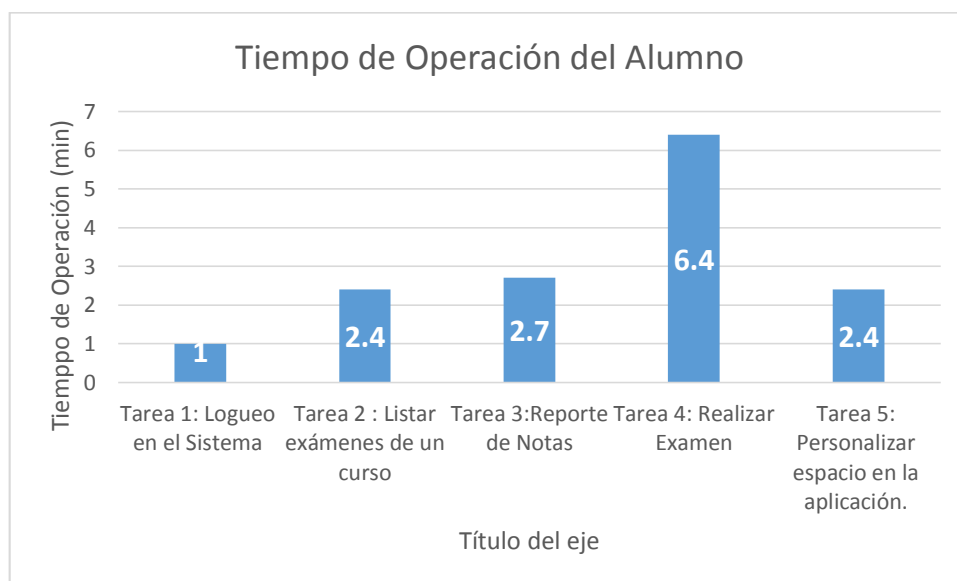


Gráfico 10 Tiempo de operación de las actividades asignadas al alumno

De acuerdo al Grafico 10 se obtuvo como resultado que el tiempo promedio para Logearse en el sistema es de 1 min, listar exámenes de un determinado curso es de 2.4 min, el tiempo promedio para realizar un reporte de notas es de 2.7 min, el tiempo para realizar un examen de 5 preguntas es de 6.4 min y para la personalización del espacio de la aplicación es de 2.4 min.



### 4.1.3 Comprensibilidad

#### 4.1.3.1 Comprensión de Funcionalidades:

En este ítem se tuvo como objetivo evaluar el número de actividades culminadas con éxito de acuerdo al número de actividades asignadas al usuario. Para el procesamiento y el análisis de los datos obtenidos se realizó el cálculo propuesto en el capítulo III en el punto de Análisis estadístico e interpretación de datos.

Realizado el cálculo anterior podemos observar los resultados obtenidos en el Gráfico 11.

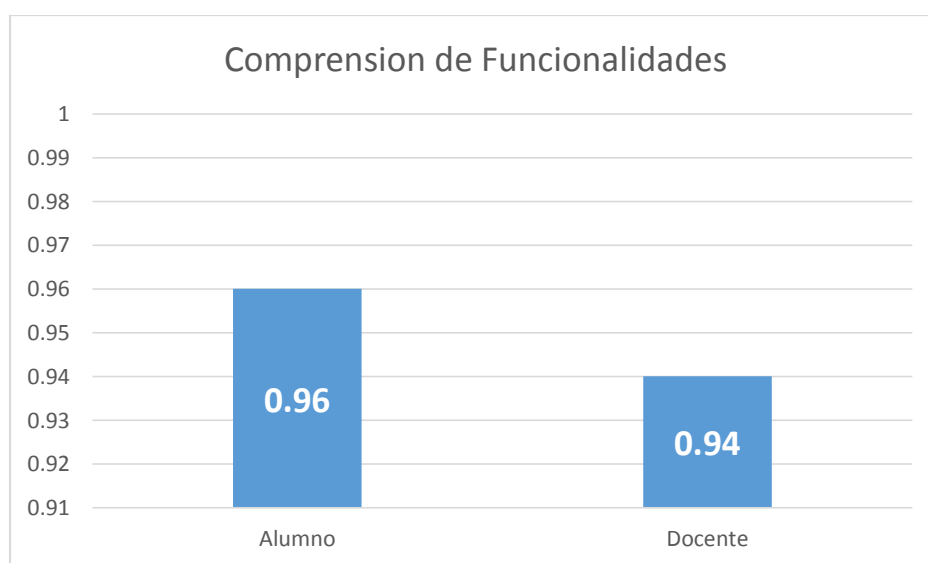


Gráfico 11 Comprensión de Funcionalidades de acuerdo al tipo de usuario.

De acuerdo a la ISO/IEC 9126-2 mientras más aproximado sea el resultado obtenido a 1 este resulta factible, la cual representa al 100%. De acuerdo a la interpretación de datos anterior, se obtuvo que el 96% de las actividades asignadas a los alumnos fueron concluidas satisfactoriamente y que 94% de las actividades asignadas a los docentes concluyeron con éxito.

#### 4.1.3.2 Salida y Entrada de elementos:

En este ítem se tuvo como objetivo evaluar el número de elementos ingresados correctamente por el usuario sobre el número de elementos presente en la interfaz de la actividad designada al usuario.

En el Gráfico 12 se presentan los resultados obtenidos referentes al grado de comprensibilidad percibida por los usuarios de acuerdo a las actividades asignadas.

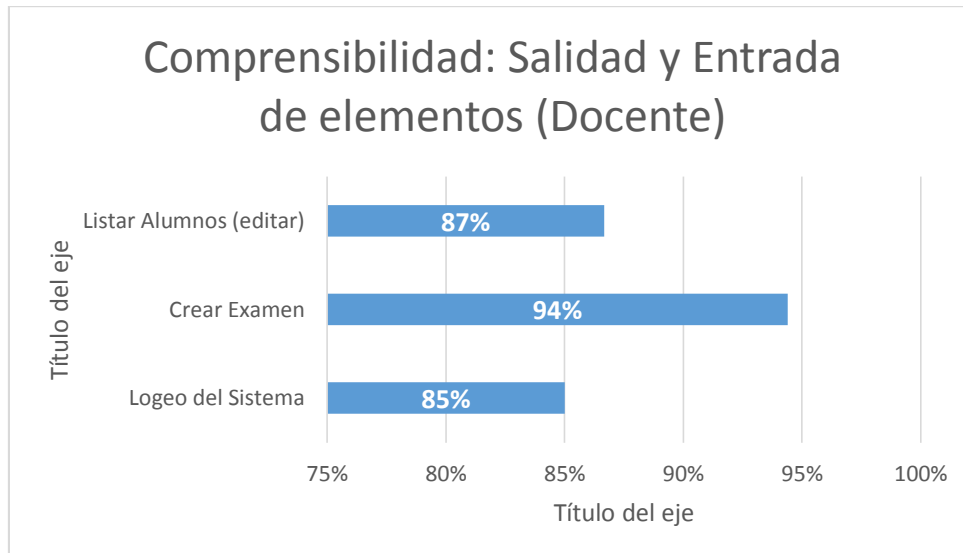


Gráfico 12 Salida y Entrada de elementos de las actividades asignadas a los docentes.

De acuerdo a la ISO/IEC 9126-2 mientras más aproximado sea el resultado obtenido a 1 este resulta factible, la cual representa al 100%. De acuerdo a la interpretación de datos anterior, se obtuvo que el 85% de los elementos GUI presentados en el Logeo del sistema fueron comprendidos correctamente, mientras que el 94% de los elementos GUI presentados para la creación de exámenes fueron comprendidos correctamente y para el listado de alumnos determinado un grado y sección para editar un alumno fue comprendió un 87%.

En el Gráfico 13 se presentan los resultados obtenidos referentes al grado comprensibilidad de acuerdo a las actividades asignadas a los alumnos.

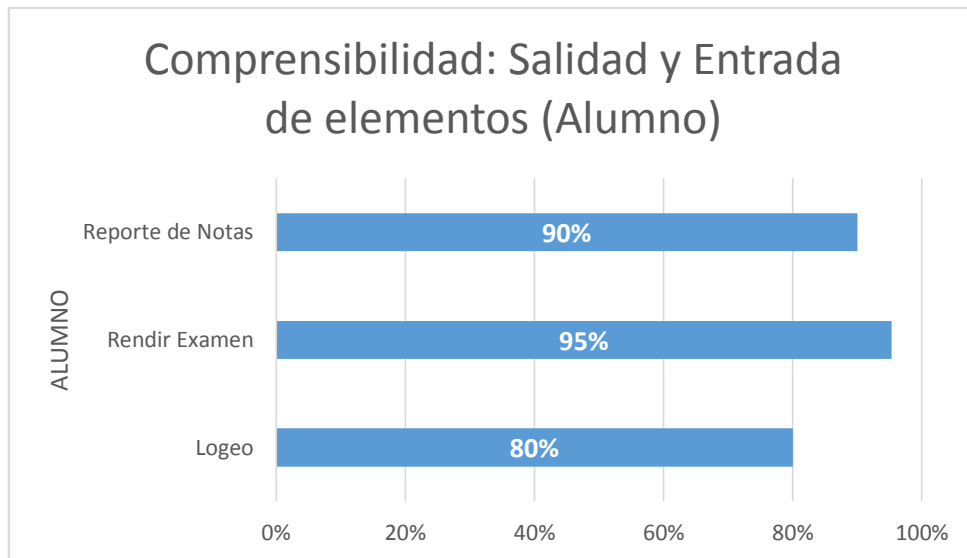


Gráfico 133 Salida y Entrada de elementos de las actividades asignadas a los alumnos.

De acuerdo a la ISO/IEC 9126-2 mientras más aproximado sea el resultado obtenido a 1 este resulta factible, la cual representa al 100%. De acuerdo a la interpretación de datos anterior, se obtuvo que el 80% de los elementos GUI presentados en el Logeo del sistema, a los alumnos, fueron comprendidos correctamente, mientras que el 100% de los elementos GUI presentados para la rendición de exámenes fueron comprendidos correctamente y para el reporte de notas para un determinado curso se comprendió un 90%, de los elementos GUI.

#### 4.1.4 Operabilidad

En este ítem se tuvo como objetivo evaluar la capacidad de los usuarios en operar el sistema por la cual se evaluó 3 sub-ítem que son la coherencia de uso, corrección de errores y la capacidad que tiene el usuario en entender los mensajes de alerta.

Los resultados que se obtuvo se presentan en el Gráfico 14.

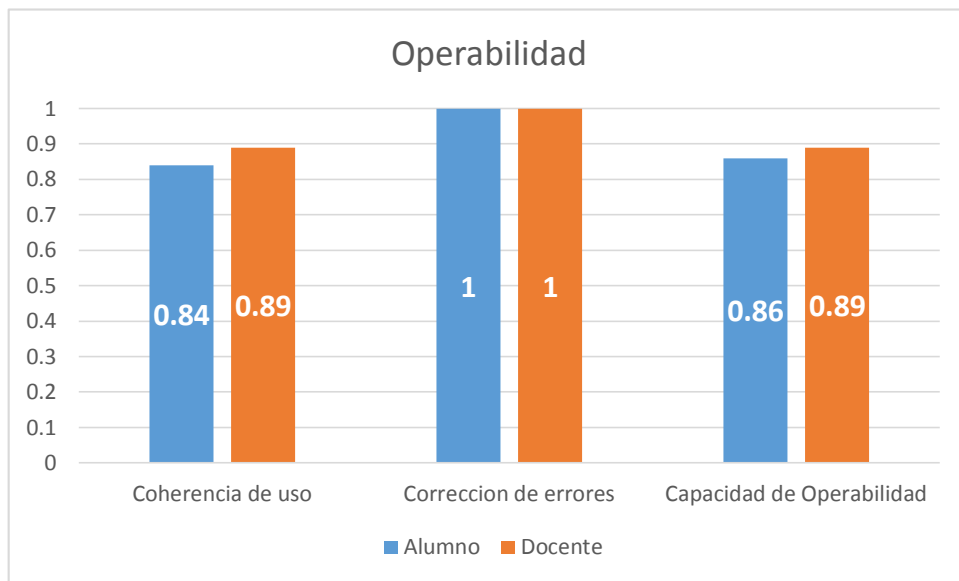


Gráfico 144 Operatividad de los usuarios en el sistema.

De acuerdo a la ISO/IEC 9126-2 mientras más aproximado sea el resultado obtenido a 1 este resulta factible, la cual representa al 100%. De acuerdo a la interpretación de datos anterior, se obtuvo que con relación a la coherencia de uso los alumnos obtuvieron un 84% de factibilidad mientras que los docentes obtuvieron un 89% de factibilidad en el manejo del sistema. Por otro lado también se analizó la cantidad de errores corregidos por los usuarios la cual se obtuvo que el 100% de los erros presentados a los alumnos fueron corregidos satisfactoriamente mientras que el 100% de los errores cometidos por los docentes fueron corregidos con éxito, la cual indica la factibilidad de los mensajes presentados al usuario para la corrección de errores. Y por último se analizó la capacidad de Operabilidad en las que los alumnos obtuvieron un 86% de su operatividad fue factible mientras que el 89%de la operatividad de los alumnos fue factible.

#### 4.1.5 Atractividad

##### 4.1.5.1 Interacción atractiva

Para poder tener la valorización subjetiva con respecto a la atractividad que tiene el usuario acerca de sistema se tomó como referencia algunas de las preguntas realizadas del cuestionario SUS la cual podemos observar en la Tabla 26.





Acrónimo	Pregunta
<b>PR1</b>	Creo que me gustaría utilizar este sistema frecuentemente
<b>PR3</b>	Pienso que el sistema es fácil de usar
<b>PR7</b>	Creo que la mayoría de la gente aprendería a hacer uso del sistema rápidamente

Tabla 26 Preguntas para obtener el nivel de atraktividad del sistema.

Estas preguntas fueron respondidas por ambos usuarios la cual el resultado se obtuvo promediando el puntaje obtenido por cada uno de los encuestados como podemos observar en la Tabla 27.

Numero	PR1	PR3	PR7	Escala Likert
1	5	3	5	4.3
2	5	3	5	4.3
3	5	3	5	4.3
4	5	3	5	4.3
5	5	5	5	5.0
6	5	5	5	5.0
7	4	5	5	4.7
8	4	5	5	4.7
9	5	5	5	5.0
10	4	5	3	4.0
11	4	5	3	4.0
12	2	5	5	4.0
13	4	5	3	4.0
14	5	4	5	4.7
15	4	4	3	3.7
16	4	4	4	4.0
17	5	5	4	4.7
18	5	1	5	3.7
19	5	5	5	5.0
20	4	5	5	4.7
Promedio				4.4

Tabla 27: Valorización de la facilidad de atraktividad del sistema.



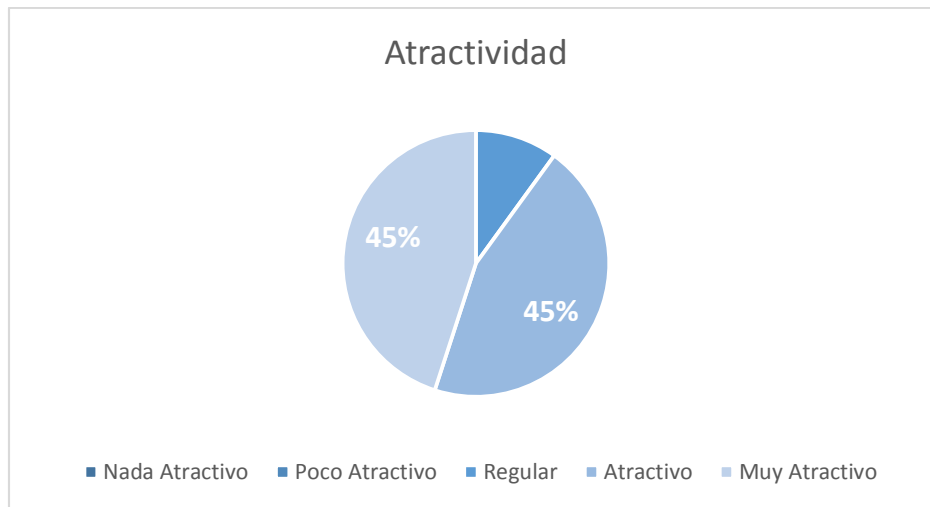


Gráfico 15 Análisis de la valorización de la atractividad del sistema

Como podemos observar en el Gráfico 15 de acuerdo a los parámetros establecidos según la escala de Likert se obtuvo que el 45% de los encuestados manifestaron que el sistema es muy atractivo para su uso, 45% indica que el sistema es atractivo para su uso mientras que otro grupo que representa 10% de los encuestados concuerdan que el sistema es de una atractividad regular.

#### 4.1.6 Cumplimiento de usabilidad

##### 4.1.6.1 Cantidad de métricas de usabilidad

Para analizar la cantidad de métricas de usabilidad evaluadas en esta investigación, contabilizo el número de métricas que se tuvo en cuenta para evaluar el prototipo funcional sobre el número de métricas que hace referencia la ISO/IEC 9126-2 con respecto a la usabilidad.

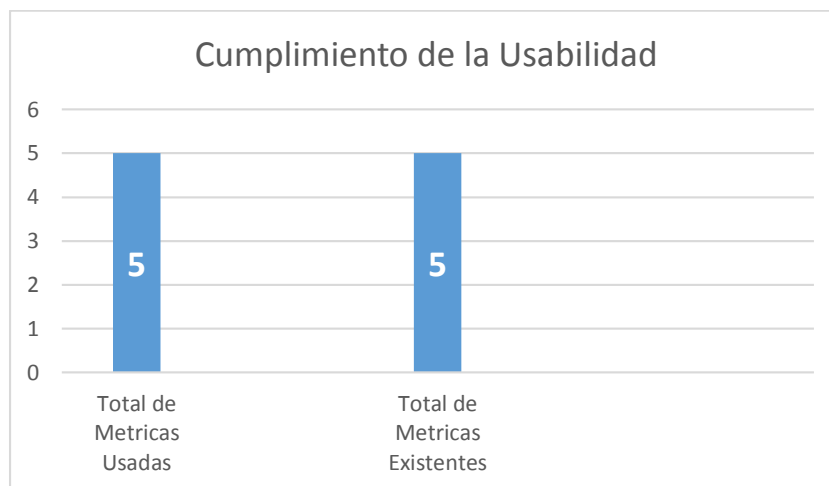


Gráfico 16: Comparación del cumplimiento de usabilidad

En el Gráfico 16 podemos observar que se evaluaron las 5 métricas propuestas por la ISO/IEC 9126-2 para medir la calidad de un producto software en términos de usabilidad.

Como parte de esta investigación se aplicó el cuestionario SUS para medir la usabilidad global subjetiva del producto software generado por la metodología de desarrollo web NDT (Navigational Development Techniques). Para ello el cuestionario SUS toma en cuenta las preguntas que se muestra en la Tabla 28.

Acrónimo	Pregunta
PR1	Creo que me gustaría utilizar este sistema frecuentemente.
PR2	Encuentro este sistema innecesariamente complejo.
PR3	Pienso que el sistema es fácil de usar.
PR4	Creo que necesitaría soporte técnico para hacer uso del sistema
PR5	Encuentro las diversas funciones del sistema bastante bien integradas.
PR6	He encontrado demasiada inconsistencia en este sistema
PR7	Creo que la mayoría de la gente aprendería a hacer uso del sistema rápidamente.
PR8	He encontrado el sistema bastante incómodo para usar
PR9	Me he sentido muy seguro haciendo uso del sistema
PR10	Necesitaría aprender un montón de cosas antes de poder manejar el sistema

Tabla 28 Preguntas del Cuestionario SUS



Estas preguntas fueron respondidas por ambos usuarios la cual el resultado se obtuvo promediando el puntaje obtenido por cada uno de los encuestados como podemos observar en la Tabla 29.

# Encuestados	PR1	PR2	PR3	PR4	PR5	PR6	PR7	PR8	PR9	PR10	Puntaje SUS
D1	5	3	3	1	5	3	5	2	5	2	80
D2	5	3	3	1	5	3	5	1	4	1	82.5
D3	5	1	3	1	5	3	5	1	4	1	87.5
D4	5	1	3	1	5	3	5	1	4	1	87.5
D5	5	1	5	1	3	4	5	1	5	1	87.5
D6	5	1	5	1	1	1	5	1	5	1	90
D7	4	1	5	2	1	3	5	1	5	1	85
D8	4	1	5	3	1	3	5	1	5	1	82.5
D9	5	1	5	2	1	3	5	1	5	1	87.5
D10	4	2	5	3	3	3	3	2	5	1	72.5
A1	4	1	5	3	3	3	3	2	5	1	75
A2	2	1	5	3	1	3	5	1	5	1	72.5
A3	4	2	5	3	3	3	3	2	5	1	72.5
A4	5	2	4	3	4	1	5	1	5	5	77.5
A5	4	1	4	3	5	2	3	2	4	2	75
A6	4	3	4	2	4	2	4	1	4	3	75
A7	5	1	5	2	4	2	4	2	4	2	82.5
A8	5	5	1	1	5	1	5	1	4	1	77.5
A9	5	3	5	1	5	3	5	4	5	5	72.5
A10	4	2	5	2	4	1	5	1	4	3	82.5
<b>Promedio</b>											80.25

Tabla 29 Análisis de las respuestas del cuestionario SUS

De acuerdo al análisis desarrollado en la Tabla 29 se obtuvo una puntuación promedio de 80.25 la cual según los rangos establecidos por el cuestionario SUS, como se hace referencia en Tabla 30, se obtuvo que la percepción global de usabilidad del sistema es excelente.

Perfecto	Excelente	Bueno	OK	Pobre	Inesperado
90 > x	80 > x	65 > x	50 > x	30 > x	30 < x

Tabla 30: Puntuación SUS basada en: (Finstad, 4 Agosto 2016).



#### 4.2 Discusión de Resultados:

- i. La NDT invierte el doble de tiempo que la UWE en la etapa de definición de requerimientos, sin embargo la UWE es la metodología en la cual se invirtió más tiempo frente a la NDT, pese a que esta última genera una mayor cantidad de artefactos frente a la UWE.
- ii. En cuanto al análisis del número de presentaciones para ser aceptado por el usuario, NDT convocó un mayor número de reuniones con los stakeholders, a comparación de la UWE, debido a que requiere definir a más detalle al sistema a elaborar en la etapa de definición de requerimientos.
- iii. La NDT es metodología que más artefactos generó frente a la UWE, sin embargo los artefactos que integran la usabilidad en la NDT son menores pese a que tiene más artefactos generados a comparación de la UWE, por otro lado en el mismo análisis también se puede apreciar que la NDT a pesar de tener menos artefactos que integran la usabilidad cuentan con el mayor número de mecanismos de usabilidad tratados como se puede observar en la tabla 23.

Realizada la evaluación y el análisis anteriormente a las metodologías de desarrollo Web con la integración de las directrices USEP's se determinó que la metodología más eficiente es la NDT dado a que da más énfasis a la etapa de definición de requerimientos, el tiempo que tomó el realizar la captura de requerimientos, definición de requerimientos, análisis y diseño del sistema con la integración de las directrices fue más corto frente a los resultados obtenidos de la UWE, también se pudo observar que la NDT es la metodología que integró un mayor número de mecanismos de usabilidad.

Cabe resaltar que la metodología NDT trabaja con matrices de trazabilidad, la cual relaciona los elementos generados por cada artefacto, con los elementos consecuentes del modelo generado producto de transformaciones, además de que cada elemento está alineado con los objetivos del proyecto, de tal forma que se mantiene la

consistencia e integridad de los requerimientos a través de una fase a otra, con la cual se procedió continuar a la fase de codificación cuyo prototipo funcional software resultante puede ser visualizado en el capítulo V: Propuesta de la Investigación.

Aplicada la evaluación con la ISO/IEC 9126-02 para medir la calidad externa de usabilidad al prototipo funcional, realizado bajo metodología de desarrollo web NDT y la integración de directrices USEP's para la captura de requerimientos de usabilidad en etapas tempranas de desarrollo de software, se ha logrado la satisfacción del cliente. Logrando obtener:

- Que el tiempo promedio del aprendizaje del funcionamiento del sistema fue de 5.85 minutos.
- El 20 % de los usuarios que manipularon el sistema indicaron que el sistema le fue muy fácil de aprender, mientras que el 40% de encuestados indico que el sistema es fácil de aprender, mientras que otro grupo que representa el 40% de los usuarios consideran que el sistema tiene un grado regular de facilidad de aprender.
- Otro punto a destacar de la evaluación del software se obtuvo que el 95 % de los usuarios que utilizaron el sistema comprendieron de una manera efectiva las funcionalidades del sistema. Pudiéndose observar también que los mensajes de error presentados en el sistema, fueron de gran utilidad para que el usuario pueda corregir los errores presentados.
- Cabe resaltar que el 88% de los usuarios que utilizaron el sistema indican que el sistema fue fácil de operar dado a que concluyeron de forma exitosa las actividades designadas.
- El 90% de los usuarios manifestó que el sistema les pareció atractivo, mientras que el 10% de los usuarios que utilizaron el sistema, indicaron que este tiene un grado regular de atraktividad.

## CAPITULO V: PROPUESTA DE LA INVESTIGACIÓN

La propuesta de investigación tuvo como objetivo evaluar las metodologías de desarrollo web que tratan requerimientos no funcionales como la usabilidad, con la integración de directrices para la captura de requerimientos de usabilidad, para ello se identificaron los diferentes factores de calidad de la usabilidad, de un producto software, propuesto por los diferentes estándares como lo son la ISO/IEC 9126, ISO/IEC 9241 Y LA ISO 25000.

Así mismo se identificó las diferentes metodologías, identificación basada en el trabajo de ESCALONA y KOCH (2004), donde se realizó un estudio de las metodologías Web bajo una visión general de las técnicas o modelos que proponen cada metodología para hacer frente a los requerimientos. El análisis y comparación de las metodologías en el trabajo de Escalona y Koch se realizaron bajo tres aspectos principales: El análisis de los tipos de requisitos manejados por cada metodología, el estudio de las técnicas empleadas y las fases que cubren en cada enfoque. El último de ellos se evalúa el grado de detalle de cada enfoque de acuerdo a las fases de desarrollo, las técnicas que aplican y los resultados obtenidos.

Seguido de ello se procedió a realizar la selección las metodologías para lo cual se tuvo como punto de partida el trabajo de investigación de ESCALONA y KOCH (2004) y (Del Valle Rojo, 2012), donde se hizo énfasis a los tipos de requerimientos tratados por cada enfoque metodológico de desarrollo de software Web, donde encontramos a los Requerimiento No funcionales, recordando que la usabilidad es una característica de calidad considerada como un requerimiento no funcional. Seleccionando las metodologías NDT y UWE, con las cuales se desarrolló el caso de estudio esta investigación, finalmente se aplicaron las metodologías de desarrollo con la integración de las directrices, para la captura de requisitos no funcionales exactamente la usabilidad en un caso de estudio.

El desarrollo de los objetivos alcanzados y mencionados anteriormente, son descritos a más detalle en los siguientes puntos.

### 5.1 Identificar los diferentes factores de calidad de la usabilidad.

Según (Sanchez, 2 Agosto del 2011) El término “usabilidad”, que deriva del inglés “Usability”, es considerado según (Navarro, Molina Dias, & Redondo, 2015) “*como un atributo de calidad ya que se puede evaluar que tan fácil puede ser la interacción de una interfaz gráfica. Por ello la usabilidad también se refiere a los métodos que son utilizados para mejorar la facilidad de uso durante la fase de diseño del ciclo de vida del software*”.

Existe una amplia gama de estándares que velan por garantizar la calidad de un producto software como son la ISO/IEC 9126, ISO/IEC 9241 y la ISO 25000, estas describen un conjunto de características de la calidad en la cual encontramos a la usabilidad, a continuación se dará una pequeña descripción desde las perspectivas que tienen contemplada cada una de estos estándares para contemplar una mayor precisión sobre la calidad de un producto software con respecto a la usabilidad.

*Tabla 31: La perspectiva del concepto de usabilidad por los distintos estándares. Fuente, elaboración propia.*

ISO/IEC 9126	ISO/IEC 9241	ISO 25000
(Sanchez, 2 Agosto del 2011) La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.	(Sanchez, 2 Agosto del 2011) Usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico	(Portal Iso, 2015) Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

Una vez realizada una descripción de cómo estos estándares miran la calidad de un producto a través de la usabilidad daremos a mencionar los siguientes factores que se tienen en cuenta cuando se hace mención a la usabilidad:





Tabla 32: Factores de Calidad de la Usabilidad. Fuente: Elaboración propia.

<i>Factor de Calidad de la Usabilidad</i>	<i>Descripción</i>
<b>Efectividad</b>	El producto software debe tener la capacidad de que los usuarios logren culminar sus actividades sin ningún problema.
<b>Eficiencia</b>	El software debe finalizar las actividades en el menor tiempo posible utilizando la menor cantidad de recursos.
<b>Satisfacción</b>	Es la comodidad y la aceptabilidad que el usuario tiene al utilizar el sistema web.
<b>Facilidad de entendimiento (compresibilidad)</b>	Es la capacidad del software para permitir que el usuario tenga una idea clara de cómo se utiliza y así pueda ser utilizado de una manera simple y sencilla.
<b>Facilidad de aprendizaje</b>	Capacidad del producto software para permitir a que los usuarios aprendan a utilizarlo sin el más mínimo esfuerzo.
<b>Operabilidad</b>	Capacidad del producto software que permite al usuario operarlo y controlarlo.
<b>Atractividad</b>	Se refiere a los atributos del software, tales como el uso de color y el diseño gráfico.
<b>Control de errores</b>	Es la capacidad que debe tener el producto software para evitar que el usuario cometa errores a la hora de utilizarlo.
<b>Accesibilidad</b>	Capacidad del producto software para que diferentes tipos de usuarios puedan utilizarlo sin importar las circunstancias.
<b>Cumplimiento de la Usabilidad</b>	Capacidad del producto software para adherirse a estándares relacionados con la usabilidad.

Una vez identificado los diferentes factores de usabilidad según los estándares ya mencionados, en el siguiente recuadro se mostrara que factor de usabilidad tiene en cuenta cada estándar.



Tabla 33: Factores de calidad de la Usabilidad, considerados por la SO/IEC 9126, ISO/IEC 9241, ISO 25000, Fuente: Elaboración propia

<i>Factores de calidad</i>	<i>ISO/IEC 9126</i>	<i>ISO/IEC 9241</i>	<i>ISO 25000</i>
<i>Efectividad</i>		X	
<i>Eficiencia</i>		X	
<i>Satisfacción</i>		X	
<i>Facilidad de entendimiento (compresibilidad)</i>	X		X
<i>Facilidad de aprendizaje</i>	X		X
<i>Operatividad</i>	X		X
<i>Atractividad</i>	X		X
<i>Control de errores</i>			X
<i>Accesibilidad</i>			X
<i>Cumplimiento de la Usabilidad</i>	X	X	

En la presente investigación se eligió evaluar los productos software del caso de estudio bajo la norma ISO/IEC 9126, dado a que es un estándar que ha sido establecida como el Modelo de Calidad de Software descrito en la Parte I dela Guía de Evaluación de Software, aprobado el número de Resolución Ministerial N° 139-2004-PCM, en el reglamento Key N° 28612 Ley que norma el uso, adquisición y adecuación del software en la administración pública aprobado mediante decreto supremo N° 024-206-PCM.

En la que señala que el desarrollo o selección del producto software con calidad es muy importante en la actualidad, ya que estas procesan información que es considerada como un activo importante de las organizaciones. También dada por la disponibilidad de la documentación de los estándares, por ser la ISO/IEC 9126-2 publicada en el 2001 y la ISO 25000 publicada recientemente en Julio del 2016. Se eligió el continuar con esta investigación bajo la selección de la ISO/IEC9126.



## 5.2 Identificar metodologías de desarrollo web, cuyo enfoque metodológico traten a la usabilidad.

En la investigación realizada por ESCALONA y KOCH (2004) se realizó el estudio de metodologías Web bajo una visión general de las técnicas o modelos que proponen cada metodología para hacer frente a los requerimientos. La mayoría de las metodologías analizadas y comparadas en este trabajo proporcionan una clasificación de requisitos.

El estudio comparativo se realizó bajo tres aspectos principales: El primero es el análisis de los tipos de requisitos manejados por cada metodología, el segundo aspecto es el estudio de las técnicas empleadas y las fases que cubren en cada enfoque. El último de ellos se evalúa el grado de detalle de cada enfoque en términos de su proceso de desarrollo, las técnicas aplicadas y los resultados obtenidos.

*Tabla 34 Requisitos que maneja cada Enfoque, Fuente: ESCALONA y KOCH (2004), página 206*

	<b>Req. Datos</b>	<b>Req. Interfaz usuario</b>	<b>Req. Navegacionales</b>	<b>Req. Personalización</b>	<b>Req, Transaccionales</b>	<b>Req. No Funcionales</b>
<i>WSDM</i>	✓			✓		✓
<i>SOHDM</i>	✓	✓			✓	
<i>RNS</i>	✓	✓	✓		✓	
<i>HFPM</i>	✓	✓	✓			✓
<i>OOHDM</i>	✓	✓	✓			
<i>UWE</i>	✓	✓	✓	✓		✓
<i>W2000</i>			✓	✓	✓	
<i>WEBML</i>	✓	✓		✓	✓	✓
<i>NDT</i>	✓	✓	✓	✓	✓	✓
<i>DDDP</i>	✓	✓	✓	✓	✓	✓

La siguiente tabla muestra las diferencias de las metodologías en cuanto a las técnicas que utilizan durante las actividades de elicitación, especificación y validación



Tabla 35: Las técnicas utilizadas en la fases de captura, definición y validación, Fuente: ESCALONA y KOCH (2004), página 207

	WSDM	SOHDM	RNA	HFPM	OOHDM	UWE	W2000	WEBML	NDT	DDDP
<b>Captura de requerimientos</b>	Entrevista	X		X		X		X	X	X
	JAD								X	
	Lluvia de ideas								X	
	Mapas conceptuales	Ro- Activida d								
	Modelado de casos de uso					X				
	Cuestionario / Lista de verificación						X			
	El bosquejar y storyboarding							X		
	Otras técnicas		DFD					Documento de análisis		
<b>Definición</b>	Leguaje Natural	X		X	X			X		
	Glosario				X	X			X	
	Patrones y Templates							X	X	
	Escenarios		SAC			X				
	Análisis de casos de uso				X	X	X	X	X	
	Lenguaje formal									
	Prototipado									X
	Otras técnicas		Lista de Eventos		Bocetos de interfaz	UIDs			Expresiones BNL	
<b>Validación</b>	Revisión /Walk-through					X			X	
	Auditoria					X				
	Matriz de trazabilidad								X	
	Prototipado				X	X				X
	Otras técnicas							Aceptación de pruebas		

Algunas metodologías se centran en gran medida en el proceso de desarrollo, otros se centran en las técnicas o en la estructura de los



resultados producidos. Pero en esta investigación evaluó a las metodologías, definiendo el grado de detalle que se encuentran en la descripción del proceso, las técnicas y los productos. Y el resultado de esta evaluación la podremos observar a continuación:

Tabla 36: Grado de detalle en los procesos, técnicas y productos. Fuente: ESCALONA y KOCH (2004, página 209).

	Orientada a procesos	Orientado a la Técnica	Orientado al producto
<i>WSDM</i>	O	-	-
<i>SOHDM</i>	-	+	-
<i>RNA</i>	+	-	-
<i>HFPM</i>	+	O	+
<i>OOHDM</i>	O	+	-
<i>UWE</i>	+	O	O
<i>W2000</i>	O	O	-
<i>WEBML</i>	O	O	+
<i>NDT</i>	O	+	+
<i>DDP</i>	+	O	-

Donde:

- a) Orientada a los procesos: el enfoque describe claramente los pasos a seguir (+), describe el proceso sin detalles (O), o no indica ningún proceso en absoluto (-).
- b) Técnica-orientado: el enfoque claramente describe las técnicas y la forma de aplicar de ellos (+), enumera las técnicas a aplicar (O), o no propone ninguna técnica (-).
- c) Producto orientado: el enfoque describe claramente la estructura del producto a ser producida (+), que describe el producto sin detallar su estructura (o), o no lo hace dar ninguna indicación sobre el producto resultante (-)



### 5.3 Seleccionar las metodologías de desarrollo web, cuyo enfoque metodológico trata requerimientos de usabilidad

El trabajo realizado por ESCALONA y KOCH (2004) muestra los tipos de requerimientos tratados por cada enfoque metodológico de desarrollo de software Web, en la cual encontramos a los Requerimiento No funcionales, recordando que la usabilidad es una característica de calidad considerada como un requerimiento no funcional.

Tabla 37: Requerimientos identificados por cada metodología. Fuente: (Rojo, 2012, página 77)

	R. DATOS	R DE INTERFAZ DE USUARIO	R. NAVEGACIONAL	R. PERZONALIZACION	R. TRANSACCIONAL	R. NO FUNCIONALES
WSDM	X			X		x
SOHDM	X	X			X	
RNA	X	X	X		X	
HFPM	X	X	X			X
OOHDM	X	X	X			
UWE	X	X		X		X
W2000			X	X	X	
WEBML	X	X		X		X
NDT	X	X	X	X	X	X
DDDP	X	X	X	X	X	X

En la investigación realizada por (Rojo, 2012, página 72) presenta el análisis de las metodologías tomando en cuenta las características de calidad que tiene la ISO/IEE 9126 para identificar que metodología consideran una de estas características. Y como resultado de su investigación se obtuvo como conclusión que la característica de calidad, usabilidad, es tratada por todas las metodologías.



Tabla 38: Requerimientos No Funcionales identificados y tratados por cada metodología de desarrollo Web – Parcial. Fuente: (Rojo, 2012, página 78)

	Atributos de Calidad						Restricciones				
	Atributos de Calidad ISO/IEC 9126-1						Restricciones del proceso de desarrollo				
							Alcance de proceso		Atributos de Calidad ISO/IEC 9126-1		
	Usabilidad	Performance	Usabilidad	Performance	Usabilidad	Performance	Usabilidad	Performance	Usabilidad	Performance	Usabilidad
<i>WSDM</i>	X		X		X				X	X	
<i>HFPM</i>	X	X							X	X	
<i>UWE</i>	X		X						X	X	
<i>WEBML</i>	X	X	X						X		X
<i>NDT</i>	X				X	X		X			X
<i>DDDP</i>	X	X	X				X		X	X	

De esta investigación se tomó el análisis que se realizó sobre la identificación de las técnicas para la elicitación, especificación y validación de requerimiento son funcionales y herramientas que usan para su administración, en cada una de las metodología ya seleccionadas anteriormente.

Tabla 39: Técnicas de Ingeniería de requerimientos y herramientas CASE aplicadas a RNF. Fuente: (Rojo, 2012, página 84)

	TÉCNICAS			HERRAMIENTA CASE	
	Elicitación/ captura	Especificación/ Definición	Validación	Soporta el proceso de desarrollo	Soporta la Administración de Requerimientos
<i>WSDM</i>					
<i>HFPM</i>			Prototipo		
<i>UWE</i>			Prototipo	ArgoUML	ArgoUML
<i>WEBML</i>		Lista	de Test de aceptación	totiiSoft	TotiiSoft



	requerimientos no funcionales	WebRatio
<i>NDT</i>	Patrón para describir los requerimientos no funcionales	- Revisiones - Auditorias - Matriz de trazabilidad - Prototipos
<i>DDDP</i>		- NDT - Tool - NDT - Tool

Y se tuvieron las siguientes afirmaciones:

- a) De acuerdo al análisis realizado se llega a la conclusión que ninguno de los procesos mencionan con exactitud que técnicas utilizan para la identificación de requerimientos no funcionales.
- b) Otro análisis que se realizó, fue que la NDT nos ofrece una plantilla para poder describir los requerimientos no funcionales de una manera eficiente.
- c) La UWE si bien nos propone lineamientos para documentar los requisitos no funcionales, hace énfasis al desarrollo de prototipos durante la primera fases del ciclo de software para captura de una manera eficiente los requisitos tanto funcionales como no funcionales.
- d) Otro punto que se concluyo es que la NDT, hace uso de diferentes técnicas para la verificación y validación de requisitos no funcionales como son las revisiones, auditoria, matriz de trazabilidad y prototipos.

Partiendo de investigaciones como las de (Del Valle Rojo, 2012) y ESCALONA y KOCH (2004) se ha optado por elegir las metodologías de desarrollo de Software Web NDT y UWE, dado a que son metodologías que tratan a los requerimientos no funcionales, específicamente a la usabilidad, además de que las técnicas y herramientas propuestas por cada metodología son de gran aporte para realizar esta investigación.



#### 5.4 Aplicar las metodologías con la integración de las directrices para la captura de requerimientos de usabilidad en un caso de estudio.

(Juristo, 2007) Propone unas guías para especificar los requisitos funcionales de usabilidad de acuerdo a cada uno de los mecanismos de usabilidad, con el fin de establecer un conjunto de preguntas que pueden ser necesarias responder con el usuario o solo con la experiencia del analista. *“Estas preguntas tratan de determinar cuál de entre todas las posibles aplicaciones del mecanismo de usabilidad, desea incorporar el usuario al sistema”*. En esta investigación se hicieron uso de las preguntas determinadas por las guías, para identificar las formas de uso de cada mecanismo de usabilidad. En el diagrama 1 y en el diagrama 2 se representa las actividades realizadas para la integración de las directrices con la metodología NDT y UWE.

Unos de los primeros pasos para proceder a la integración de las directrices con las metodologías consisten en interpretar las guías de mecanismo de usabilidad que se hace referencia en (Ignacio Panach, Juristo, Valverde, & Pastor, 2015) para poder identificar los posibles mecanismo de usabilidad que se puedan utilizar a la hora de desarrollar el modelado de acuerdo a cada una de las metodologías. Seguido de ello procedió a realizar la captura de requerimientos del sistema, los cuales fueron analizados para poder identificar los mecanismos de usabilidad que se alinean al caso de estudio, añadiendo cierta funcionalidad para mejorar la interacción y navegabilidad del usuario en el sistema. El análisis realizado en esta Actividad, retroalimenta a la matriz de requerimientos, responder a las preguntas establecidas en las Guías implica identificar nuevos requerimientos, que fueron discutidos con el Stakeholders en reuniones. Luego se procedió a plasmar los requerimientos en los artefactos indicados por las metodologías UWE Y NDT.

En los siguientes puntos se explicara a más detalle los documentos generados por las actividades mencionadas anteriormente.

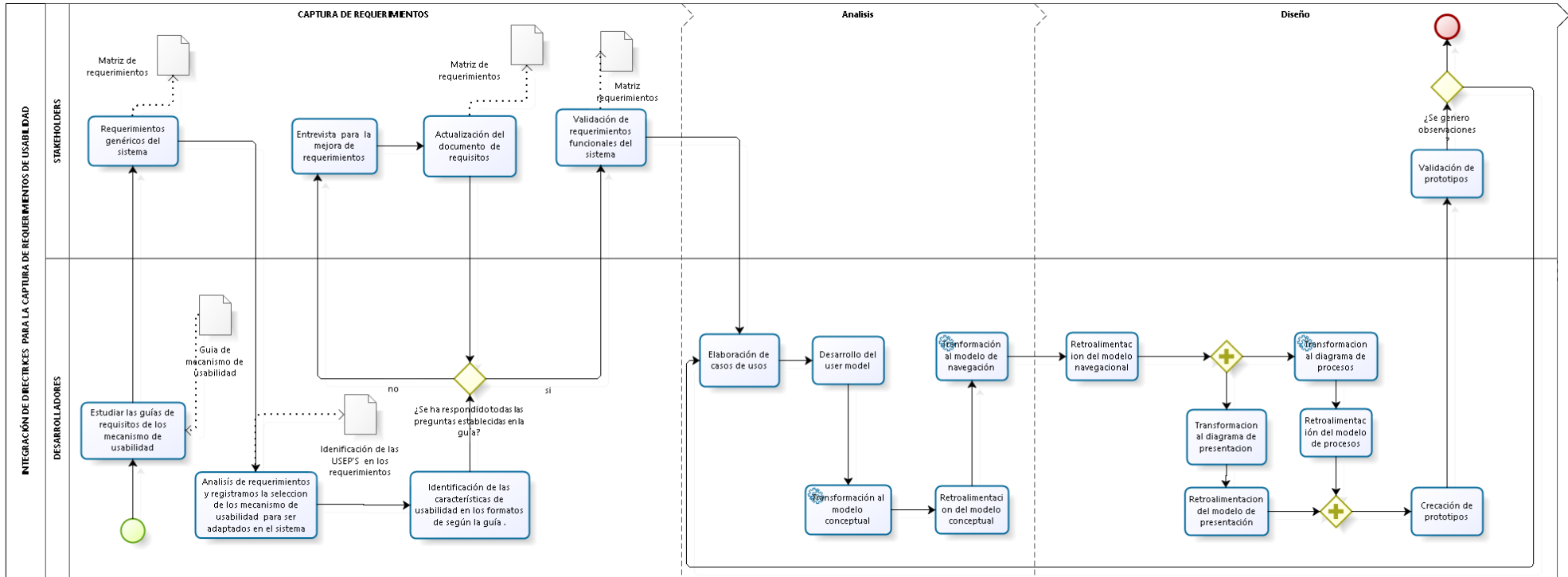


Diagrama 1 Integración de la metodología UWE con las Directrices para la captura de Requerimientos de Usabilidad.



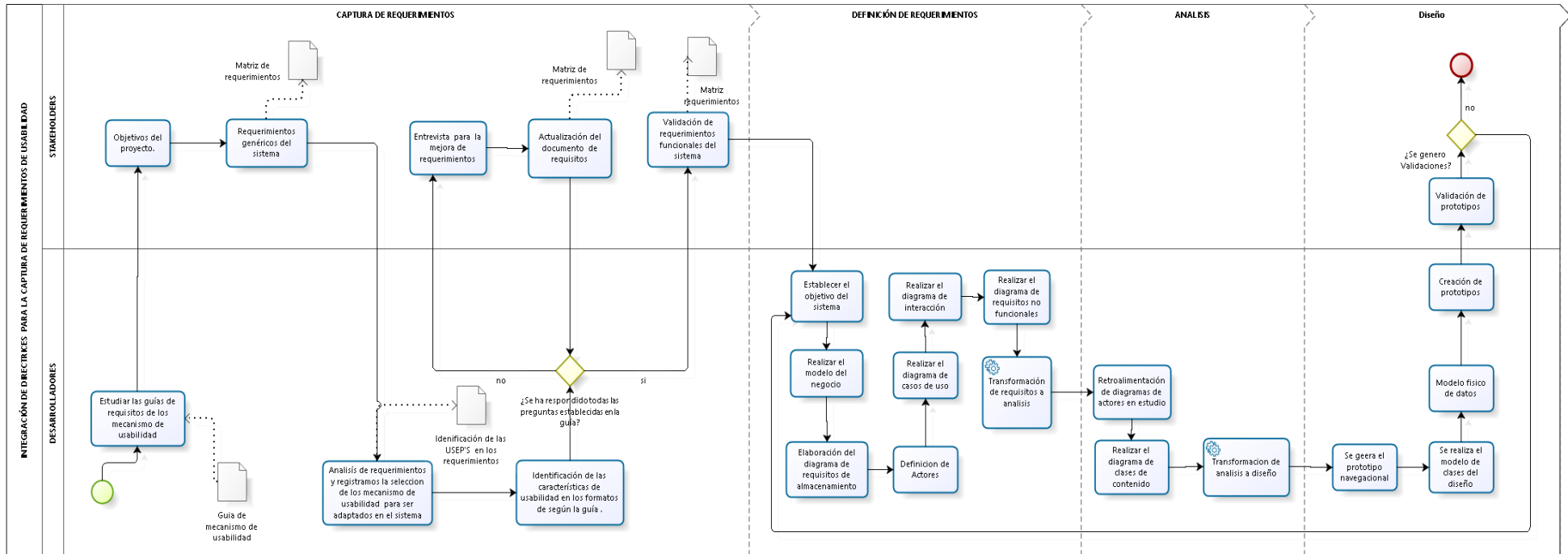


Diagrama 2 Integración de la metodología NDT con las Directrices para la captura de Requerimientos de Usabilidad.



### 5.4.1. CAPTURA DE REQUERIMIENTOS

La fase de captura de requerimientos, es compartida por las dos metodologías dado que esta fase tuvo como objetivo la comprensión de los diferentes procesos del negocio del stakeholders, para ello hizo uso de entrevistas para realizar la captura de requerimientos del sistema, teniendo como resultado la definición de Matriz de Requerimientos, el documento de Análisis Preliminar De Requerimiento, para integrar las directrices Usability Elicitation Patterns(USEPs), así como también se generó el documento de identificación de las características de Usabilidad, aplicando la Guía USEPs y el documento de definición de requerimientos funcionales de usabilidad.

#### 5.4.1.1. Matriz de Requerimientos.

La matriz de requisitos es utilizado para asegurar que cada requerimiento agrega un valor agregado al negocio mostrándote el vínculo entre los requisitos y las necesidades del negocio.

Nº	REQUERIMIENTO	TIPO DE REQUERIMIENTO	FECHA DE INCLUSIÓN	FUENTE	SUSTENTO DE SU INCLUSIÓN
1	El sistema debe ser desarrollado en una Plataforma WEB	Requerimiento No Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
2	Debe facilitar el registro de los docentes del centro educativo	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
3	Debe facilitar el registro de los alumnos del centro educativo	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
4	Cada docente debe contar con un usuario y una contraseña para poder acceder al sistema.	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex

5	Cada alumno debe contar con un usuario y una contraseña para poder acceder al sistema.	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
6	Debe facilitar el registro de grados y secciones	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
7	Debe facilitar el registro de niveles de educación	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
8	Facilitar el registro de tipos de evaluaciones	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
9	Facilitar la elaboración de exámenes	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
10	Facilitar la programación de exámenes	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
11	Asignar un puntaje diferente a cada pregunta de cada examen.	Requerimiento Funcional	09/08/2016	Entrevista	Docente de la I.E.P. Roxana Locorni Siesquen
12	Que la cantidad de preguntas varíen por cada examen.	Requerimiento Funcional	09/08/2016	Entrevista	Docente de la I.E.P. Roxana Locorni Siesquen
13	Cada pregunta debe estar relacionada a una capacidad de un determinado curso.	Requerimiento Funcional	09/08/2016	Entrevista	Docente de la I.E.P. Roxana Locorni Siesquen
14	Facilitar realizar tipos de pregunta para cada examen a crear (selección única, selección múltiple, responder, para poner juicio crítico)	Requerimiento Funcional	09/08/2016	Entrevista	Docente de la I.E.P. Roxana Locorni Siesquen
15	Cada examen debe de contar con un tiempo determinado a desarrollar	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
16	Se puede programar el número de veces que un alumno puede realizar una evaluación	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
17	Cada evaluación pertenece a un Bimestre y es un tipo de examen específico (IBM, practicas, exámenes mensuales y exámenes	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex



	bimestrales, etc.).				
18	Al resolver el alumno debe visualizar el tiempo, que le queda para enviar el examen	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
19	El docente puede visualizar el reporte de logros por cada alumno	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
20	El docente podrá programar el día y hora en la que los exámenes estarán habilitados	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
21	Docente puede visualizar el reporte de logros de la clase por cada examen	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
22	El docente puede visualizar el progreso de cada sección en un determinado curso	Requerimiento Funcional	06/08/2016	Lluvia de ideas	Representante de la I.E.P. Melgar Neciosup Alex
23	El alumno puede visualizar su progreso por curso	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
24	El alumno puede visualizar un reporte de los resultados obtenidos por cada examen	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
25	El alumno debe visualizar sus diferentes cursos y así poder acceder a su examen	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
26	Las preguntas serán designadas de forma alternada para cada alumno	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
27	El calificativo de cada examen es de 0 a 20 (vigesimal)	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
28	Una vez enviado el examen el alumno solo podrá visualizar la evaluación con las respuestas dadas	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
29	El docente podrá habilitar y deshabilitar la evaluación con sus respuestas	Requerimiento Funcional	06/08/2016	Entrevista	Representante de la I.E.P. Melgar Neciosup Alex
30	Los Reportes dados al docentes deben de ser exportados a Excel	Requerimiento	06/08/2016	Lluvia de	Representante de la I.E.P.



		Funcional		ideas	Melgar Neciosup Alex
31	Una vez rendido el examen el alumno pueda visualizar sus respuestas y pueda descargar en pdf.	Requerimiento Funcional	09/08/2016	Entrevista	Docente de la I.E.P. Roxana Locorni Siesquen
32	El calificativo también se deberá mostrar por calificación por letras.	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Marleny Selada Delgado
33	Los alumnos pueden navegar entre las preguntas (regresar e ir a la siguiente).	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
34	Para los exámenes que tengan preguntas de juicio crítico, estos deben ser corregidos, evaluados por los docentes, para ello se deben de cargar en una especie de bandeja en espera de ser calificados.	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
35	El docente puede programar la fecha de visualización de la resolución de la evaluación	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
36	El docente decide la fecha de publicación de notas	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
37	El docente se autentificará en el sistema con el número de su DNI y una contraseña	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
38	El alumno se autentificará en el sistema con el número de su DNI y una contraseña	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
39	Las contraseñas son otorgadas en secretaria, las cuales pueden ser modificadas	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
40	Las contraseñas son otorgadas a los alumnos por el tutor del grado y sección.	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
41	El alumno en su perfil solo puede cambiar su contraseña, mas no sus datos informativos, como nombre apellido, etc.	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
42	Para recuperación de contraseñas, el alumno solicita su contraseña al tutor del aula.	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge



43	Las contraseñas pueden incluir letras, números y símbolos.	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
44	El docente puede visualizar el reporte de logros por cada alumno ante una evaluación considerando, preguntas correctas, preguntas erróneas, preguntas en blanco así como la nota obtenida.	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
45	El docente puede acceder a una vista previa del examen realizado por el alumno	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge
46	El docente puede visualizar el reporte de logros obtenidos la clase en un examen en específico, así como el progreso general de la clase teniendo en cuenta, número de preguntas respondidas correctamente, número de preguntas incorrectas, número de preguntas en blanco y promedio general.	Requerimiento Funcional	<u>09/08/2016</u>	Entrevista	Docente de la I.E.P. Querevalú Paiva Jorge



### 5.4.1.2. INTEGRACIÓN DE LAS DIRECTRICES USABILITY ELICITATION PATTERNS, USEPS: ANÁLISIS PRELIMINAR DE REQUERIMIENTO.

Se adaptó las USEPs a nuestra versión de análisis preliminar de los requisitos de software generados, en las entrevistas con los interesados (Stakeholders) del proyecto, plasmado en la Matriz de requerimientos, seguido de ello se identificó y seleccionó todos los mecanismos de usabilidad que están relacionadas con la funcionalidad del sistema en base a su grupo de características de usabilidad y el objetivo general de cada mecanismos. En la siguiente tabla se ha realizado la descripción de los requerimientos, seguido de la identificación preliminar de los mecanismos de usabilidad, acorde a la funcionalidad deseada a implementar en el sistema por los principales Stakeholders.

#### REQUISITO FUNCIONAL DOCUMENTADO: ANÁLISIS PRELIMINAR DE REQUERIMIENTO

##### Requerimiento A: Logueo

RA

1. El docente se identificará en el sistema con el número de su DNI y una contraseña (*Structured Text Entry*) (*Warning*).
2. El alumno se identificará en el sistema con el número de su DNI y una contraseña (*Structured Text Entry*) (*Warning*).
3. Las contraseñas serán otorgadas en secretaría, las cuales pueden ser modificadas.
4. Las contraseñas son otorgadas a los alumnos por el tutor del grado y sección.
5. Para recuperación de contraseñas, el alumno solicita su contraseña al tutor del aula.
6. Las contraseñas pueden incluir letras, números y símbolos.

##### Requerimiento B: sistema en General

RB

1. Alerta de conexión a internet. (*System Status Feedback*)
2. Alerta de conexión con el servidor. (*System Status Feedback*)
3. Alertas para los exceptions de los try catch. (*System Status Feedback*)

##### Requerimiento C: Crear Exámenes (*Step by Step*)

RC

Una vez que el usuario se ha autenticado en el sistema, este podrá crear evaluaciones, para ello debe: elegir el grado y sección el cual tiene a cargo, seguido de seleccionar la asignatura a evaluar.

Seleccionado la asignatura a evaluar, da clic en el menú de crear Exámenes, para ello deberá seguir los siguientes pasos.

#### 1. Programación del examen (*Help*)



- 1.1. Ingresar un título para evaluación (*Structured Text Entry*)
  - 1.2. Selecciona el tipo de evaluación
  - 1.3. seleccionar la fecha en la que se llevará a cabo el examen (*Structured Text Entry*)
  - 1.4. Ingresar la hora de inicio y la hora de fin de la evaluación (*Structured Text Entry*)
  - 1.5. Cantidad de preguntas (*Structured Text Entry*)
  - 1.6. habilitar o deshabilitar el modo random para asignación de preguntas (*Warning*).
  - 1.7. Habilitar la opción de publicación automática de notas (*Warning*).
- 2. Registro de preguntas (*Help*)**
- 2.1. Seleccionar el tipo de pregunta
  - 2.2. Ingresar el enunciado de la pregunta (*Structured Text Entry*)
  - 2.3. Asignar la capacidad a la que pertenece dicha pregunta, opcional
  - 2.4. Asignar el puntaje que tendrá la pregunta (*Structured Text Entry*)
  - 2.5. Guardar las preguntas (*Interaction Feedback*) (*Warning*) (*Progress Feedback*) (*Abort Operation*)
  - 2.6. Vista previa de las preguntas registradas
- 3. Programación de la vista de resultados**
- 3.1. El docente programa la fecha de visualización de la resolución de la evaluación. (*Structured Text Entry*)
  - 3.2. El deberá seleccionar el tipo de calificativo a mostrar (0 a 20 vigesimal) o por calificación por letras. (*Warning*).
- 4. Guardar Examen (*Interaction Feedback*) (*Progress Feedback*) (*Warning*) (*Abort Operation*) (*Shortcuts*).**

## Requerimiento D: Visualización de Reportes

RD

1. El docente puede visualizar el reporte de logros por cada alumno
2. Docente puede visualizar el reporte de logros de la clase por cada examen
3. El docente puede visualizar el progreso de cada sección en un determinado curso
4. Los Reportes dados a los docentes deben de ser exportados a Excel. (por cada reporte) (*Progress Feedback*) (*Abort Operation*)
5. El docente puede visualizar el reporte de logros por cada alumno ante una evaluación considerando, preguntas correctas, preguntas erróneas, preguntas en blanco así como la nota obtenida. (dentro de las opciones de arriba)
6. El docente puede acceder a una vista previa del examen realizado por el alumno. (dentro de las opciones de arriba)
7. El docente puede visualizar el reporte de logros obtenidos la clase en un examen en específico, así como el progreso general de la clase teniendo en cuenta, número de preguntas respondidas correctamente, número de preguntas incorrectas, número de preguntas en blanco y promedio general. (Dentro de las opciones de arriba).
8. Los reportes en caso de usar colores para mostrar el estado de un alumno o la nota lograda, es necesario manejar una leyenda de colores (*Help*)

## Requerimiento E: Corrección de exámenes

RE



**(Help)**

1. El usuario visualiza los exámenes a corregir en una especie de bandeja, donde puede visualizar la resolución de la evaluación por cada alumno.
  - 1.1. Al abrir el examen del alumno se le carga las preguntas con su determina respuesta, la cual el docente podrá otorgar una calificativo (*Structured Text Entry*) a las preguntas que son de tipo argumentativo.
  - 1.2. Finalizado la corrección de la evaluación confirma el guardar la evaluación (*Interaction Feedback*) (*Warning*) (*Abort Operation*).
  - 1.3. Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación (*Interaction Feedback*) (*Progress Feedback*) (*Warning*) (*Abort Operation*).
2. El docente podrá habilitar y deshabilitar el publicar las respuestas correctas de la evaluación. (*Warning*).

**Requerimiento F: Publicación de notas (Help)**

**RF**

1. El docente visualiza todos los exámenes ya corregidos y sus estados de publicación, en caso de que no esté habilitado la opción de publicación de notas este podrá programar la fecha publicación de las notas (*Structured Text Entry*) (*Warning*).

**Requerimiento G: Dar Exámenes (Help)**

**RG**

Una vez que el usuario se ha autenticado en el sistema, este podrá rendir evaluaciones, para ello debe: seleccionar la asignatura, seleccionar entre los exámenes habilitados el examen a realizar.

1. El alumno podrá visualizar el tiempo designado para la prueba, así como un cronómetro que le indicara el tiempo que le queda para finalizar la prueba (*Warning*).
2. **Resolver el examen**

Para dar respuesta al examen este podrá navegar entre las preguntas (Pagination) y seleccionar la pregunta a desarrollar, si la pregunta es de tipo:

  - 2.1. Opción única, debe seleccionar entre las alternativas de respuesta, una única opción
  - 2.2. Opción múltiple, debe seleccionar entre las alternativas de respuesta, las opciones que respondan o guarden relación a la pregunta
  - 2.3. Juicio Crítico, el alumno debe de ingresar su opinión o respuesta argumentativa. (*Structured Text Entry*)
  - 2.4. Relacionar (*Help*)
3. **Enviar el examen**
  - 3.1. El alumno selecciona el botón de enviar examen (*Interaction Feedback*) (*Warning*) (*Progress Feedback*) (*Abort Operation*).
  - 3.2. Una vez enviado el examen el alumno solo podrá visualizar la evaluación con las respuestas dadas (*Progress Feedback*) (*Abort Operation*).
  - 3.3. Para los exámenes que tengan preguntas de juicio crítico, estos deben ser corregidos, evaluados por los docentes, para ello se deben de cargar en una especie de bandeja en espera de ser calificados.
  - 3.4. El calificativo que le mostrara al alumno dependerá a la configuración dada por el docente.

**Requerimiento H: Visualización de Reportes**

**RH**



1. El alumno puede visualizar su progreso por curso
2. El alumno puede visualizar un reporte de los resultados obtenidos por cada examen (*warning*) (*Progress Feedback*) (*Abort Operation*)
3. Una vez rendido el examen el alumno pueda visualizar sus respuestas y pueda descargar en pdf. (*Progress Feedback*) (*Abort Operation*)
4. Los reportes en caso de usar colores para mostrar el estado de un alumno o la nota lograda, es necesario manejar una leyenda de colores (*Help*)

### 5.4.2. IDENTIFICACIÓN DE LAS CARACTERÍSTICAS DE USABILIDAD, APLICANDO LA GUÍA USEPS

La Guía USEPs tiene por cada mecanismo de usabilidad dos elementos que sirven para establecer los requisitos de usabilidad funcional: En la sección de Guía de Elicitación, se encuentran las recomendaciones dadas por los expertos HCI y una lista de problemas a discutir con los stakeholders. En las Sigüientes tablas podremos ver los temas discutidos y los nuevos requerimientos que surgieron a partir de la aplicación de la guía.

#### 1. FEEDBACK

##### a. System Status Feedback

<b>IDENTIFICADOR</b>	
<b>Nombre:</b> System Status Feedback	
<b>Familia:</b> Feedback	
<b>Alias:</b> Status Display Modelling Feedback Area	
<b>PROBLEMA</b>	
Qué información se debe obtener y especificar para que la aplicación proporcione a los usuarios retroalimentación del estado del sistema.	
<b>CONTEXTO</b>	
Cuando cambios que son importantes para el usuario ocurren durante: <ul style="list-style-type: none"> <li>➤ La ejecución de una tarea</li> <li>➤ Porque no hay suficientes recursos en el sistema</li> <li>➤ Porque recursos externos no funcionan correctamente</li> </ul>	
<b>SOLUCION</b>	
<b>Guía de Obtención de los Mecanismo de Usabilidad</b>	
<b>Temas a discutir con los Stakeholders</b>	<b>Temas a discutir con los Stakeholders</b>
1.1 ¿El usuario quiere que el sistema le avise de los estados del sistema?, en ese caso, ¿de cuáles?	<b>RB1</b> Alerta de conexión a internet (señal



<p>1.2 ¿El usuario quiere que el sistema le avise de fallos del sistema (cualquier operación que el sistema no es capaz de completar, pero no son fallos provocados por entradas incorrectas del usuario)? En ese caso ¿Cuáles?</p> <p>1.3 ¿El usuario quiere que el sistema avise si no hay suficientes recursos para ejecutar las órdenes en curso? En ese caso, ¿qué recursos?</p> <p>1.4 ¿El usuario quiere que el sistema le avise si hay problemas con los recursos externos o dispositivos con los que el sistema interactúa? En ese caso, ¿Cuáles?</p>	<p>baja).</p> <p><b>RB2</b> Alerta de conexión con el servidor.</p> <p><b>RB3</b> Alertas para los exceptions de los try catch.</p>
<p>2.1 ¿Qué información se debe mostrar al usuario?</p> <p>2.2 ¿Qué información se tendrá que mostrar de manera bloqueante porque pertenece a una situación crítica? Esta información se suele representar en una ventana emergente de la ventana principal que impide al usuario seguir hasta que no la cierre.</p> <p>2.3 ¿Cuál de esta información tendrá que ser resaltada porque es importante pero no crítica? Usando diferentes colores, sonidos, tamaños, etc.</p> <p>2.4 ¿Cuál de esta información será mostrada en el área del estado? Para cada componente que muestre el estado del sistema, de acuerdo a su importancia, las posibilidades de visualización va desde los componentes bloqueantes (por ejemplo, una ventana en el área principal que impide al usuario continuar hasta que sean cerradas), Pasando por resaltar ciertas zonas (con diferentes colores, sonidos, movimientos o tamaños) hasta las percepciones del ojo humano (como la identificación de un icono ubicado en el área del estado del sistema).</p>	<p><b>RNB1.1</b> Mensaje: Verificar su conexión a internet.</p> <p><b>RNB1.2</b> Mensaje: No se puede encontrar la página web</p> <p><b>RNB1.3</b> Los mensajes generados por expresiones tendrán la siguiente estructura: “Mensaje de Alerta” seguido de la descripción de la incidencia percibida, valor que es retornado por el manejador de excepciones.</p> <p><b>RNBSS1</b> Cada mensaje de System Status Feedback debe de estar enlazada a un icono que represente el tipo alerta.</p> <p><b>RNBSS2</b> Los mensajes se le mostrarán a través de una ventana emergente (modal).</p>



<p>Se puede observar que durante el proceso de elicitación de requisitos, el debate sobre el tipo exacto de respuesta se puede dejar hasta la fase de diseño de la interfaz. Es en el diseño cuando se debe pensar sobre la importancia de la información a mostrar y el tipo de ventana (bloqueante, resaltada o estándar).</p>	
<p>3.1 ¿Las personas que usan el sistema son de diferentes culturas? En ese caso, el sistema debe presentar el estado del sistema en la forma adecuada (de acuerdo a la cultura del usuario). Así que se debe preguntar al usuario sus costumbres y cultura.</p> <p>3.2 ¿Cuál es el mejor lugar para ubicar la información de retroalimentación de cada situación?</p>	<p><b>RNB1.6</b> Los usuarios a utilizar el sistema son de habla hispana.</p> <p><b>RNBSS3</b> La ventana de alerta o emergente se debe ubicar en la parte central de la interfaz del usuario.</p>

## b. Interaction Feedback

<b>IDENTIFICADOR</b>	
<b>Nombre:</b> Interaction Feedback	
<b>Familia:</b> Feedback	
<b>Alias:</b> Interaction Feedback Modelling Feedback Area	
<b>PROBLEMA</b>	
Qué información debe ser obtenida y especificada para mostrar al usuario que el sistema ha escuchado su petición.	
<b>CONTEXTO</b>	
Cuando el usuario realiza un evento de interacción, como un clic de ratón, movimiento de ratón, apretar una tecla, etc. El sistema debe informar al usuario de que la interacción se ha aceptado.	
<b>SOLUCION</b>	
<b>Guía de Obtención de los Mecanismo de Usabilidad</b>	
<b>Temas a discutir con los Stakeholders</b>	<b>Temas a discutir con los Stakeholders</b>
<p>1.1 Para cada acción que el usuario solicite al sistema se debe decidir, qué tipo de retroalimentación mostrará el sistema (marcando un botón, resaltando una palabra, cambiando el tamaño del cursor o los objetos involucrados)</p> <p>Se debe verificar que la aplicación</p>	<p>Cuando de en las siguientes acciones:</p> <p><b>RC2.5</b> Guardar las preguntas</p> <p><b>RC4</b> Guardar Examen</p> <p><b>RE1.2.</b> Finalizado la corrección de la evaluación confirma el guardar la</p>



<p>proporciona la retroalimentación en 0.1 milisegundos después de cada vez que se pulse una tecla, se mueva el ratón, o el usuario haga cualquier otra actividad sobre el sistema.</p>	<p>evaluación.</p> <p><b>RE1.3</b> Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación</p> <p><b>RG3.1</b> El alumno selecciona el botón de enviar examen</p> <p><b>RNIF1</b> Cuando el usuario realiza un evento de interacción con botones comprometidos con acciones principales, por ejemplo la pulsación de una tecla, el clic de ratón o movimiento de ratón, el sistema deberá responder de forma diferenciadora a este evento.</p> <p><b>RNC2.5</b> Al clic del ratón o movimiento de ratón, sobre el botón “Guardar las preguntas” este deberá responder a la interacción, cambiando de tamaño.</p> <p><b>RNC4</b> Al clic del ratón o movimiento de ratón, sobre el botón “Guardar Examen” este deberá responder a la interacción, cambiando de tamaño.</p> <p><b>RNE1.2</b> Al clic del ratón o movimiento de ratón, sobre el botón “Guardar Corrección” este deberá responder a la interacción, cambiando de tamaño.</p> <p><b>RNE1.3</b> Al clic del ratón o movimiento de ratón, sobre el botón “Finalizar Corrección” este deberá responder a la interacción, cambiando de tamaño.</p> <p><b>RNG3.1</b> Al clic del ratón o movimiento de ratón, sobre el botón “Enviar examen” este deberá responder a la interacción, cambiando de tamaño.</p>
---	---

### c. Progress Feedback

<b>IDENTIFICADOR</b>
<b>Nombre:</b> Progress Feedback
<b>Familia:</b> Feedback



<p><b>Alias:</b> Progress Indicator Progress Show Computer is Thinking, Time to Do Something Else Modelling Feedback Area</p>	
<p><b>PROBLEMA</b></p>	
<p>Qué información debe ser obtenida y especificada para proporcionar a los usuarios la información que está relacionada con la evolución de las tareas solicitadas.</p>	
<p><b>CONTEXTO</b></p>	
<p>Cuando un proceso interrumpe la interfaz de usuario durante dos segundos o más.</p>	
<p><b>SOLUCIÓN</b></p>	
<p><b>Guía de Obtención de los Mecanismos de Usabilidad:</b></p>	
<p><b>Temas a discutir con los Stakeholders</b></p>	
<p>1.1 ¿Qué servicios es probable que duren más de 2 segundos y cuáles de ellas son críticas? 1.2 ¿Cómo se informará al usuario cuando el proceso termine?</p>	<p>RC4. Guardar Examen RD4. Los Reportes generados por docentes deben de ser exportados a Excel. (por cada reporte) RE1.3 Una vez corregido todos los exámenes este podría dar por finalizado el proceso de evaluación. RG3.1 El alumno selecciona el botón de enviar examen. RG3.2 Una vez enviado el examen el alumno solo podrá visualizar la evaluación con las respuestas dadas. RH2 El alumno puede visualizar reportes de de sus logros. RH3 Una vez rendido el examen el alumno pueda visualizar sus respuestas y pueda descargar en pdf.  RNPF1. Los Progress feedback se deben de mostrar a través de una venta ventana emergente (modal).</p>
<p>2.1. ¿Cómo se informará al usuario sobre el progreso de los diferentes servicios y qué información se desea para cada uno? 2.2. Se debe verificar que la aplicación no consume más de 1 minuto para mostrar el indicador de progreso y actualizar la retroalimentación a una velocidad que dé la impresión al usuario de que la operación aún se sigue realizando, por ejemplo cada 2 segundos.</p>	<p>RNC4: Se debe de mostrar un mensaje de progreso cuando el sistema está ejecutando la acción de Guardando un Examen. Y al finalizar la acción debe de mostrar un mensaje del estado de la acción que puede ser “Se ha guardado correctamente los datos del examen” ó “Registro Incorrecto”.  RND4: Se debe de mostrar el progreso cuando se está exportando a Excel. Y al finalizar debe de mostrar un mensaje del estado de la acción “Su archivo ha sido</p>





	<p>exportado correctamente”</p> <p>RNE1.3 Una vez dado por finalizado la corrección de exámenes se debe de mostrar un mensaje de progreso de la acción guardado de registros, y al finalizar la acción mostrar un mensajes de “Se guardó correctamente su corrección de examen” o “Registro Incorrecto”</p> <p>RNG3.1 Cuando el alumno envíe el examen se debe de mostrar su progreso de envío, y al finalizar mostrar un mensaje de “Su examen ha sido enviado correctamente” o “Envío de Examen Incorrecto”</p> <p>RNG3.2 Cuando el Alumno seleccione ver sus respuestas de cada examen se mostrará su progreso al mostrar dicha acción.</p> <p>RNH2 Cuando el alumno selecciones un tipo de reporte, se le mostrará el progreso para cargar los datos de dicho reporte.</p> <p>RNH3.1 Una vez rendido el examen se debe de mostrar el progreso mientras se carga la vista de su examen realizado.</p> <p>RNH3.2 Al solicitar la exportación del examen realizado, se debe de mostrar el progreso de generación de pdf, y al finalizar debe de mostrar un mensaje del estado de la acción “Su archivo ha sido exportado correctamente”.</p>
--	---

## 2. WARNING

<b>IDENTIFICADOR</b>
<b>Nombre:</b> Warning
<b>Familia:</b> Feedback
<b>Alias:</b> Warning
<b>PROBLEMA</b>

Que información se debe elicitar y especificar para solicitar al usuario la confirmación de la acción solicitada en caso de que esta tenga consecuencias irreversibles.	
<b>CONTESTO</b>	
Cuando una acción que tiene serias consecuencias es solicitado por el usuario.	
<b>SOLUCION</b>	
<b>Guía de Obtención de los Mecanismo de Usabilidad</b>	
<b>Temas a discutir con los Stakeholders</b>	<b>Temas a discutir con los Stakeholders</b>
<p>1.1 Ver qué acciones se pueden realizar y cuáles serían sus consecuencias.</p>	<p><b>RA1</b> El docente se identificará en el sistema con el número de su DNI y una contraseña.  <b>RA2</b> El alumno se identificará en el sistema con el número de su DNI y una contraseña  <b>RC1.6</b> habilitar o deshabilitar el modo random para asignación de preguntas  <b>RC1.7</b> Habilitar la opción de publicación automática de notas  <b>RC2.5</b> Guardar las preguntas  <b>RC3.2</b> El deberá seleccionar el tipo de calificativo a mostrar ya sea vigesimal (0 a 20) o por calificación por letras.  <b>RC4</b> Guardar Examen  <b>RE1.2</b> Finalizado la corrección de la evaluación confirma el guardar la evaluación  <b>RE1.3</b> Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación  <b>RE2</b> El docente podrá habilitar y deshabilitar el publicar las respuestas correctas de la evaluación  <b>RF1</b> El docente visualiza todos los exámenes ya corregidos y sus estados de publicación, en caso de que no esté habilitado la opción de publicación de notas este podrá programar la fecha publicacion de las notas  <b>RG1</b> El alumno podrá visualizar el tiempo designado para la prueba, así como un cronómetro que le indicara el tiempo que le queda para finalizar la prueba  <b>RG3.1</b> El alumno selecciona el botón de enviar examen.</p>
<p>a. ¿Qué información se le presentara al usuario cuando quiera confirmar una acción?                  Se mostrara las consecuencia que tendrá dicha acción y las alternativas que se le dará al usuario</p>	<p><b>RNA1:</b> Se debe mostrar un mensaje de alerta cuando el docente haya ingresado incorrectamente su usuario y contraseña.  <b>RNA2:</b> Se debe mostrar un mensaje de alerta cuando el alumno haya ingresado incorrectamente su usuario y contraseña.  <b>RNC1.6</b> Se debe de mostrar un mensaje que</p>



si el usuario realmente quiere deshabilitar el modo random.

**RNC1.7** Se debe de mostrar un mensaje que si el usuario realmente quiere habilitar la opción de publicación automática de notas

**RNC2.5** Guardar las preguntas

**RNC3.2** El deberá seleccionar el tipo de calificativo a mostrar (0 a 20 vigesimal) o por calificación por letras.

**RNC4.1** Se debe mostrar un mensaje que informe al usuario que hay datos que son requeridos para completar la tarea de Guardar Examen y están incompletos.

**RNC4.2** Se debe mostrar un mensaje que informe al usuario para confirmar Guardar Examen o si realmente desea guardar el examen.

**RNE1.2** Se le debe mostrar un mensaje al usuario si desea finalizar la corrección de la evaluación confirma el guardar la evaluación.

**RNE2** Se debe de mostrar un mensaje donde el usuario confirme que desea habilitar o deshabilitar el publicar las respuestas correctas de la evaluación.

**RNF1** Se le mostrar un mensaje informando al usuario cuando la fecha para la publicación de notas es incorrecta.

**RNG1** Cuando el usuario esté rindiendo el examen, se le mostrará mensaje “El examen está por culminar”, se le mostrará 5 minutos antes y 2 minutos antes de que culmine el examen.

**RNG3.1** Se le mostrará al alumno un mensaje de confirmación cuando selecciona el botón de enviar examen, el mensaje sera de la siguiente forma “Desea enviar el examen”.

**RNW1** Los mensajes deben ser

	<p>presentados mediante un modal de alerta, el cual debe de tener una presentación uniforme, de preferencia de una tonalidad roja, de acuerdo a los colores de la institución educativa.</p> <p><b>RNW2</b> El icono utilizado para este tipo de mensaje debe ser uniforme para todas las alertas.</p>
--	--

### 3. WIZARD

#### a. Step by Step

IDENTIFICADOR	
<b>Nombre:</b> Step BY Step	
<b>Familia:</b> Wizard	
<b>Alias:</b> Step by Step	
PROBLEMA	
Que información se debe elicitarse y especificar para proporcionar a los usuarios el mecanismo paso a paso	
CONTESTO	
Cuando un usuario novel necesita realizar una acción compleja y poco frecuente que consiste en varios pasos en los cuales se deben tomar ciertas decisiones.	
SOLUCION	
Guía de Obtención de los Mecanismo de Usabilidad	
Temas a discutir con los Stakeholders	Temas a discutir con los Stakeholders
<p>1.1 ¿Qué tareas requieren varios pasos para ejecutarse? ¿Cuáles son esos pasos?</p> <p>1.2 ¿Qué información es necesaria para el usuario en cada tarea? ¿Cuál es la mejor forma para preguntar por ella?</p>	<p>RNC1 Tarea: Crear Exámenes, que tiene varios pasos para ejecutarse, la cuales son:                      RNC1.1 Programación del Examen                      RNC1.2 Registro de preguntas                      RNC1.3 Programación de la vista de resultados                      RNC1.4 Guardar examen</p> <p>RNW1 El sistema debe de mostrar el paso en el que se encuentra el usuario para completar dicha tarea.</p>
<p>2.1 ¿Cuántos pasos son necesarios para culminar una tarea?</p> <p>2.2 ¿Cuál es la mejor forma de presentar estos pasos al usuario?</p>	<p>RNW2 Para concretar la tarea de creación de Exámenes el usuario debe completar 4 pasos.</p> <p>RNW3 Para las tareas que requieren varios</p>



	pasos, se debe de hacer uso de un Wizar, step by setp, que guíe al usuario en los pasos a seguir y los campos a completar la tarea.
3.1 ¿Cuáles serán las opciones de navegación hacia atrás en cada uno de los pasos, ya sea al paso anterior o al comienzo de todas las tareas?	RNW4 Los pasos de navegación se deben de realizar a través de botones que indican en los pasos a seguir, quienes serán los encargados mantener la navegabilidad entre los pasos, solo se podrá retroceder y adelantar en aquellos pasos que han sido concretados.

#### 4. USER INPUT ERROR PREVENTION

##### a. Structured Text Entry

IDENTIFICADOR	
<b>Nombre:</b> Structured Text Entry	
<b>Familia:</b> User Input Error Prevention	
<b>Alias:</b> Form/ Field validation	
PROBLEMA	
Que información se debe elicitar y especificar para proporcionar texto de entrada de manera estructurada	
CONTESTO	
Cuando el sistema sólo puede aceptar información con un formato en específico.	
SOLUCION	
Guía de Obtención de los Mecanismo de Usabilidad	
Temas a discutir con los Stakeholders	
<p>1.1 ¿Dónde el usuario necesitara ingresar información con un formato específico?</p> <p>1.2 ¿Cómo guiar al usuario en la introducción de información que requiere un formato específico? (valores por defecto, uso de checklists o comboboxes)</p>	<p>RC1.1 Ingresar un título para evaluación</p> <p>RC1.2 Selecciona el tipo de evaluación</p> <p>RC1.3 Seleccionar la fecha en que se desarrollara el examen</p> <p>RC1.4 Ingresar la hora en que se rendirá el examen (Inicio y Fin).</p> <p>RC1.5 Cantidad de preguntas</p> <p>RC2.2 Ingresar el enunciado de la pregunta</p> <p>RC2.4 Asignar el puntaje que tendrá la pregunta</p> <p>RC3.1 El docente programa la fecha de visualización de la resolución de la evaluación.</p> <p>RE1.1 Al abrir el examen del alumno se le carga las preguntas con su determina respuesta, la cual el docente podrá otorgar una calificativo</p> <p>RF1 El docente visualiza todos los exámenes ya corregidos y sus estados de publicación,</p>



	<p>en caso de que no esté habilitado la opción de publicación de notas este podrá programar la fecha publicación de las notas</p> <p>RG2.3 Juicio Crítico, el alumno debe de ingresar su opinión o respuesta argumentativa</p>
--	--

## 5. Undo y Cancel

### a. Abort Operation

IDENTIFICADOR	
<b>Nombre:</b> Abort Operation	
<b>Familia:</b> Undo / Cancel	
<b>Alias:</b> Emergency exit Go back to a safe place Go back Prominent cancel	
PROBLEMA	
Qué información se debe elicitarse y especificar para proporcionar al usuario el mecanismo de abortar operación.	
CONTEXTO	
Cuando el usuario debe salir de una aplicación o abortar la ejecución de un servicio.	
SOLUCIÓN	
Guía de Obtención de los Mecanismos de Usabilidad:	
Temas a discutir con los stakeholders	
1.1 ¿Necesitará el usuario la opción de salir de la aplicación?	RNUC1 Al usuario se le debe presentar la opción de salir para cada acción que este realice
1.2 En caso afirmativo, ¿Dónde y cómo se mostrará esta opción al usuario?	RNUC2 Cada vez que el usuario confirme la opción salir, se le redirigirá a la pestaña anterior, en la que estuvo.
2.1 ¿Qué servicios requerirán una opción de Cancelar? Se debe prestar especial atención a aquellos servicios que requieren varios pasos. Además, los servicios en los que el usuario es preguntado o se solicita información deben ser potencialmente cancelables.	RC4 Guardar Examen RD4 Los Reportes dados a los docentes deben de ser exportados a Excel RE1.2 Finalizado la corrección de la evaluación confirma el guardar la evaluación RE1.3 Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación
2.2 ¿Cómo se presentará la opción de cancelar al usuario (se debe tener en cuenta que la forma más común es mediante un botón de cancelar)?	RG3.1 El alumno selecciona el botón de enviar examen
2.3 ¿Cuál será el estado al que vaya el	RG3.2 Una vez enviado el examen el alumno solo podrá visualizar la evaluación con las



<p>sistema cuando se ejecute la opción de cancelar?</p>	<p>respuestas dadas RH2 El alumno puede visualizar un reporte de los resultados obtenidos por cada examen RH3 Una vez rendido el examen el alumno pueda visualizar sus respuestas y pueda descargar en pdf.  RNUC3: La opción salir o cancelar se presentará en todas las opciones en que el usuario tenga acceso.  RNUC4 Cuando el usuario cancele una acción el sistema volverá a su estado normal, conforme estaba desde antes que se ejecute la acción que se canceló.</p>
<p>3.1 ¿Qué actividades necesitaran más de 10 segundos para culminar su ejecución? 3.2 ¿Cómo se va a relacionar con el mecanismo Progres Feedback?</p>	<p>RNUC4 Los servicios que se requieren más de 10 segundos para su ejecución son los requisitos: <b>RC4 Guardar Examen, RE Corrección de exámenes, RF: Publicación de notas, RG: Dar Exámenes,etc.</b>  RNUC5: Cada una de estas acciones se relaciona con el Progress Feedback, la cual se le va a mostrar el proceso en que se está realizando dicha acción.</p>

## 6. HELP

### a. Multilevel Help

<b>IDENTIFICADOR</b>	
<b>Nombre:</b> Multilevel Help	
<b>Familia:</b> Help	
<b>Alias:</b> Multilevel Help	
<b>PROBLEMA</b>	
Que información se debe elicitarse y especificar para proporcionar al usuario el mecanismo de información multilevel.	
<b>CONTEXTO</b>	
Es utilizada cuando la aplicación a desarrollar cuenta con actividades complejas las cuales algunos usuarios necesitaran ayuda para interactuar con el sistema.	
<b>SOLUCIÓN</b>	
<b>Guía de Obtención de los Mecanismos de Usabilidad:</b>	
<b>Temas a discutir con los Stakeholders</b>	
1.3 ¿Qué tan complejo es realizar un examen a través del sistema?	<b>RC1 Programación del examen</b> <b>RC2 Registro de preguntas</b>
1.4 ¿Qué tipos de usuarios utilizaran el	<b>RD8</b> Los reportes en caso de usar colores



<p>sistema?</p> <p>1.5 ¿Cuáles son las actividades que el usuario necesitara ayuda?</p> <p>1.6 ¿Cuál sería la ayuda más eficiente para cada tarea?</p>	<p>para mostrar el estado de un alumno o la nota lograda, es necesario manejar una leyenda de colores</p> <p><b>RE Corrección de exámenes</b></p> <p><b>RF: Publicación de notas</b></p> <p><b>RG: Dar Exámenes</b></p> <p><b>RG2.4 Relacionar</b></p> <p>RH4 Los reportes en caso de usar colores para mostrar el estado de un alumno o la nota lograda, es necesario manejar una leyenda de colores</p> <p>RNH1: Los usuarios a utilizar el sistema la gran mayoría son usuarios noveles.</p> <p>RNH2: Cuando el usuario solicite ayuda en el sistema, se le debe de mostrar una guía de cómo se debe realizar dicha acción.</p>
--	--

## 7. USER PROFILE

### a. PERSONAL OBJECT SPACE

<b>IDENTIFICADOR</b>	
<b>Nombre:</b> Personal Object Space	
<b>Familia:</b> User Profile	
<b>Alias:</b> Personal Object Space	
<b>PROBLEMA</b>	
Qué información se debe elicitarse y especificar para proporcionar al usuario el mecanismo de personalizar la localización de los elementos de la interfaz.	
<b>CONTEXTO</b>	
Cuando la interfaz del sistema es compleja y tiene muchos iconos que se pueden organizar de diferentes maneras.	
<b>SOLUCIÓN</b>	
<b>Guía de Obtención de los Mecanismos de Usabilidad:</b>	
<b>Temas a discutir con los stakeholders</b>	
<p>1.1 ¿Los usuarios podrán organizar la ubicación de los elementos visibles de las interfaces?</p> <p>1.2 ¿El sistema habilitará esta posibilidad de organización para todos los usuarios?</p> <p>1.3 ¿Qué elementos de la interfaz podrá ordenar el usuario?</p>	<p>RNUP1 Los usuarios podrán personalizar la posición del menú.</p> <p>RNUP2 Los usuarios podrán elegir un color de tema para la presentación del aplicativo.</p>





### 5.4.3. DEFINICIÓN DE REQUERIMIENTOS

#### 5.4.3.1. DEFINICIÓN DE REQUERIMIENTOS FUNCIONALES DE USABILIDAD

Se estableció los requerimientos de forma de uso generados y como estos afectan a los requerimientos genéricos establecidos para el sistema, así mismo identificamos los nuevos requerimientos de usabilidad funcional generados. Esta actividad es compartida para ambas metodologías.

#### 1. FEEDBACK

##### 1.1. SYSTEM STATUS FEEDBACK

Según (Ignacio Panach, Juristo, Valverde, & Pastor, 2015) El SYSTEM STATUS FEEDBACK tiene como el objetivo de informar al usuario de cambios que se realizan en el sistema de manera inesperada, ya sean por fallos durante la ejecución de un servicio o debido a que no existen los recursos suficiente, etc. Para este mecanismo de usabilidad se han obtenido los siguientes requisitos:

Requerimientos de Forma de Uso	Requerimientos Afectados	Nuevos Requerimientos Funcionales de Usabilidad
<p><b>RNBSS1</b> Cada mensaje de System Status Feedback debe de estar enlazada a un icono que represente el tipo alerta.</p> <p><b>RNBSS2</b> Los mensajes se le mostrarán a través de una ventana emergente (modal).</p> <p><b>RNBSS1</b> Cada mensaje de System Status Feedback</p>	<p><b>RB1</b> Alerta de conexión con el servidor de base de datos.</p>	<p><b>RNB1.1</b> Mensaje: No se puede conectar al servidor.</p>
	<p><b>RB2</b> Alerta de conexión con el servidor.</p>	<p><b>RNB1.2</b> Mensaje: Pagina web no encontrada</p>
	<p><b>RB3</b> Alertas para los exceptions de los try catch.</p>	<p><b>RNB1.3</b> Los mensajes generados por excepción tendrán la siguiente estructura: “Mensaje de Alerta” seguido de la descripción de la incidencia percibida, valor que es retornado por el manejador de excepciones.</p>



<p>debe de estar enlazada a un icono que represente el tipo alerta.</p> <p><b>RNBSS3</b> La ventana de alerta o emergente se debe ubicar en la parte central de la interfaz del usuario.</p>		
--	--	--

Tabla 40System Status Feedback Elaboración propia.

## 1.2. INTERACTION FEEDBACK

Según (Ignacio Panach, Juristo, Valverde, & Pastor, 2015) El INTERACION FEEDBACK tiene como objetivo informar al usuario que sus acciones realizadas en el sistemas han sido recibidas, mayormente es utilizada cuando el usuario realiza una interacción ya sea con la pulsación de una tecla o el movimiento del mouse, que esté relacionado a un proceso o subproceso del sistema.

Requerimiento de Forma de uso	Requerimientos Afectados	Nuevos Requerimientos Funcionales de Usabilidad
<p><b>RNIF1</b> Cuando el usuario realiza un evento de interacción con botones comprometidos con acciones principales, el sistema deberá responder de forma diferenciadora a este evento.</p>	<p><b>RC2.5</b> Guardar las preguntas</p>	<p><b>RNC2.5</b> Al clic del ratón o movimiento de ratón, sobre el botón “Guardar las preguntas” este deberá responder a la interacción, cambiando de tamaño.</p>
	<p><b>RC4</b> Guardar Examen</p>	<p><b>RNC4</b>Al clic del ratón o movimiento de ratón, sobre el botón “Guardar Examen” este deberá responder a la interacción, cambiando la intensidad del color del botón.</p>
	<p><b>RE1.2.</b> Finalizado la corrección de la evaluación confirma el guardar la evaluación.</p>	<p><b>RNE1.2</b> Al clic del ratón o movimiento de ratón, sobre el botón “Guardar Corrección” este deberá responder a la interacción, cambiando la intensidad del color</p>



		del botón.
	<b>RE1.3</b> Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación	<b>RNE1.3</b> Al clic del ratón o movimiento de ratón, sobre el botón “Finalizar Corrección” este deberá responder a la interacción, cambiando la intensidad del color del botón..
	<b>RG3.1</b> El alumno selecciona el botón de enviar examen	<b>RNG3.1</b> Al clic del ratón o movimiento de ratón, sobre el botón “Enviar examen” este deberá responder a la interacción, cambiando la intensidad del color del botón.

### 1.3. PROGRESS FEEDBACK

Según (Ignacio Panach, Juristo, Valverde, & Pastor, 2015) El PROGRESS FEDBACK tiene como objetivo de informar al usuario el progreso de una actividad para su finalización. Es utilizada para aquellas actividades cuya ejecución requiera más de dos segundos.

Requerimientos de Forma de Uso	Requerimientos Afectados	Nuevos Requerimientos Funcionales de Usabilidad
<b>RNPFG2.</b> Los Progress feedback se deben de mostrar a través de una venta ventana emergente (modal).	<b>RC4.</b> Guardar Examen	<b>RNC4:</b> Se debe de mostrar un mensaje de progreso cuando el sistema está ejecutando la acción de Guardando un Examen. Y al finalizar la acción debe de mostrar un mensaje del estado de la acción que puede ser “Se ha guardado correctamente los datos del examen” o “Registro Incorrecto”
	<b>RD4.</b> Los Reportes generados por docentes deben de ser exportados a Excel. (por cada reporte)	<b>RND4:</b> Se debe de mostrar un del progreso cuando se está exportando a Excel. Y al finalizar debe de mostrar un mensaje del estado de la acción “Su archivo ha sido exportado correctamente”



<p><b>RE1.3</b> Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación.</p>	<p><b>RNE1.3</b> Una vez dado por finalizado la corrección de exámenes se debe de mostrar un mensaje de progreso de la acción guardado de registros, y al finalizar la acción mostrar un mensajes de “Se guardó correctamente su corrección de examen” o “Registro Incorrecto”</p>
<p><b>RG3.1</b> El alumno selecciona el botón de enviar examen.</p>	<p><b>RNG3.1</b> Cuando el alumno envíe el examen se debe de mostrar su progreso de envío, y al finalizar mostrar un mensaje de “Su examen ha sido enviado correctamente” o “Envío de Examen Incorrecto”</p>
<p><b>RG3.2</b> Una vez enviado el examen el alumno solo podrá visualizar la evaluación con las respuestas dadas.</p>	<p><b>RNG3.2</b> Cuando el Alumno seleccione ver sus respuestas de cada examen se mostrará su progreso al mostrar dicha acción.</p>
<p><b>RH2</b> El alumno puede visualizar reportes de sus logros.</p>	<p><b>RNH2</b> Cuando el alumno selecciones un tipo de reporte, se le mostrará el progreso para cargar los datos de dicho reporte.</p>
<p><b>RH3</b> Una vez rendido el examen el alumno pueda visualizar sus respuestas y pueda descargar en pdf.</p>	<p><b>RNH3.1</b> Una vez rendido el examen se debe de mostrar el progreso mientras se carga la vista de su examen realizado.</p>
	<p><b>RNH3.2</b> Al solicitar la exportación del examen realizado, se debe de mostrar el progreso de generación de pdf, y al finalizar debe de mostrar un mensaje del estado de la acción “Su archivo ha sido exportado correctamente”.</p>

#### 1.4. WARNING

Según (Ignacio Panach, Juristo, Valverde, & Pastor, 2015) El mecanismo de usabilidad WARNING tiene la funcionalidad de informar al usuario cuando la acción realizada es incorrecta o que dicha acción puede producir un cambio en el sistema, evitando así que el usuario



cometa errores en el manejo del sistema que pueden tener consecuencias irreversibles.

<b>Requerimientos de Forma de Uso</b>	<b>Requerimientos Afectados</b>	<b>Nuevos Requerimientos Funcionales de Usabilidad</b>
<p><b>RNW1</b> Los mensajes deben ser presentados mediante un modal de alerta, el cual debe de tener una presentación uniforme, de preferencia de una tonalidad roja, de acuerdo a los colores de la institución educativa.</p>	<p><b>RA1</b> El docente se identificará en el sistema con el número de su DNI y una contraseña.</p>	<p><b>RNA1</b> El sistema deberá mostrar un mensaje de alerta cuando el docente ingrese incorrectamente su usuario y contraseña.</p>
	<p><b>RA2</b> El alumno se identificará en el sistema con el número de su DNI y una contraseña.</p>	<p><b>RNA2</b> El sistema deberá mostrar un mensaje de alerta cuando el alumno ingrese incorrectamente su usuario y contraseña.</p>
	<p><b>RC1.6</b> habilitar o deshabilitar el modo random para asignación de preguntas.</p>	<p><b>RNC1.6</b> Se debe de mostrar un mensaje que si el usuario realmente quiere deshabilitar el modo random.</p>
	<p><b>RC1.7</b> Habilitar la opción de publicación automática de notas.</p>	<p><b>RNC1.7</b> Se debe de mostrar un mensaje que si el usuario realmente quiere habilitar la opción de publicación automática de notas.</p>
	<p><b>RC2.5</b> Guardar las preguntas.</p>	<p><b>RNC2.5</b> Guardar las preguntas.</p>
	<p><b>RC3.2</b> El deberá seleccionar el tipo de calificativo a mostrar ya sea vigesimal (0 a 20) o por calificación por letras.</p>	<p><b>RNC3.2</b> El deberá seleccionar el tipo de calificativo a mostrar (0 a 20 vigesimal) o por calificación por letras.</p>
	<p><b>RC4</b> Guardar Examen.</p>	<p><b>RNC4.1</b> Se debe mostrar un mensaje que informe al usuario que hay datos que son requeridos para completar la tarea de Guardar Examen y están incompletos.</p>
	<p><b>RE1.2</b> Finalizado la corrección de la evaluación confirma el guardar la evaluación.</p>	<p><b>RNC4.2</b> Se debe mostrar un mensaje que informe al usuario para confirmar Guardar Examen o si realmente desea guardar el examen.</p>



<p><b>RNW2</b> El icono utilizado para este tipo de mensaje debe ser uniforme para todas las alertas.</p>	<p><b>RE1.3</b> Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación.</p>	<p><b>RNE1.2</b> Se le debe mostrar un mensaje al usuario si desea finalizar la corrección de la evaluación confirma el guardar la evaluación.</p>
	<p><b>RE2</b> El docente podrá habilitar y deshabilitar el publicar las respuestas correctas de la evaluación.</p>	<p><b>RNE2</b> Se debe de mostrar un mensaje donde el usuario confirme que desea habilitar o deshabilitar el publicar las respuestas correctas de la evaluación.</p>
	<p><b>RF1</b> El docente visualiza todos los exámenes ya corregidos y sus estados de publicación, en caso de que no esté habilitado la opción de publicación de notas este podrá programar la fecha publicación de las notas.</p>	<p><b>RNF1</b> Se le mostrar un mensaje informando al usuario cuando la fecha para la publicación de notas es incorrecta.</p>
	<p><b>RG1</b> El alumno podrá visualizar el tiempo designado para la prueba, así como un cronómetro que le indicara el tiempo que le queda para finalizar la prueba.</p>	<p><b>RNG1</b> Cuando el usuario esté rindiendo el examen, se le mostrará mensaje “El examen está por culminar”, se le mostrará 5 minutos antes y 2 minutos antes de que culmine el examen.</p>
	<p><b>RG3.1</b> El alumno selecciona el botón de enviar examen.</p>	<p><b>RNG3.1</b> Se le mostrará al alumno un mensaje de confirmación cuando selecciona el botón de enviar examen, el mensaje sera de la siguiente forma “Desea enviar el examen”</p>

## 2. WIZAR

### 2.1. STEP BY STEP

(Ignacio Panach, Juristo, Valverde, & Pastor, 2015) El mecanismo de usabilidad STEP BY STEP es utilizada para ayudar a los diferentes usuarios en las diversas tareas que este realiza y que son demasiadas complejas o que requieren una gran



cantidad de información a ingresar por el usuario, para ello STEP BY STEP ayuda a subdividir el proceso en varios pasos indicando al usuario cuanto le falta para terminar y así no se vuelva tedioso interactuar el sistema.

Requerimientos de Forma de Uso	Requerimientos Afectados	Nuevos Requerimientos Funcionales de Usabilidad
<b>RNW3</b> Para las tareas que requieren varios pasos, se debe de hacer uso de un Wizar, step by setp, que guíe al usuario en los pasos a seguir y los campos a completar la tarea.	<b>RNC1</b> Tarea: Crear Exámenes, que tiene varios pasos para ejecutarse, la cuales son: <b>RNC1.1</b> Programación del Examen <b>RNC1.2</b> Registro de preguntas <b>RNC1.3</b> Programación de la vista de resultados <b>RNC1.4</b> Guardar examen	<b>RNW1</b> El sistema debe de mostrar el paso en el que se encuentra el usuario para completar dicha tarea.
		<b>RNW2</b> Para concretar la tarea de creación de Exámenes el usuario debe completar 4 pasos.
		<b>RNW4</b> Los pasos de navegación se deben de realizar a través de botones que indican en los pasos a seguir, quienes serán los encargados mantener la navegabilidad entre los pasos, solo se podrá retroceder y adelantar en aquellos pasos que han sido concretados.

### 3. USER INPUT ERROR PREVENTION

#### 3.1. STRUCTURED TEXT ENTRY

Panach (2010) Este mecanismo de usabilidad es utilizado para especificar entradas de datos en la interfaz con un formato específico.

Nuevo Requerimientos de Forma de Uso	Requerimientos Afectados
<b>RNSTE 1</b> Orientar al usuario para la introducción de información para ello se requiere el uso de un formato específico: Tooltip.	<b>RC1.1</b> Ingresar un título para evaluación <b>RC1.2</b> Selecciona el tipo de evaluación <b>RC1.3</b> Seleccionar una fecha en la que se realizara el examen. <b>RC1.4</b> Escoger la hora de inicio y la hora fin para rendir la evaluación. <b>RC1.5</b> Cantidad de preguntas <b>RC2.2</b> Ingresar el enunciado de la pregunta <b>RC2.4</b> Asignar el puntaje que tendrá la pregunta



	<p><b>RC3.1</b> El docente programa la fecha de visualización de la resolución de la evaluación.</p> <p><b>RE1.1</b> Al abrir el examen del alumno se le carga las preguntas con su determina respuesta, la cual el docente podrá otorgar una calificativo</p> <p><b>RF1</b> El docente visualiza todos los exámenes ya corregidos y sus estados de publicación, en caso de que no esté habilitado la opción de publicación de notas este podrá programar la fecha publicación de las notas</p> <p><b>RG2.3</b> Juicio Crítico, el alumno debe de ingresar su opinión o respuesta argumentativa.</p>
--	--

#### 4. UNDO Y CANCEL

##### 4.1. ABORT OPERATION

Panach (2010) Este mecanismo de usabilidad es utilizado para representar la funcionalidad de cancelar una acción antes de que finalice su ejecución.

<b>Nuevos Requerimientos Funcionales de Usabilidad.</b>	<b>Requerimientos Afectados</b>
<p><b>RNUC1</b> La opción salir de la aplicación debe de estar visible en todo momento, en la parte derecha superior.</p> <p><b>RNUC2</b> Al usuario se le debe presentar la opción abordar la ejecución de una acción, para cada acción que este realice y necesite confirmar, mediante un modal.</p> <p><b>RNUC3</b> Cada vez que el usuario aborde la ejecución de un servicio, se le redirigirá a la pestaña anterior, en la que estuvo.</p> <p><b>RNUC4</b> Cada vez que el usuario requiera salir de la aplicación, se culmina la sesión rediriéndolo al login.</p>	RC4 Guardar Examen.
	RD4 Los Reportes dados a los docentes deben de ser exportados a Excel.
	RE1.2 Finalizado la corrección de la evaluación confirma el guardar la evaluación.
	RE1.3 Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación.
	RG3.1 El alumno selecciona el botón de enviar examen.
	RG3.2 Una vez enviado el examen el alumno solo podrá visualizar la evaluación con las respuestas dadas.
	RH2 El alumno puede visualizar un reporte de los resultados obtenidos por cada examen.
RH3 Una vez rendido el examen el alumno pueda visualizar sus respuestas y pueda descargar en pdf.	





## 5. HELP

### 5.1 MULTILEVEL HELP

Panach (2010) Este mecanismo de usabilidad que introduce la funcionalidad en el sistema para ayudar al usuario sobre el funcionamiento del sistema. Haciendo énfasis a los tipos de usuario que interactuarán con el sistema.

<b>Nuevos Requerimientos Funcionales de Usabilidad.</b>	<b>Requerimientos Afectados</b>
<p><b>RNH1:</b> Los usuarios a utilizar el sistema la gran mayoría son usuarios noveles.</p> <p><b>RNH2:</b> Cuando el usuario solicite ayuda en el sistema, se le debe de mostrar una guía de cómo se debe realizar dicha acción.</p>	<b>RC1</b> Programación del examen
	<b>RC2</b> Registro de preguntas
	<b>RD8</b> Los reportes en caso de usar colores para mostrar el estado de un alumno o la nota lograda, es necesario manejar una leyenda de colores.
	<b>RE</b> Corrección de exámenes.
	<b>RF:</b> Publicación de notas.
	<b>RG:</b> Dar Exámenes. <b>RG2.4</b> Relacionar
	<b>RH4</b> Los reportes en caso de usar colores para mostrar el estado de un alumno o la nota lograda, es necesario manejar una leyenda de colores.

## 6. USER PROFILE

### 6.1. PERSONAL OBJECT SPACE

Panach (2010) Este mecanismo de usabilidad es utilizado para incorporar en una aplicación la funcionalidad, que posibilita al usuario modificar la posición de los elementos en la interfaz.

### Requerimientos Funcionales de Usabilidad



**RNUP1** Los usuarios podrán personalizar la posición del menú.

**RNUP2** Los usuarios podrán elegir un color de tema para la presentación del aplicativo.

## A. DEFINICIÓN DE REQUERIMIENTOS: NDT

La NDT considera esta fase como la fase de ingeniería de requisitos tiene como objetivo final definir un catálogo de requerimientos que defina la necesidad del sistema. En él, se deben establecer los requerimientos catalogados según su tipología y no entrar en ningún aspecto de detalle sobre cómo se solventará el desarrollo. A continuación se presenta los artefactos generados en esta etapa:

### 1. OBJETIVO DEL PROYECTO

La NDT estable como primera actividad establecer el objetivo del proyecto, el cual se ha definido de la siguiente manera: El objetivo principal de este proyecto de investigación es de automatizar el proceso de gestión de Exámenes en la Institución Educativa Particular Santa María de la Paz, ubicada en Pimentel de manera que, los docentes puedan crear, programar, corregir exámenes vía Internet, así como también los alumnos puedan realizar sus exámenes en línea. Así mismo poder agilizar el proceso de corrección de exámenes, poder obtener reportes del progreso y calificaciones de un alumno o de toda la clase.

### 2. DIAGRAMA OBJETIVOS DEL SISTEMA

El diagrama 3 plasma los objetivos del sistema. Los objetivos que tiene un círculo verde hacen referencia a los objetivos aprobados, los amarillo a los Objetivos Obligatorios y los que poseen un círculo anaranjado son Objetivos Propuestos.

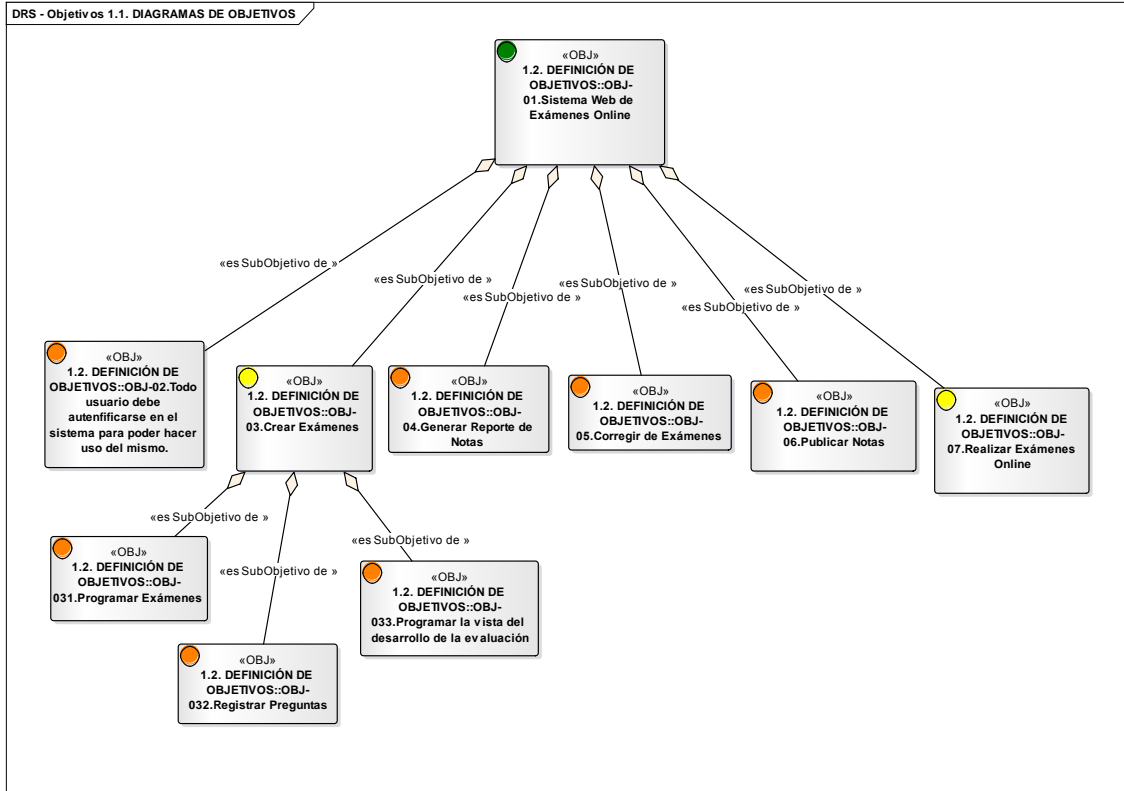


Diagrama 3 Diagrama de Objetivos del Sistema de Examen Online, NDT.

### 3. REQUISITOS DE ALMACENAMIENTO DE INFORMACIÓN

Según (IWT2, 2016, pág. 57) “Los requisitos de almacenamientos de información, junto con las nuevas naturalezas, determinar todas las necesidades de almacenamiento que se detectan durante la captura de requerimientos”.

En la tabla 33 podemos observar los diferentes mecanismos de usabilidad que se pudieron representar en los requisitos de almacenamiento de información.

Tabla 41 Definición de elementos a utilizar, para las alertas System Status Feedback

Tipo de Alerta	Estado	Icono
m Status Feedb	“No hay conexión al servidor de base de datos”	

Los mensajes generados por excepciones	
“No se ha podido encontrar la página web ”	

MECANISMO DE USABILIDAD	REQUERIMIENTO DE ALMACENAMIENTO DE INFORMACIÓN	DESCRIPCIÓN
System Status feedback	<p>DRS - Requisitos de Almacenamiento 3.1.1. DIA...</p> <div style="border: 1px solid black; padding: 5px;"> <p>«RA»</p> <p><b>3.1.3. DEFINICIÓN DE NUEVAS NATURALEZAS::RA-17.mensaje_alerta</b></p> <ul style="list-style-type: none"> <li>- descripcion: Texto</li> <li>- estado: Texto</li> <li>- icono: Texto</li> <li>- id: Entero {id}</li> <li>- tipo_alerta: Texto</li> </ul> </div>	Los diferentes mensajes de alertar que puedan ser presentados al usuario, deberán de ser almacenados para luego ser mostrados en el sistema cuando estos se requieran.
Warning		
Help	<p>DRS - Requisitos de Almacenamiento 3.1.1. DIAGRAMAS DE ...</p> <div style="border: 1px solid black; padding: 5px;"> <p>«RA»</p> <p><b>3.1.2. DEFINICIÓN DE REQUISITOS DE ALMACENAMIENTO::RA-11.categoría_contenido_ayuda</b></p> <ul style="list-style-type: none"> <li>- descripcion: Texto</li> <li>- estado: Logico</li> <li>- id: Entero {id}</li> </ul> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px;"> <p>«RA»</p> <p><b>3.1.2. DEFINICIÓN DE REQUISITOS DE ALMACENAMIENTO::RA-10.Ayuda</b></p> <ul style="list-style-type: none"> <li>- descripcion: Texto</li> <li>- id: Entero {id}</li> <li>- multimedia_ruta: Texto</li> <li>- titulo: Texto</li> </ul> </div>	El sistema deberá de brindar la opción de ayuda al usuario, para ello deberá de almacenar la información correspondiente para mostrar al usuario cuando este lo necesite.
User Profile	<p>DRS - Requisitos de Almacenamiento 3.1.1. DIAGRAMAS DE ...</p> <div style="border: 1px solid black; padding: 5px;"> <p>«RA»</p> <p><b>3.1.2. DEFINICIÓN DE REQUISITOS DE ALMACENAMIENTO::RA-16.espacio_personal_configuracion</b></p> <ul style="list-style-type: none"> <li>- choose_skin: Texto</li> <li>- estado_ayuda: Logico</li> <li>- id: Entero {id}</li> <li>- Right_Left: Texto</li> </ul> </div>	El sistema deberá almacenar la información correspondiente a la personalización que le da el usuario al sistema, en cuanto al Choose Skin (tema o color) según su preferencia así como la locación del menú, ya sea izquierda o derecha.



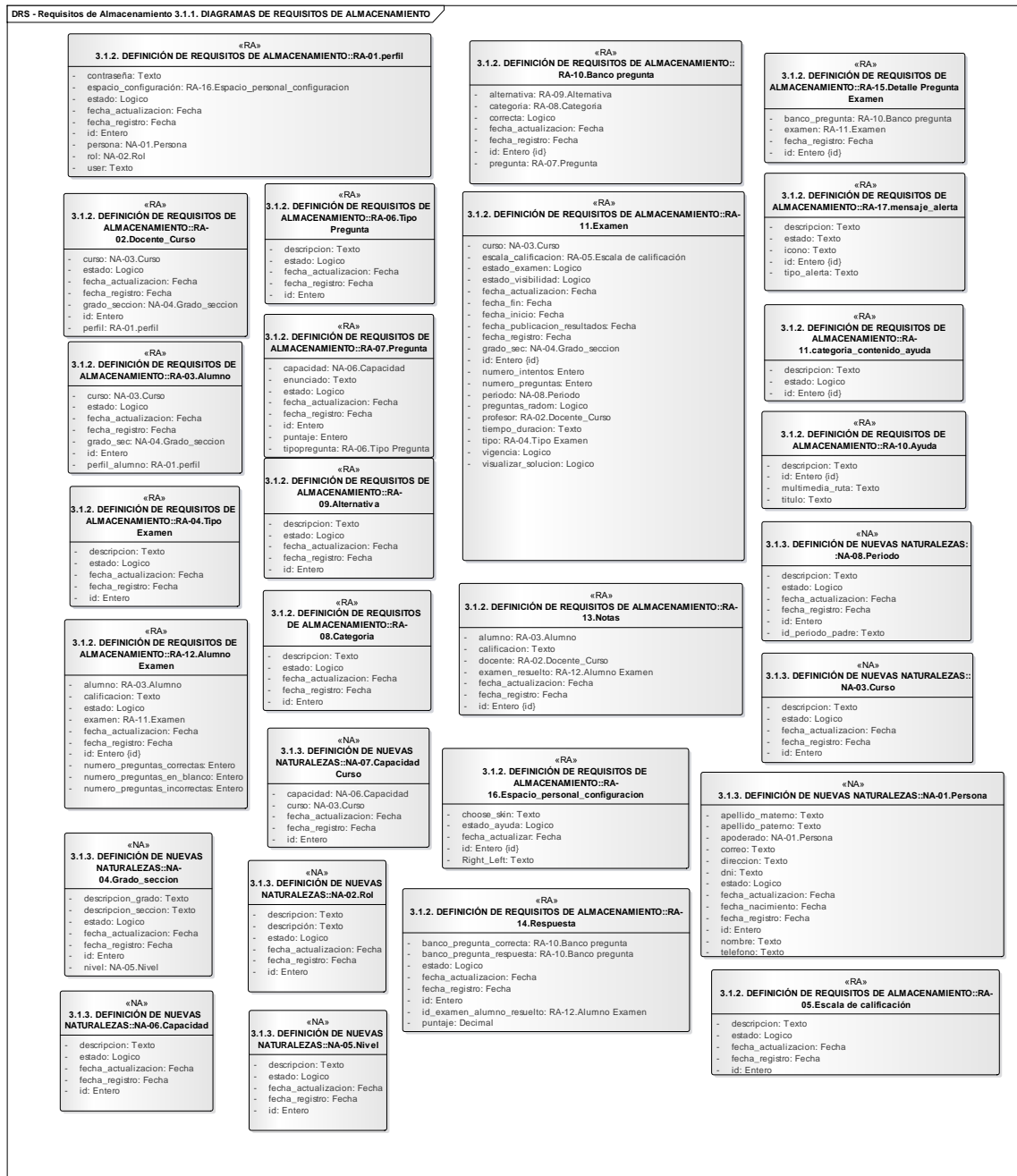


Diagrama 4 Diagrama de Requerimientos de almacenamiento de información del Sistema de Examen Online, NDT.

#### 4. DEFINICIÓN DE ACTORES

En el diagrama de actores define los actores del sistema mediante el artefacto AC. Los actores para este proyecto son: Alumno, Docente y Administrador.



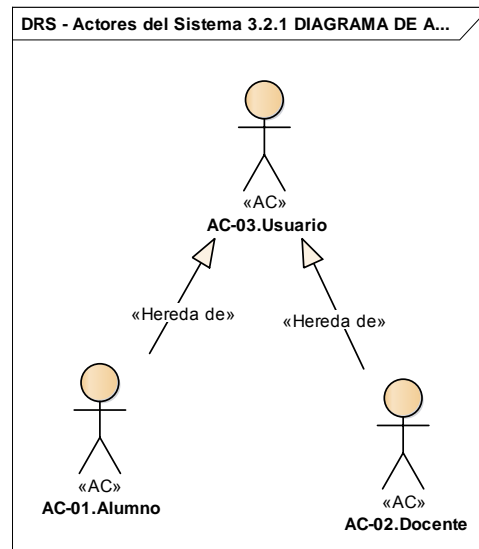


Diagrama 5 Diagrama Actores del Sistema de Examen Online, NDT.

## 5. DIAGRAMAS DE CASOS DE USO

A través de los diagramas de casos de usos se representa los diferentes requerimientos funcionales que los Stakeholders desean para el sistema, representado las relaciones entre los actores y las funcionalidades que requieren del sistema.

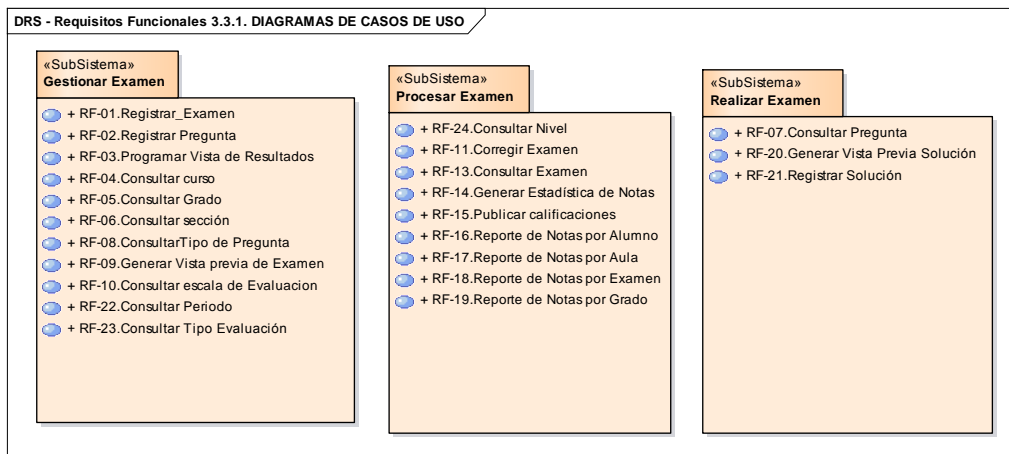


Diagrama 6 Diagrama de Requisitos funcionales del Sistema de Examen Online, NDT.

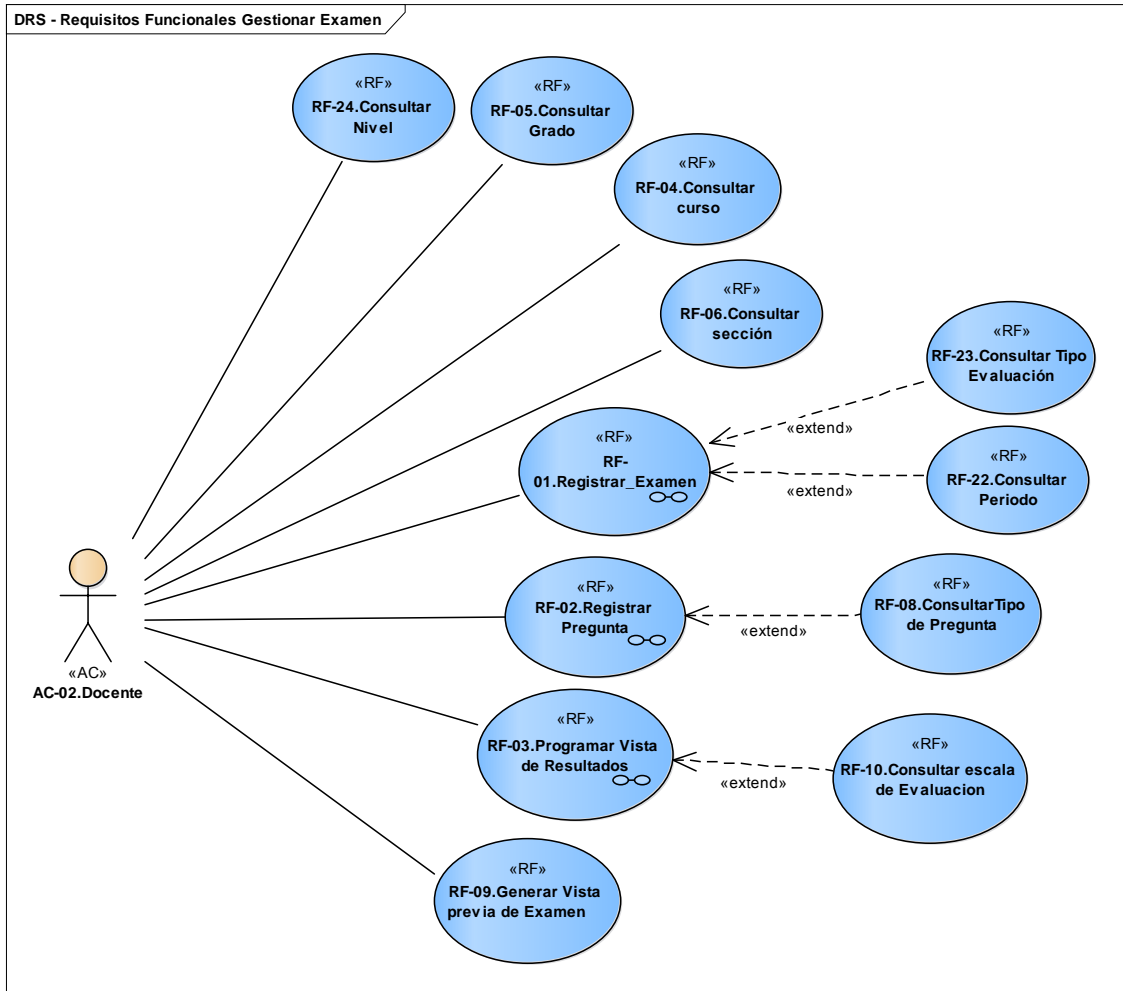


Diagrama 7Requisitos funcionales: Gestionar Exámenes NDT.

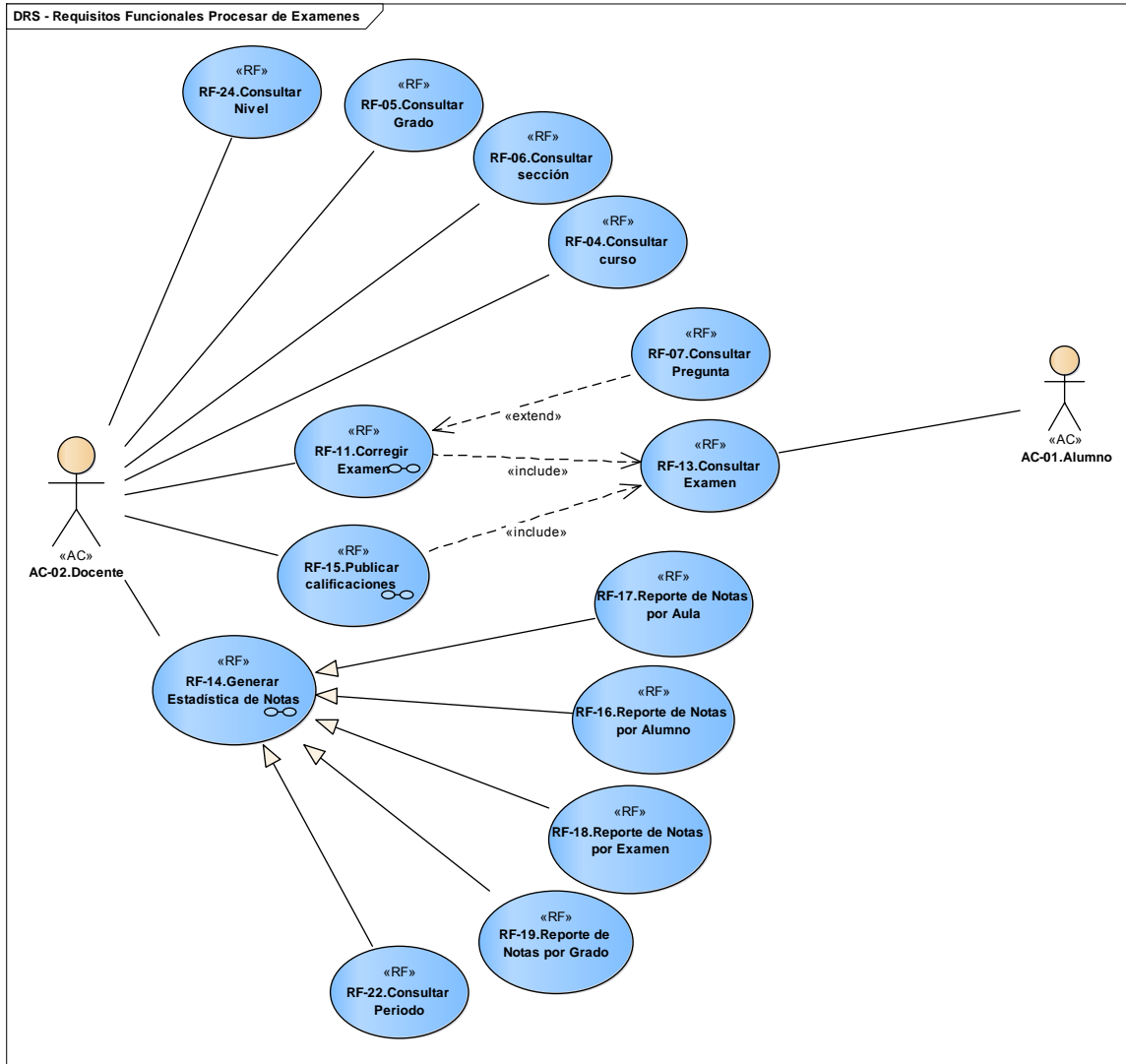


Diagrama 8Requisitos funcionales: Procesar Exámenes NDT.



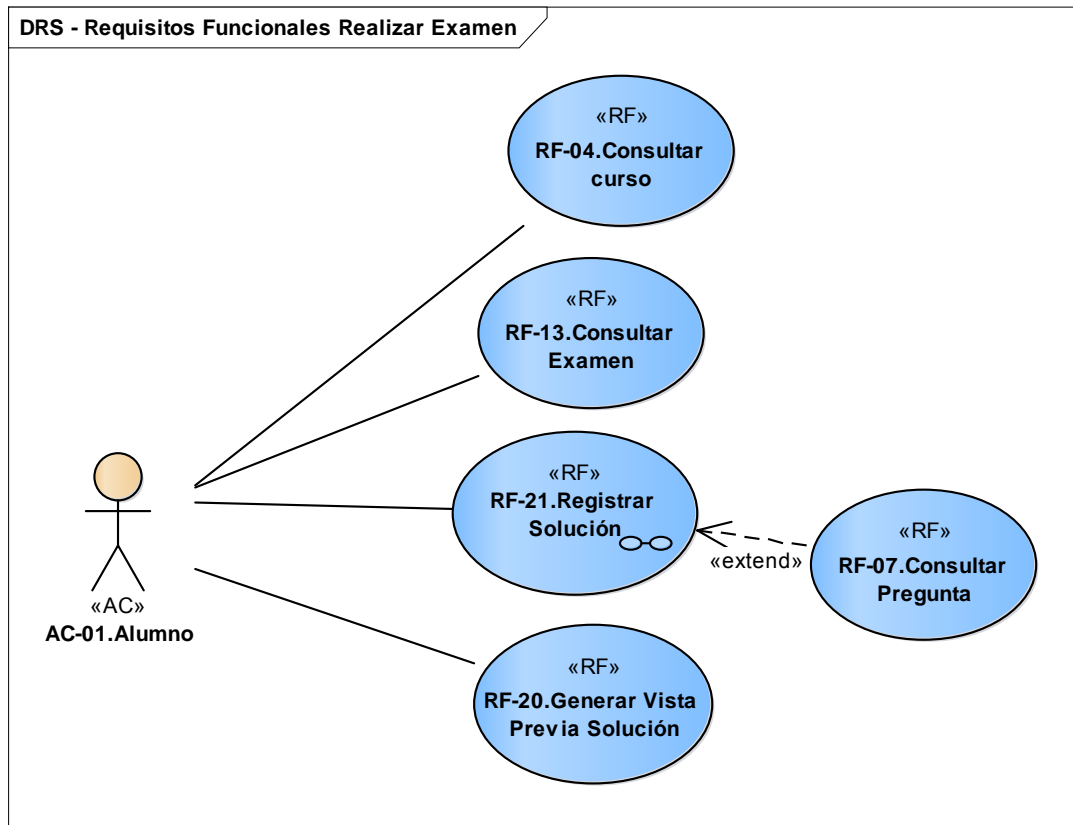


Diagrama 9 Requisitos funcionales: Realizar Examen, NDT.

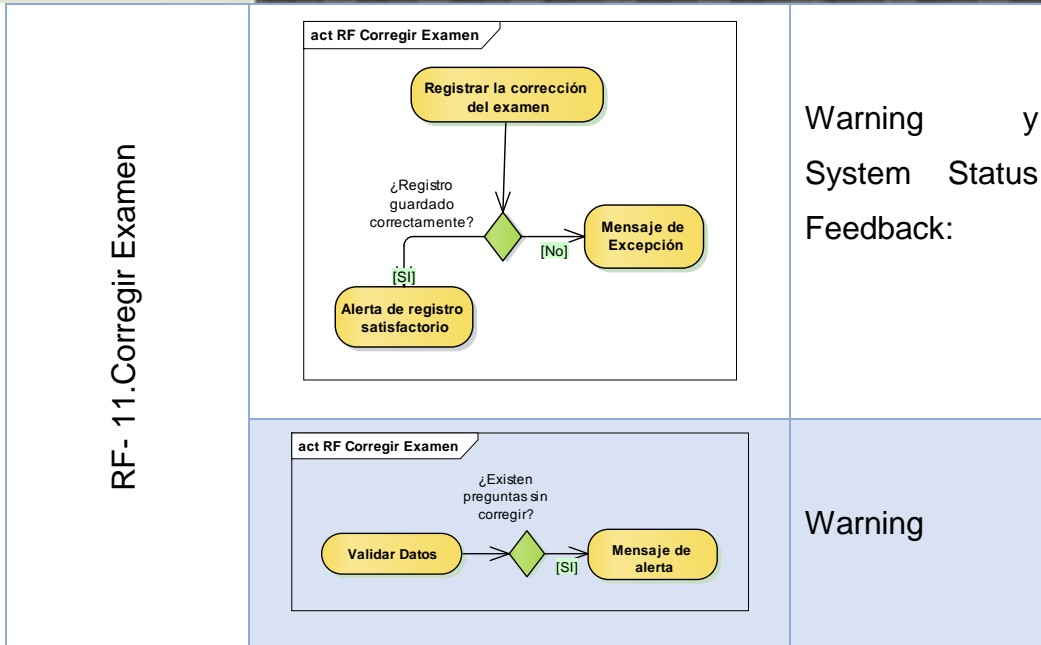
En esta etapa de la metodología NDT se han desarrollado también diagramas de actividad, los cuales modelan el comportamiento del sistema, y la forma en que estos comportamientos están relacionados en un flujo global del sistema (es decir, interacciones de elementos dinámicos).

En los diagramas de actividades se han incluido los requerimientos funcionales de usabilidad generados a partir de la aplicación de UPSEPS, que hacen referencia a los mecanismos de usabilidad:

- a. Warning, plasmando las peticiones de confirmación que el sistema debe de solicitar al usuario, cada vez que este realice una acción que tenga consecuencias irreversibles.
- b. System Status Feedback, plasmando los mensajes de información que debe de solicitar el sistema cuando se producen exceptions en el sistema.

Nombre del diagrama	Elementos del artefacto	Mecanismo de Usabilidad
RF-01 Registrar Examen	<pre> graph TD     A[Muestra un mensaje de confirmación] --&gt; B[Confirmar el proceso]     B -- Confirmacion --&gt; C{ }             </pre>	Warning
RF-02 Registrar Pregunta	<pre> graph TD     A[Validar la existencia de programación de examen temporal] --&gt; B{¿Existe el examen?}     B -- [No] --&gt; C[Mensaje "¿Desea crear examen?"]     C --&gt; D{¿Crear examen?}             </pre>	Warning
	<pre> graph TD     A[Mensaje si desea agregar otra categoria]             </pre>	Warning
	<pre> graph TD     A[Mensaje : "Desea agregar otra pregunta"] --&gt; B{¿Agregar otra pregunta?}             </pre>	Warning
	<pre> graph TD     A[Guardar temporalmente] --&gt; B{¿Registro guardado correctamente?}     B -- [Si] --&gt; C[Mensaje de alerta de registro correcto]     B -- [No] --&gt; D[Mensaje de Excepción]             </pre>	Warning y System Status Feedback:





A continuación presentaremos los principales diagramas de Actividades.



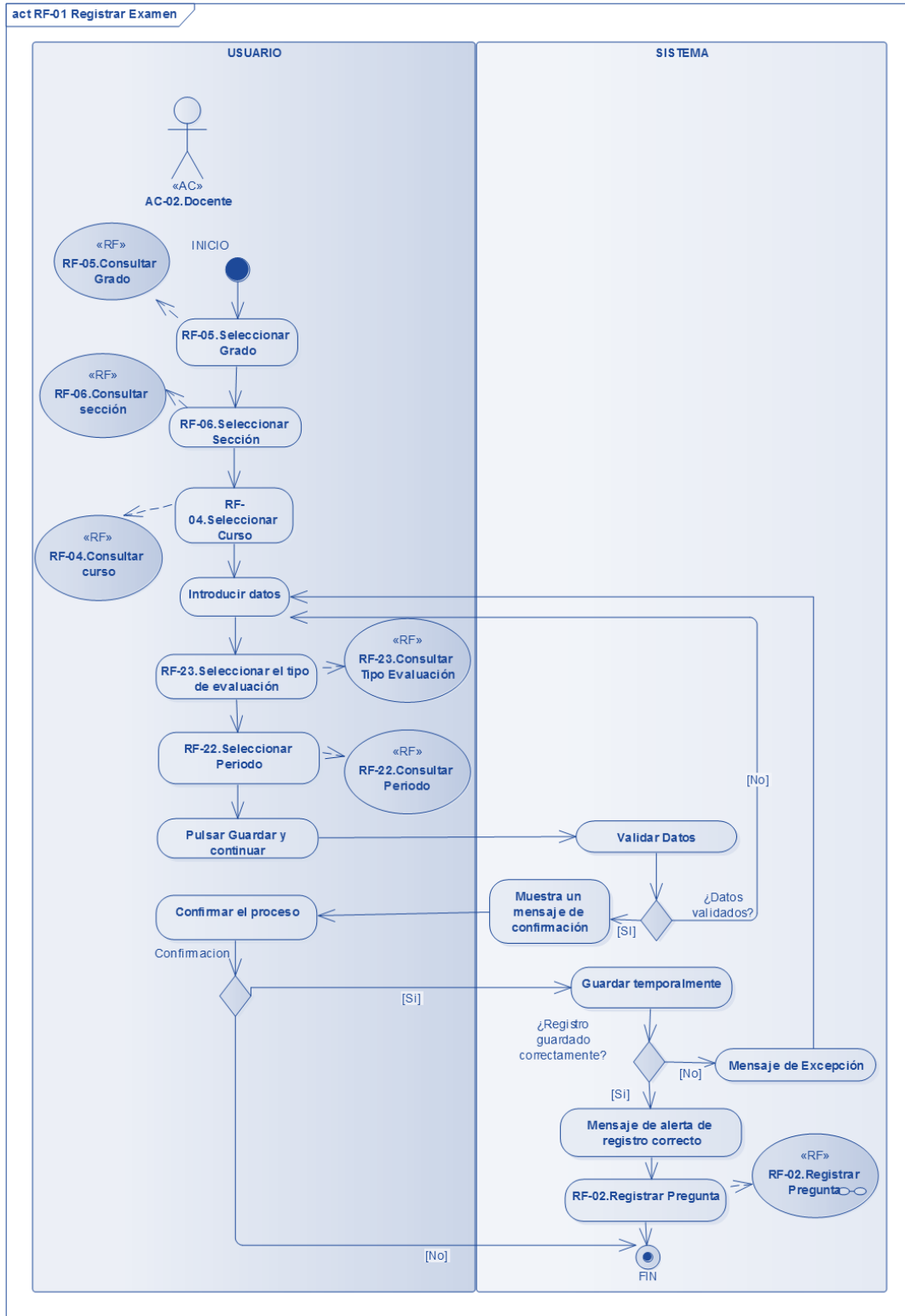


Diagrama 10 Requisitos funcionales: Realizar Examen, Diagrama de Actividades, NDT.



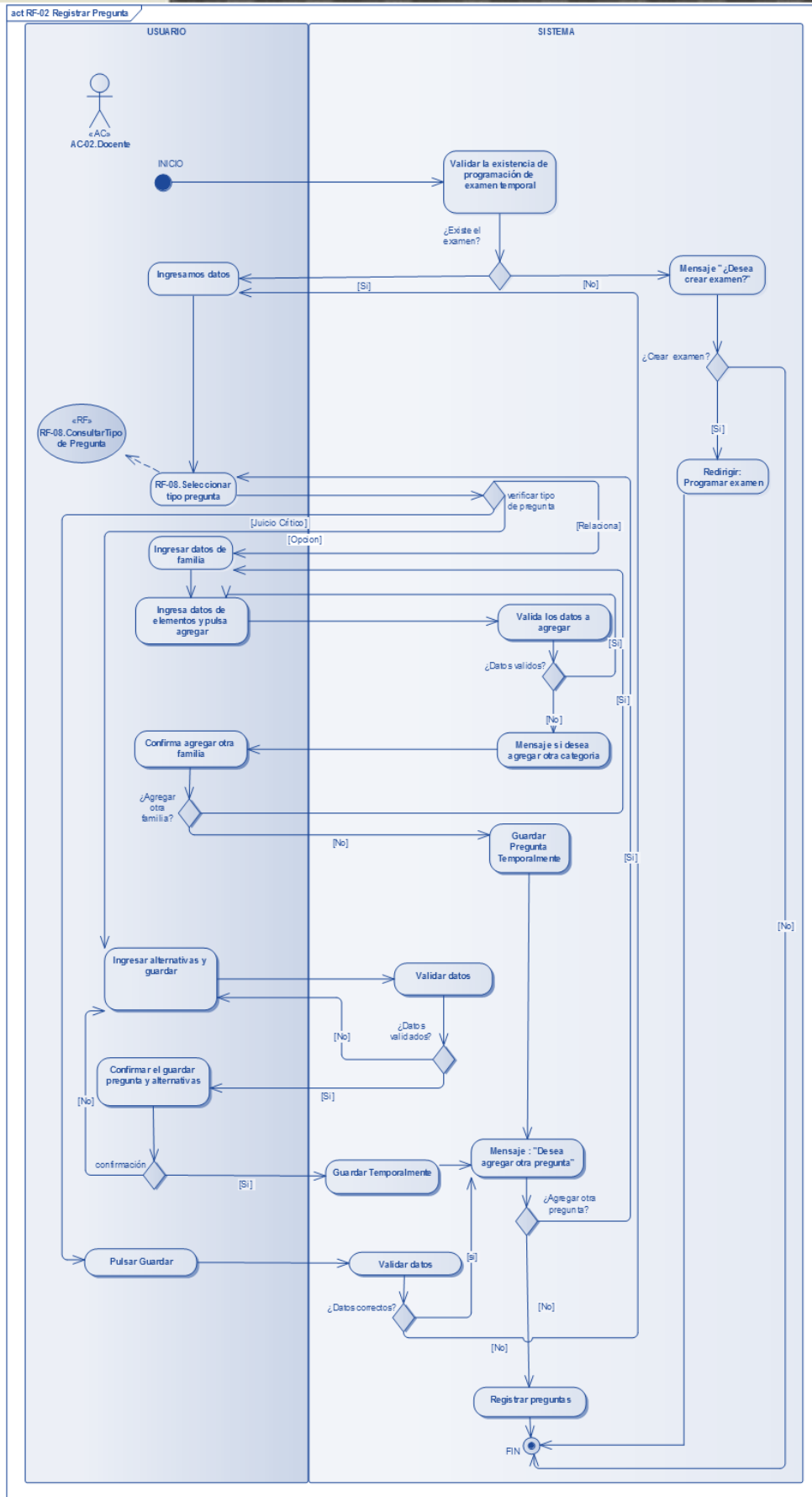


Diagrama 11 Requisitos funcionales: Realizar Examen, Diagrama de Actividades, NDT.



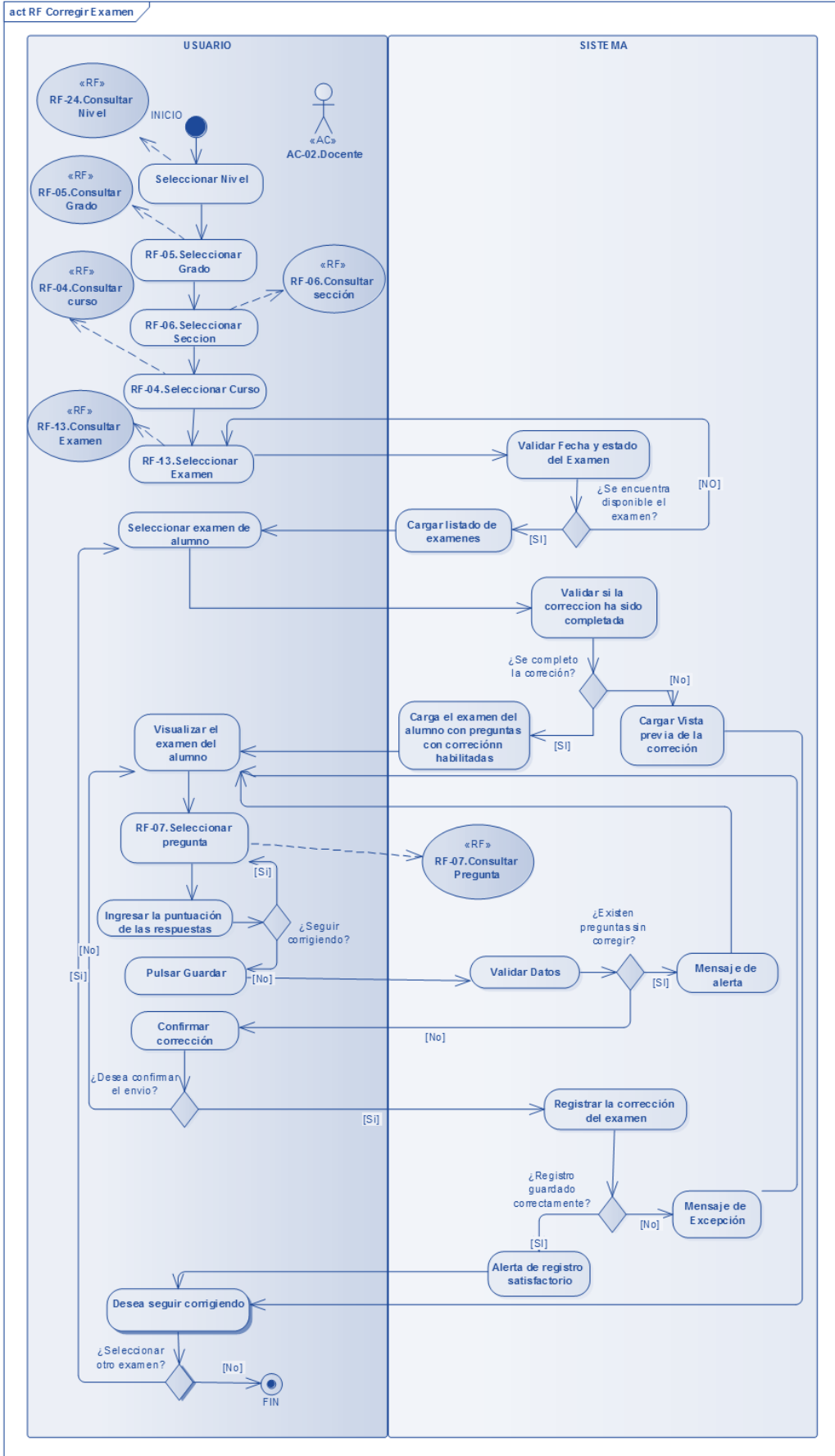


Diagrama 12 Requisitos funcionales: Realizar Examen, Diagrama de Actividades, NDT



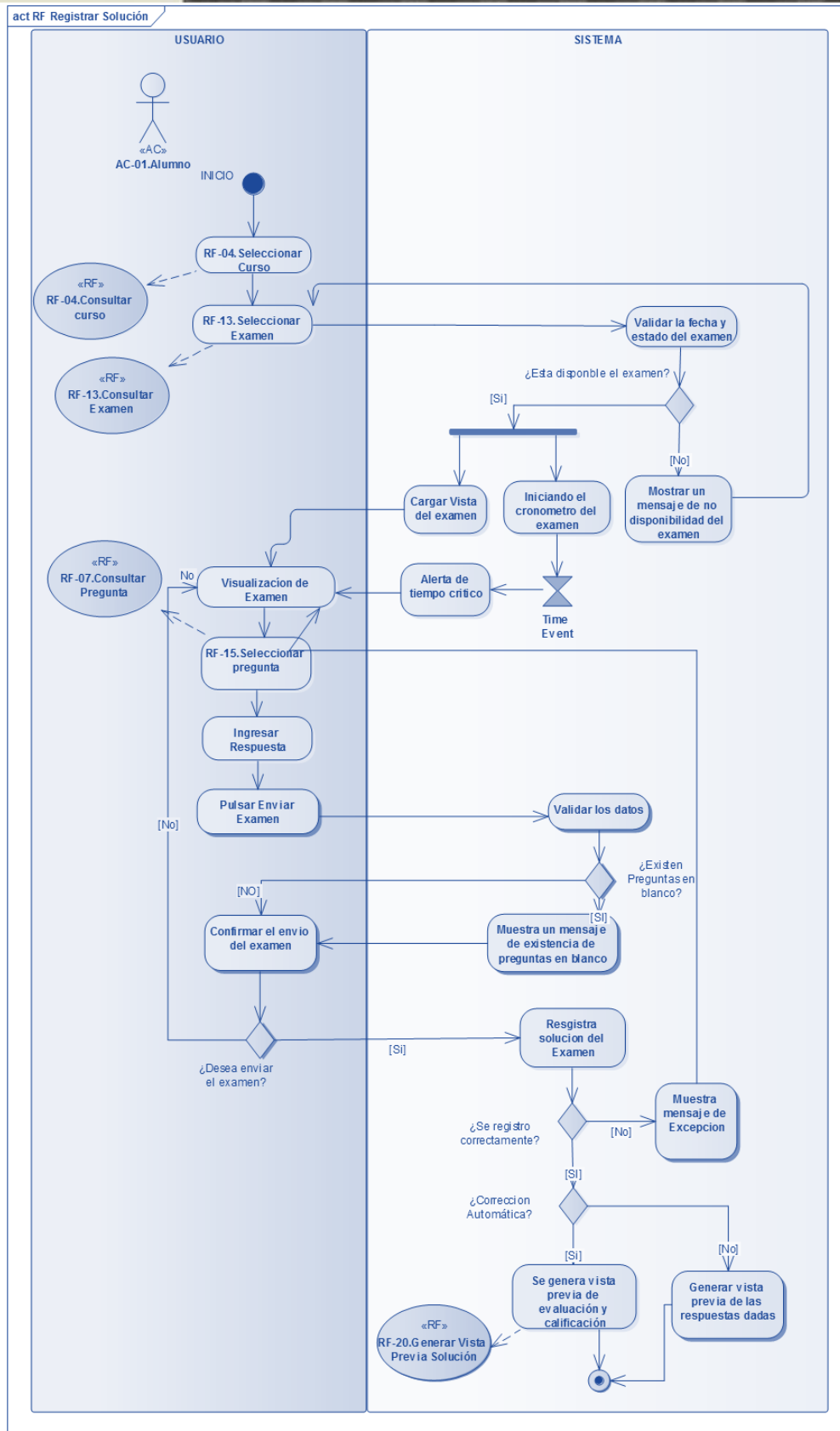


Diagrama 13Requisitos funcionales: Realizar Examen, Diagrama de Actividades NDT.



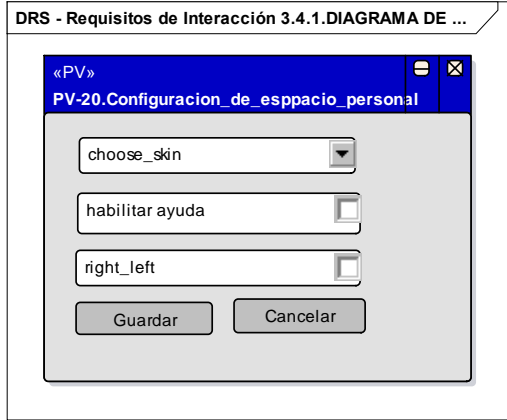
### 6. DIAGRAMA DE REQUISITOS DE INTERACCIÓN

(IWT2, 2016, pág. 71) Los diagramas de interacción sirve para recoger la forma en que los diferentes actores pueden interactuar con el sistema durante la navegación y así poder recuperar la información que el usuario desea, la cual se representan a través de dos aspectos: Uno es el criterio de la recuperación o frase y los prototipos de visualización. En esta etapa se incluyó a los siguientes mecanismos de usabilidad como son el User Profile y Help.

MECANISMO DE USABILIDAD	REQUERIMIENTO DE ALMACENAMIENTO DE INFORMACIÓN	DESCRIPCIÓN
Help	<p>DRS - Requisitos de Interacción 3.4.1.DIAGRAMA DE INTERACCIÓN</p> <pre>                     usecaseDiagram                         actor Actor                         usecase FR9 as «FR» FR-9.Recuperar_Ayuda                         usecase LI14 as «L» LI-14.Listado de ayuda                         usecase PV19 as «PV» PV-19.Ayuda                         Actor --&gt; FR9                         FR9 --&gt; LI14 : «Interactúa con»                         LI14 --&gt; PV19 : «Interactúa con»                         Actor --&gt; PV19 : «Interactúa con»                     </pre> <p>The diagram illustrates the interaction requirements for the Help mechanism. It consists of three use cases: FR-9 (FR) titled 'Recuperar_Ayuda', LI-14 (L) titled 'Listado de ayuda', and PV-19 (PV) titled 'Ayuda'. FR-9 includes a 'categoria_ayuda' dropdown, 'Buscar', and 'cancelar' buttons. LI-14 includes a 'titulo_ayuda' input field and a 'Ver' button. PV-19 includes 'categoria_ayuda', 'titulo_ayuda', 'imagen_ayuda', 'descripcion_ayuda' input fields, and 'Siguiete_ayuda' and 'Salir' buttons. Arrows labeled '«Interactúa con»' show dependencies from FR-9 to LI-14, from LI-14 to PV-19, and from an actor to PV-19.</p>	<p>Elemento incorporado en el modelo de interacción para plasmar el flujo de interfaces para atender la solicitud de ayuda del usuario.</p>





<p>User Profile</p>		<p>Elemento incorporado en el modelo de interacción para plasmar las interacciones que deben de considerar en el sistema para que el usuario pueda configurar al sistema de tal forma que este se adapte a sus preferencias.</p>
---------------------	---	--

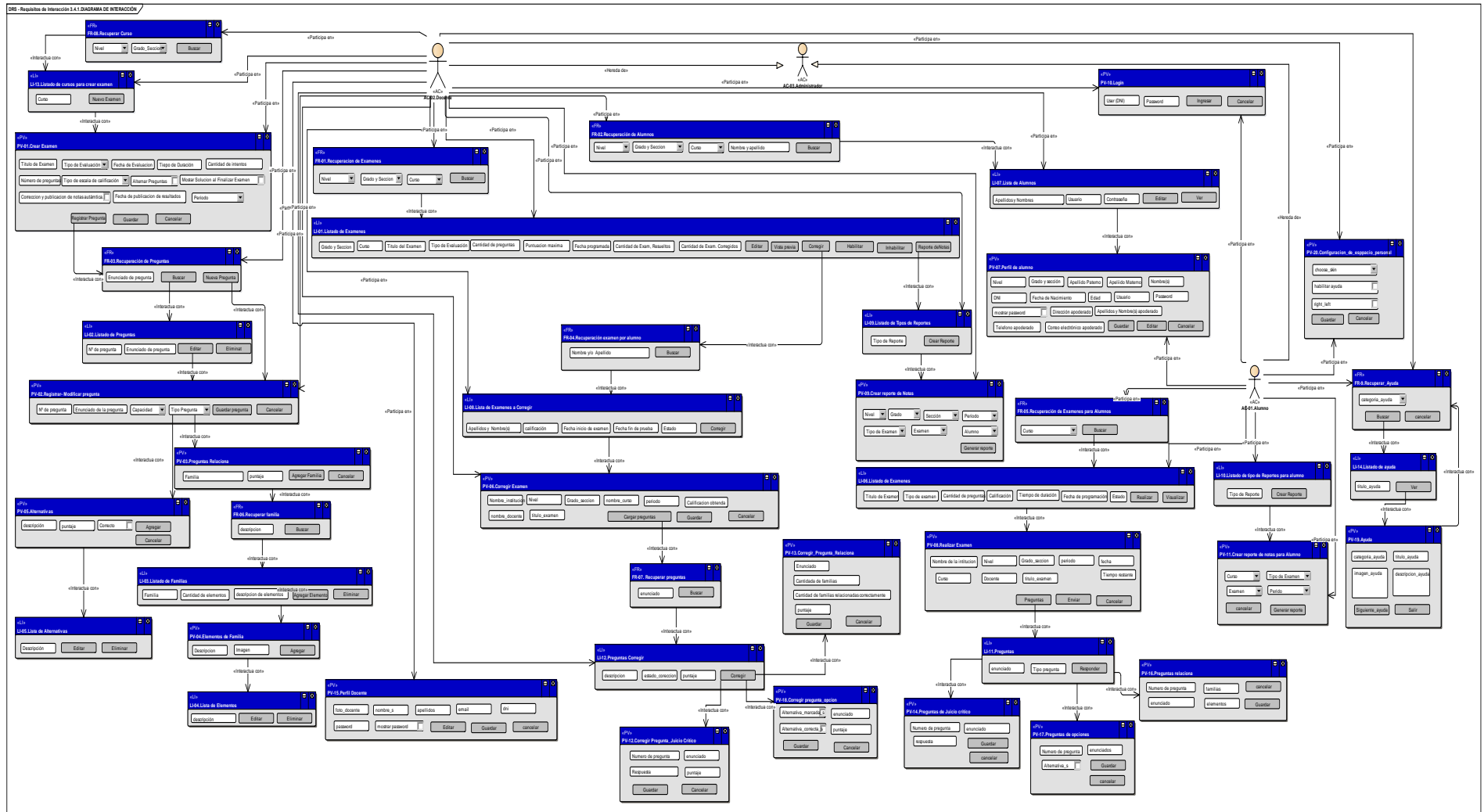


Diagrama 14 Requisitos funcionales: Realizar Examen, NDT.



## 7. DIAGRAMA DE REQUISITOS NO FUNCIONALES

Es este diagrama se plasman los requisitos no funcionales del sistema.

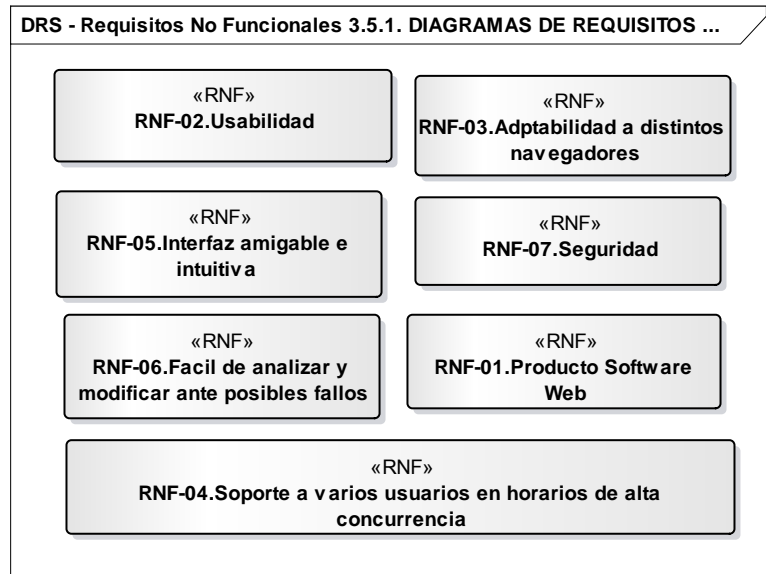


Diagrama 15Requisitos no funcionales, NDT.

## B. DEFINICIÓN DE REQUERIMIENTOS: UWE

La UWE considera esta esta fase esta fase como la fase de ingeniería de requisitos tiene como realizar una descripción general de los diferentes procesos que podría contener el sistema para ello se plasma a través de casos de usos, que son una descripción del comportamiento del sistema al afrontar una tarea o requisito de negocio. A continuación se presenta los artefactos generados en esta etapa:

### 1. DEFINICIÓN DE ACTORES

El diagrama de actores sirve para la identificación de los diferentes actores que harán uso del sistema mediante sus propios artefactos. Los actores identificados para este proyecto son Alumno, Docente y Administrador.

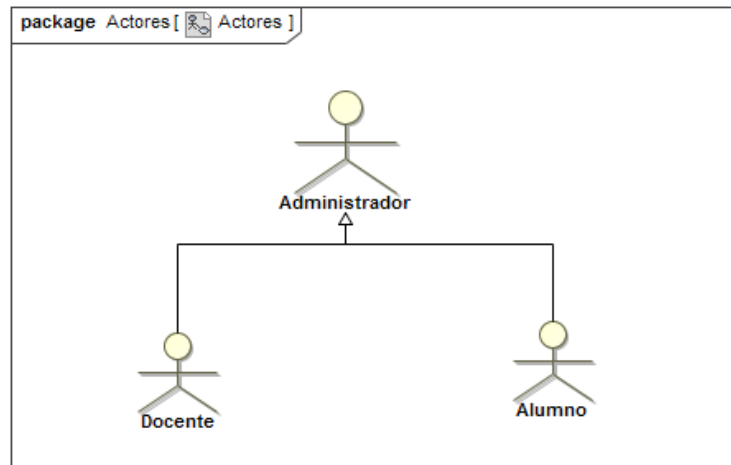


Diagrama 16: Identificación de Actores, UWE.

## 2. DIAGRAMA DE CASO DE USO

Los diagramas de casos de uso se utilizan para describir cuál es la relación entre los actores y el sistema y poder así identificar las actividades que realiza cada actor. A continuación veremos los siguientes estereotipos que se utilizan en los diferentes diagramas de casos de uso desarrollados en la presente investigación:

- a) «navegación» (□): Se utiliza cuando, el usuario realiza cualquier acción que no afecta drásticamente a la aplicación o no realiza ningún cambio a nivel de base de datos.
- b) «proceso» (Σ): Sirve para ilustrar si los datos persistentes de la aplicación son modificados o no.

En el diagrama 17, da al equipo de proyecto una visión general de lo que contendrá el sistema, la cual cuenta con 3 procesos que son la creación de exámenes, la corrección de exámenes y rendir el examen.

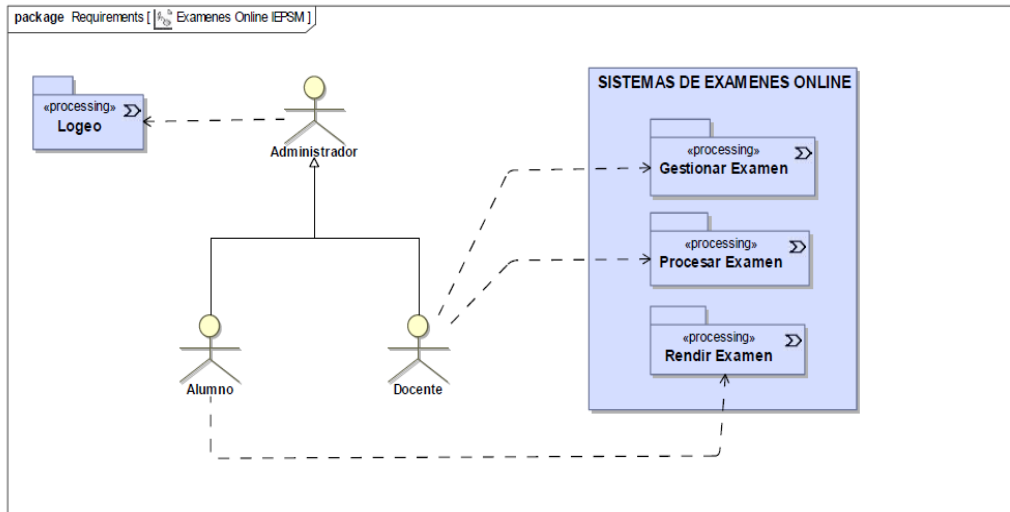


Diagrama 17: Visión general del Sistema de Exámenes Online

A continuación mostraremos los diferentes diagramas que se realizó en los casos de uso:

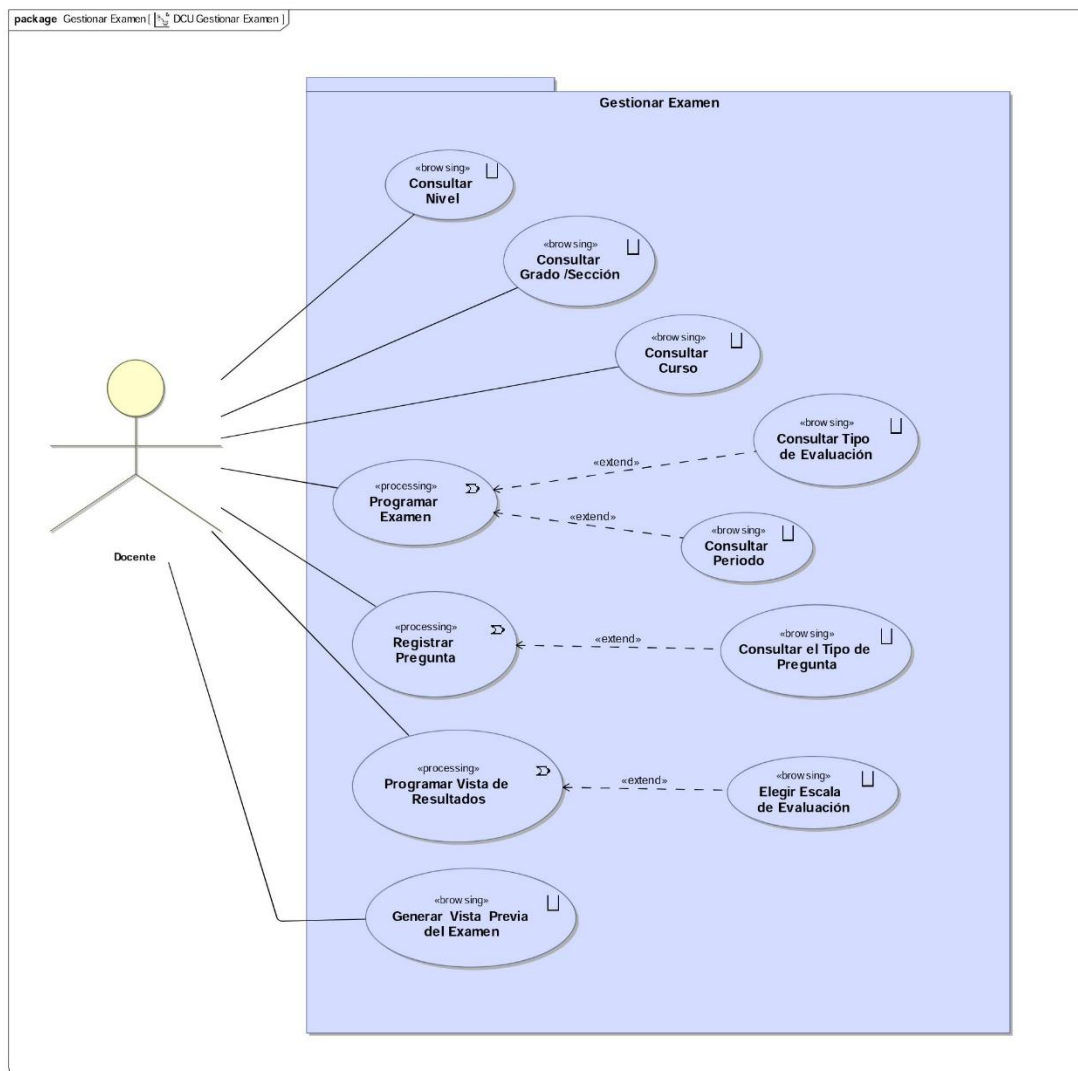


Diagrama 18 Casos de Uso: Gestión de Examen UWE.

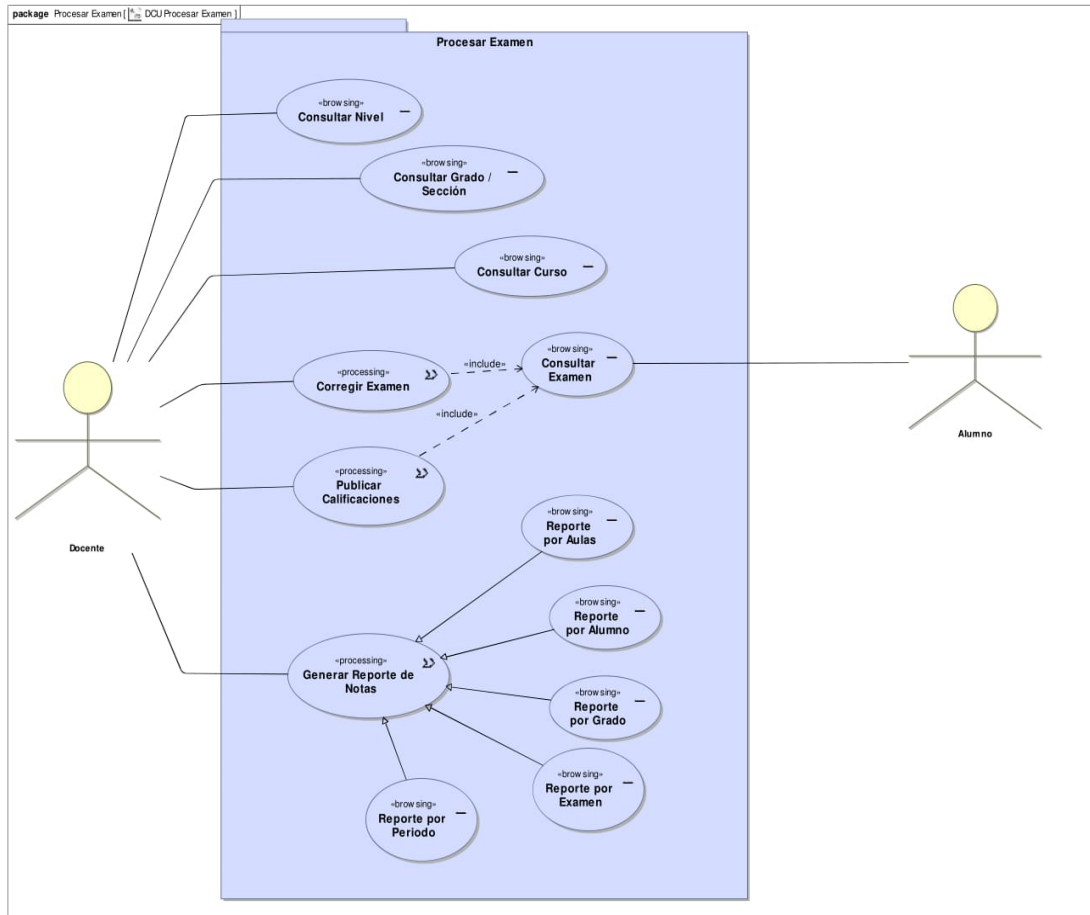


Diagrama 19 Caso de Uso: Procesar Exámenes, UWE

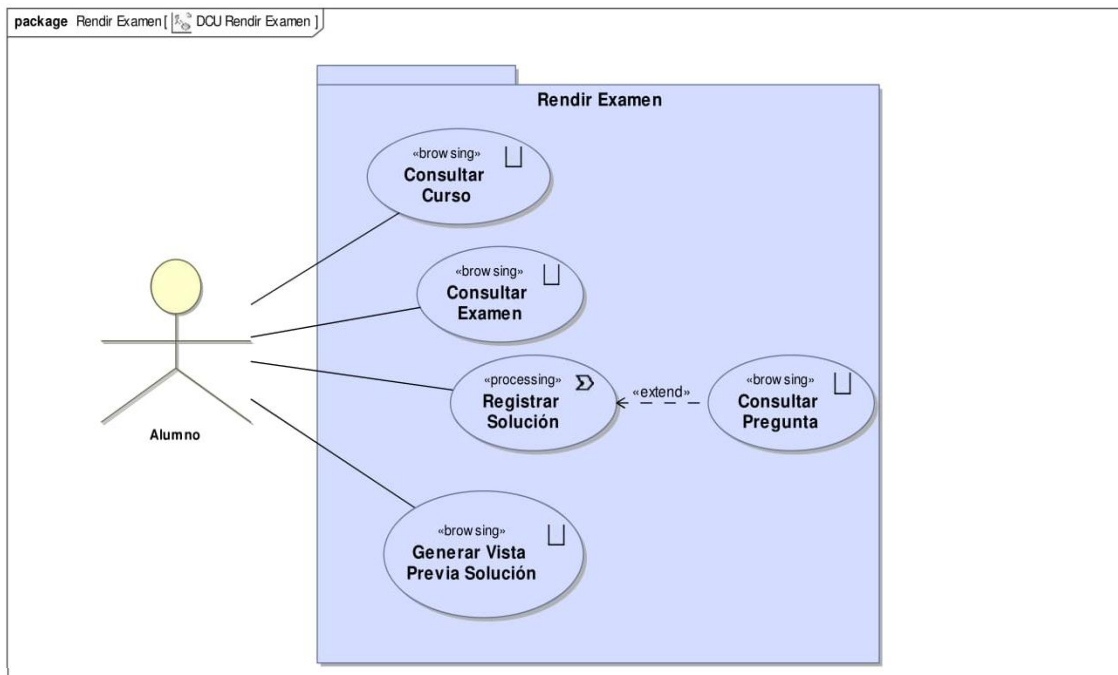


Diagrama 20: Caso de Uso: Rendir Exámenes, UWE.



### 3. DIAGRAMAS DE FLUJO DE PROCESOS (DFP)

Los diagramas de procesos, son una serie de secuencia de pasos, que sirve para especificar el comportamiento del sistema, modelando cada una de las diferentes actividades que se encuentran en cada proceso. Como con casos de uso solamente es posible capturar poca información, cada caso de uso puede ser descrito más detalladamente mediante un proceso.

Para el desarrollo de los diferentes diagramas de proceso que veremos a continuación, veremos el uso de diferentes estereotipos como son:

- a) «user Action» (⤴): Es utilizado para indicar interacciones que el usuario realiza en el sistema.
- b) «system Action» (⤵) describe acciones que son ejecutados por el sistema.

A continuación mostraremos los diferentes diagramas que se realizaron el diagrama del flujo de proceso.

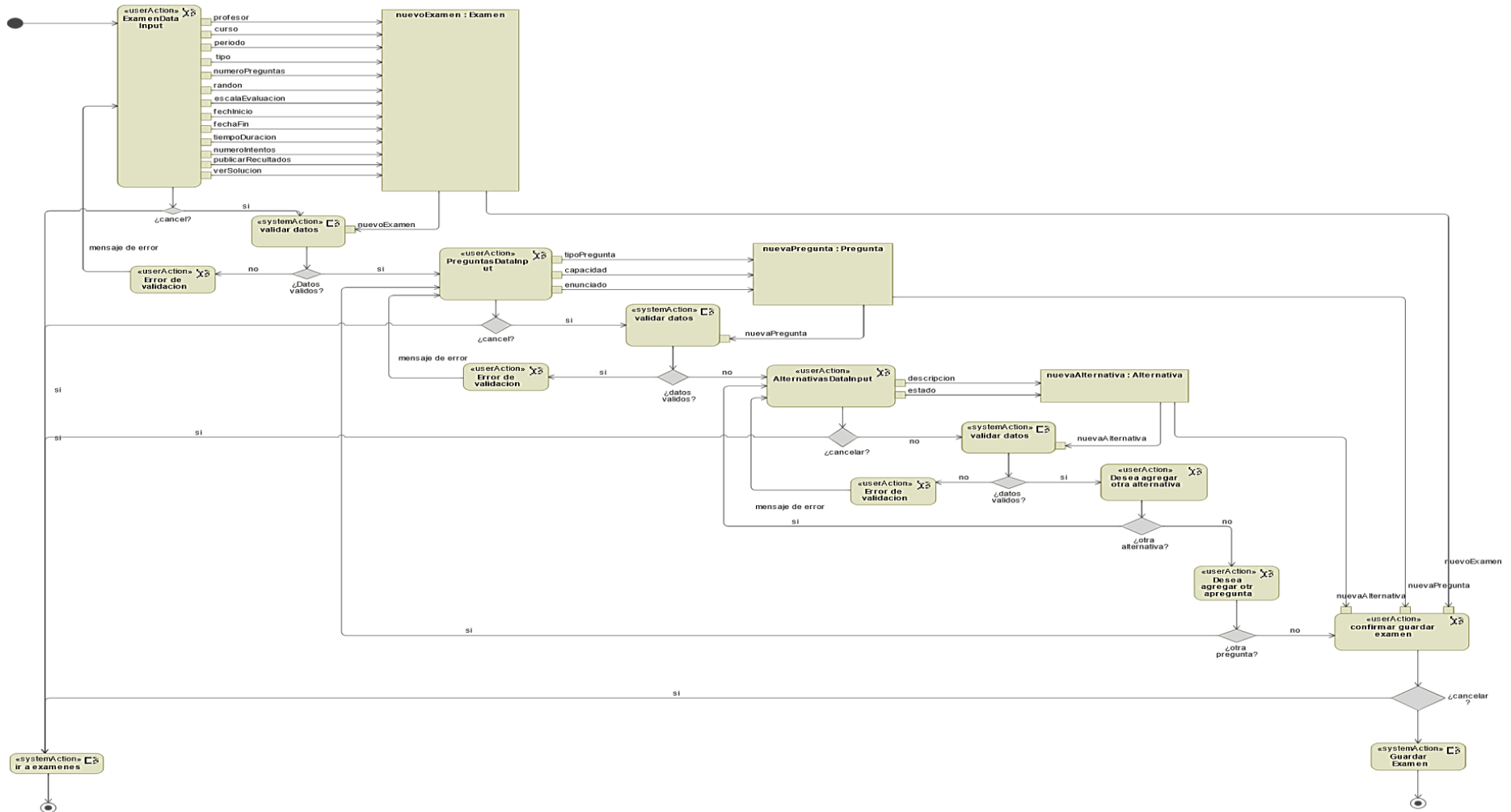


Diagrama 21: Diagrama del Flujo del Proceso: Crear Examen, UWE.





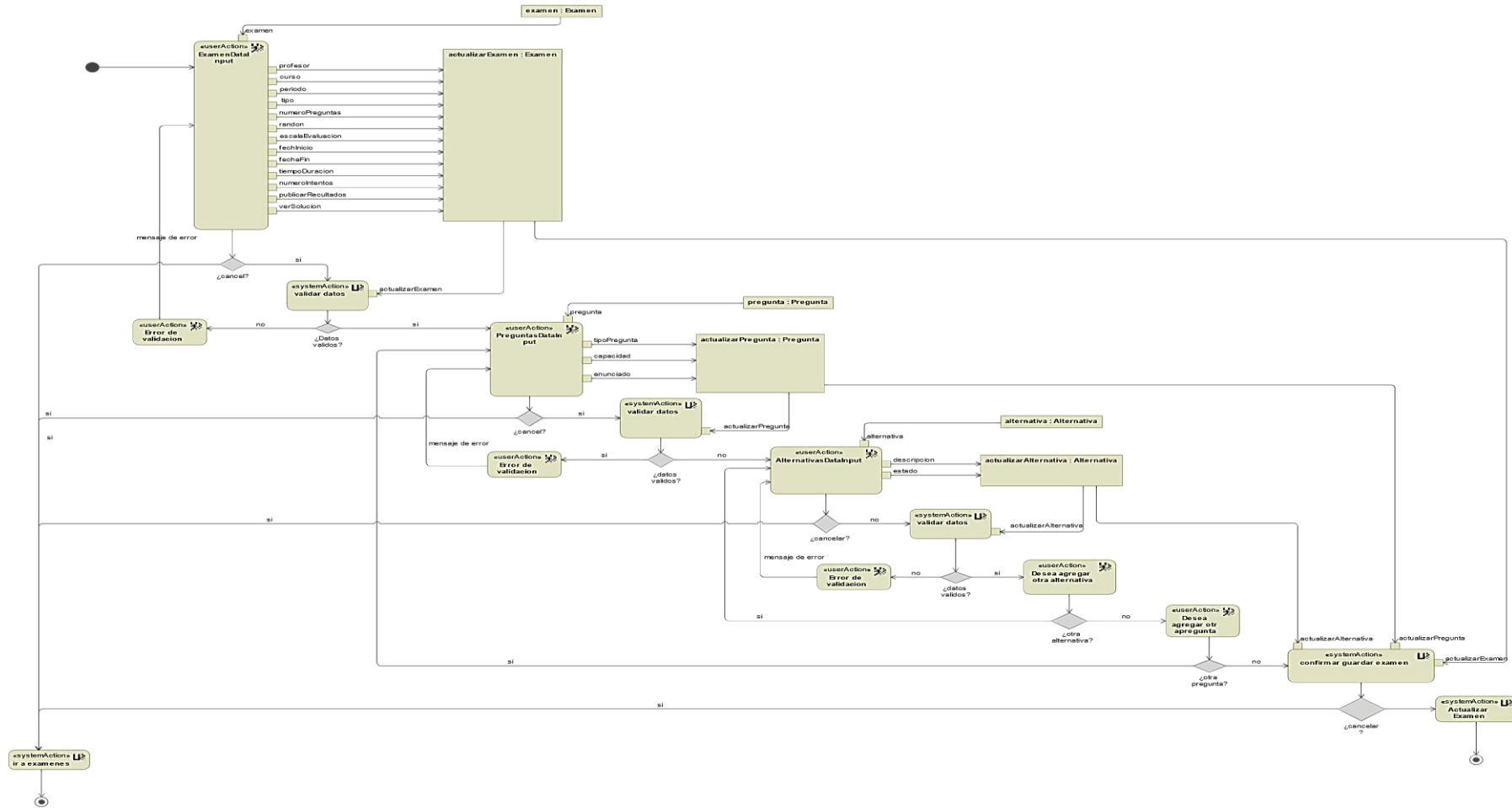


Diagrama 22 Diagrama del Flujo del Proceso: Actualizar Examen, UWE.



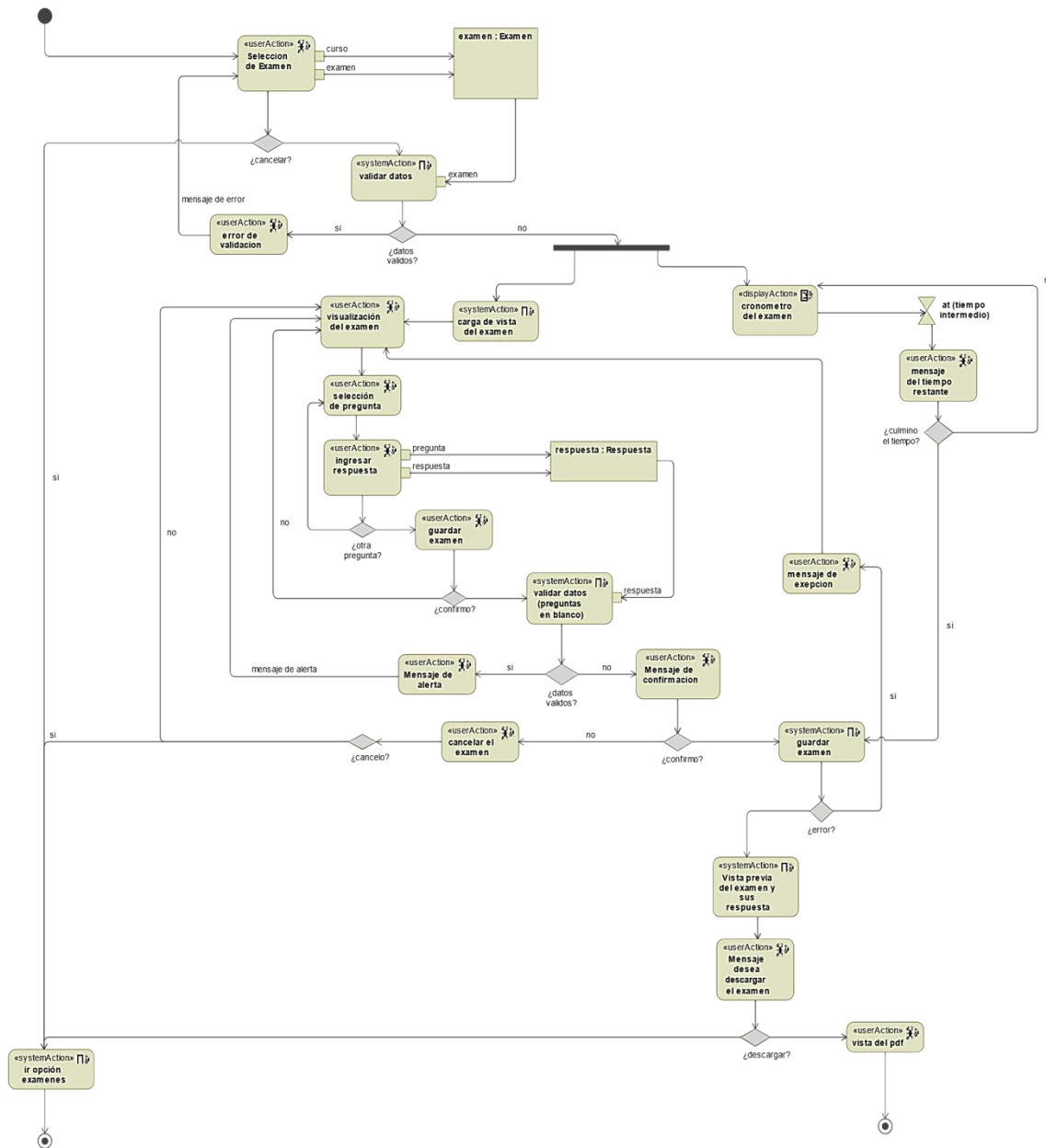


Diagrama 23 Diagrama del Flujo de Proceso: Rendir Examen, UWE.



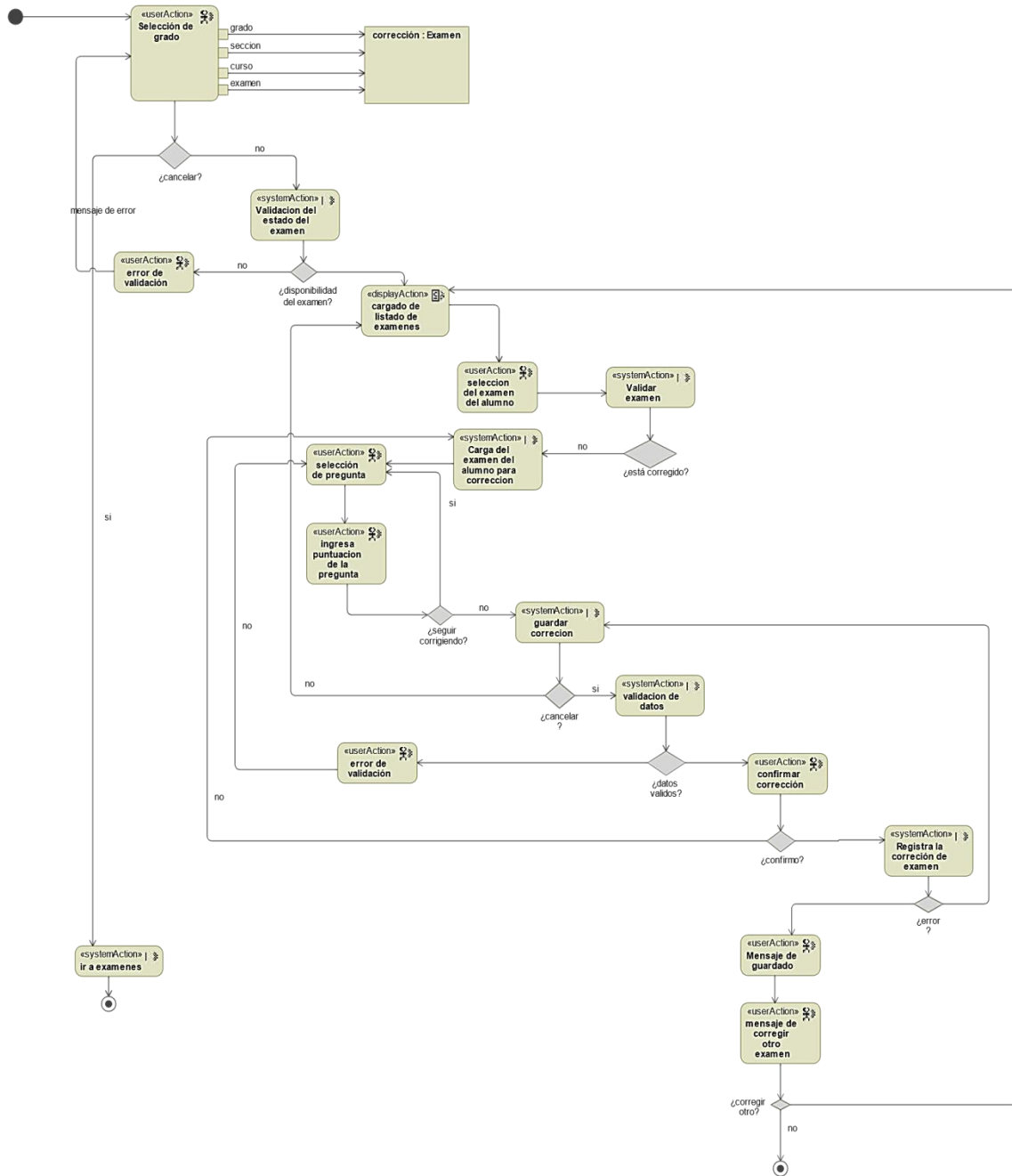


Diagrama 24: Diagrama del flujo del proceso: Corregir Examen, UWE



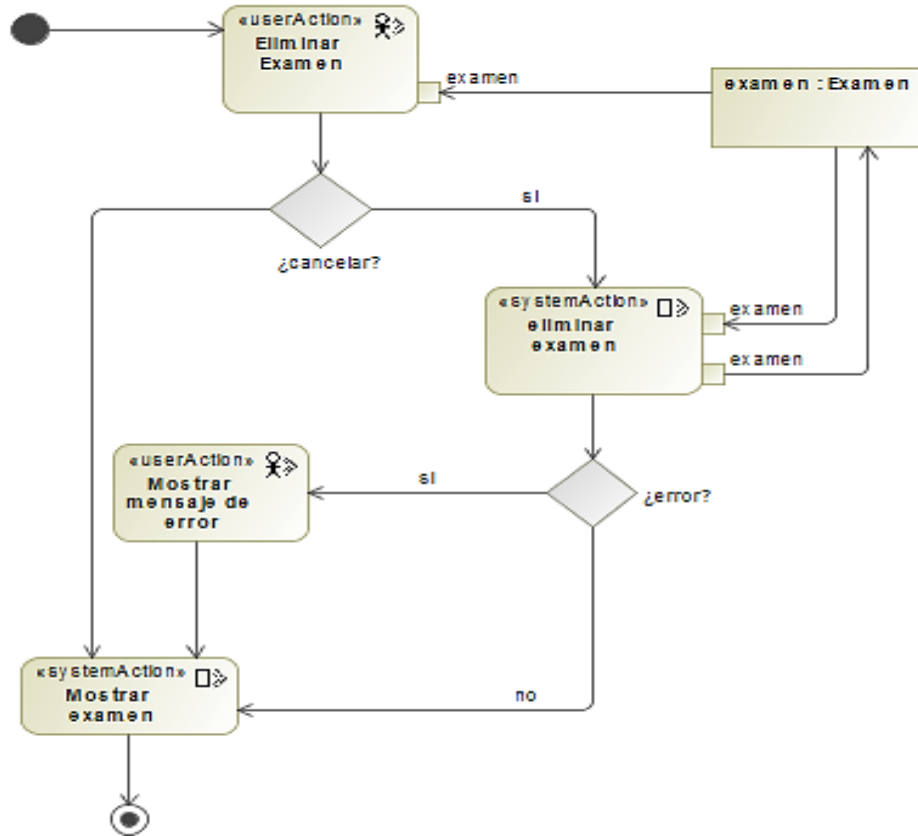


Diagrama 25 Diagrama del Flujo del Proceso Eliminar Examen, UWE.

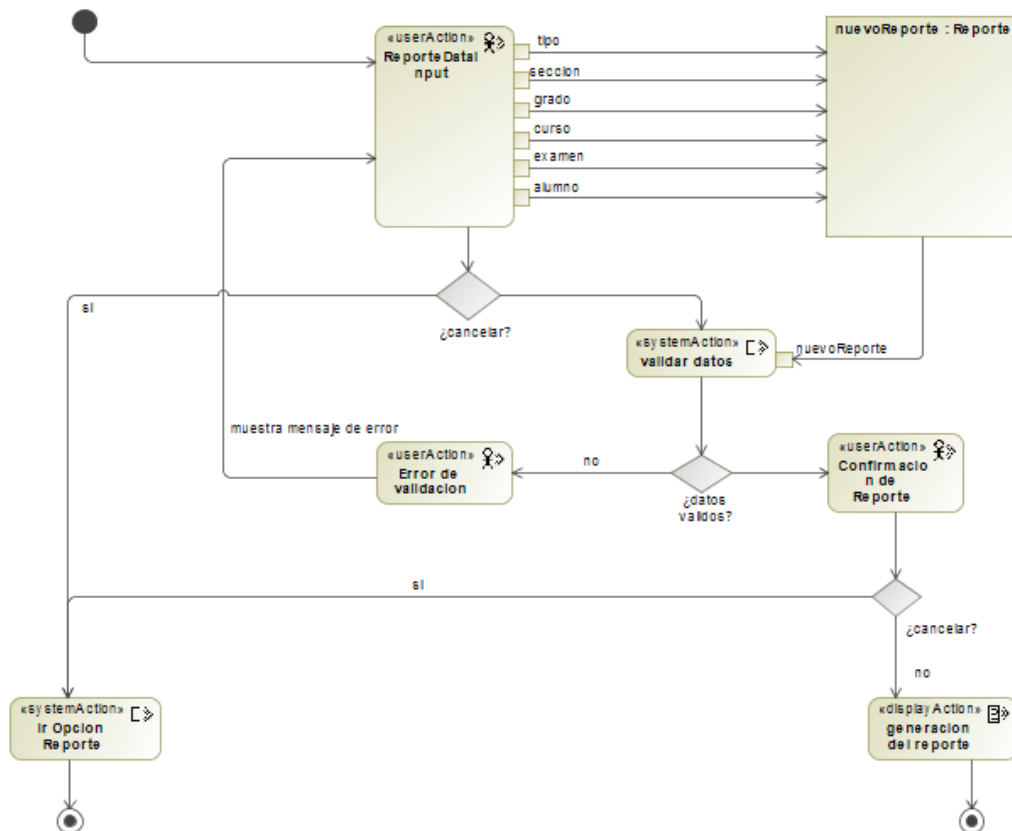


Diagrama 26 Diagrama del Flujo del Proceso Generar Reporte, UWE.



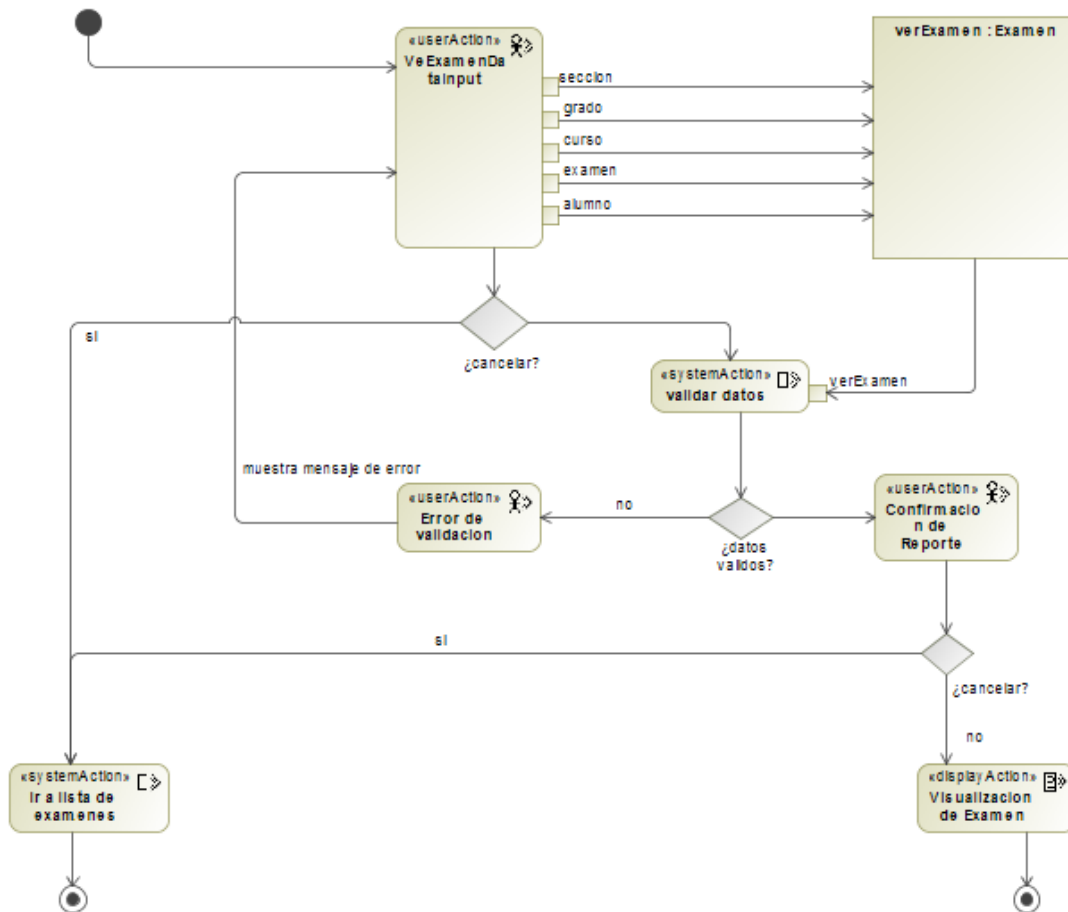


Diagrama 27 Diagrama del Flujo del Proceso Ver examen, UWE.

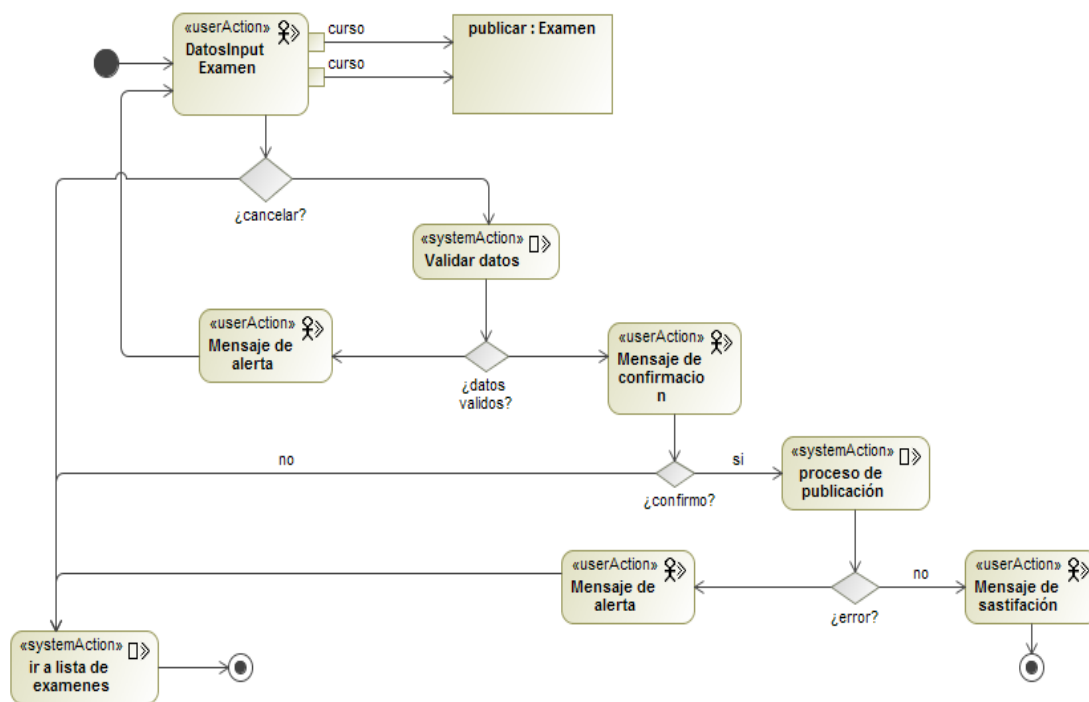


Diagrama 28 Diagrama del Flujo del Proceso Publicar Notas, UWE.



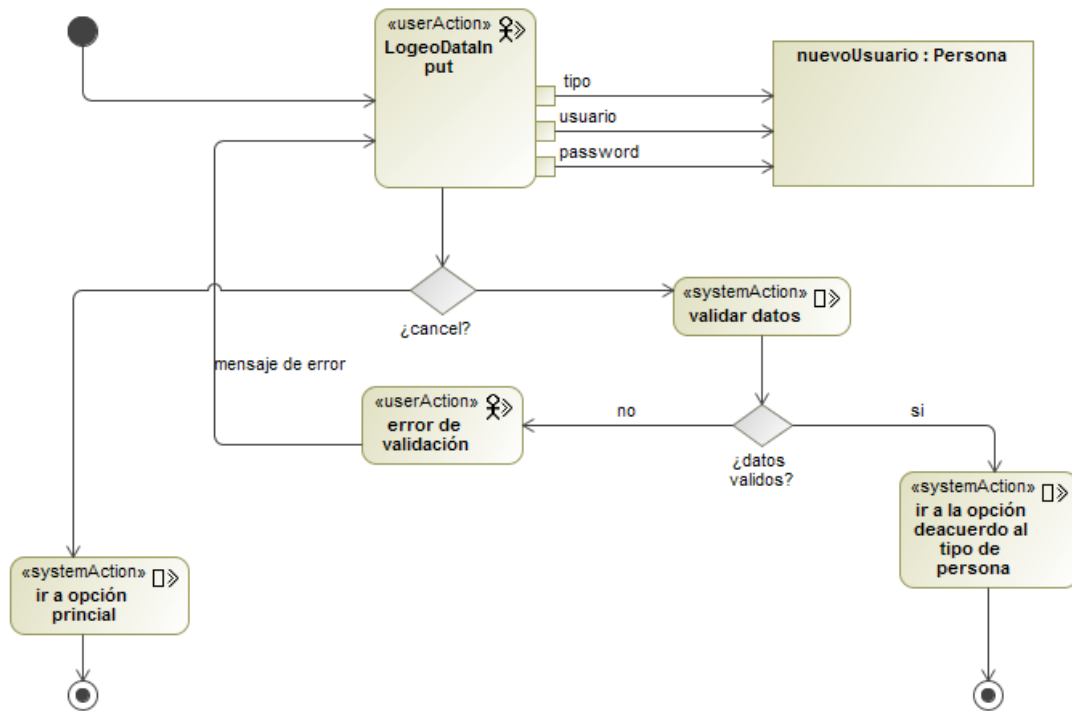
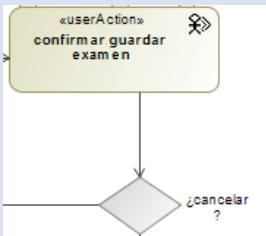


Diagrama 29: Diagrama del Flujo del Proceso Login, UWE

En los diagramas de flujo de proceso se han incluido los requerimientos funcionales de usabilidad generados a partir de la aplicación de UPSEPS, que hacen referencia a los mecanismos de usabilidad:

- a) Warning.
- b) System Status Feedback:
- c) Interaction Feedback:
- d) Abort Operation:

Nombre del diagrama	Componente representativo	Mecanismo de Usabilidad
DFP Crear Examen		Warning

		System Status Feedback
		Interaction Feedback
		Abort Operation
DFP Actualizar Examen		Warning
		System Status Feedback
		Interaction Feedback
		Abort Operation
DFP Rendir		Warning



<p><b>Examen</b></p>		
		<p>System Status Feedback</p>
		<p>Interaction Feedback</p>
		<p>Abort Operation</p>
<p><b>DFP Corregir Examen</b></p>		<p>Warning</p>
		<p>System Status Feedback</p>
		<p>Interaction Feedback</p>

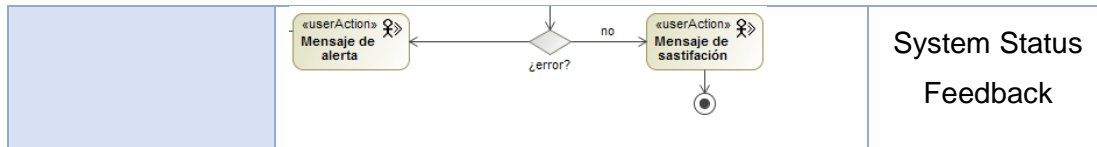




		<p>Abort Operation</p>
		<p>Warning</p>
		<p>System Status Feedback</p>
<p><b>DFP Eliminar Examen</b></p>		<p>Interaction Feedback</p>
		<p>Abort Operation</p>
<p><b>DFP Generar Reporte</b></p>		<p>Warning</p>
		<p>System Status Feedback</p>

	<p>UML Use Case Diagram for 'Reporte Datos Input'. It shows a user action 'Reporte Datos Input' with parameters: tipo, seccion, grado, curso, examen, alumno. A decision diamond asks '¿cancelar?'. If 'no', it leads to a system action 'validar datos'. If 'si', it leads to a user action 'Confirmación de ver examen'.</p>	<p>Abort Operation</p>
<p><b>DFP Ver Examen</b></p>	<p>UML Use Case Diagram for 'Error de validación'. It shows a user action 'Error de validación' triggered by 'muestra mensaje de error'. A decision diamond asks '¿datos validos?'. If 'no', it leads to 'Error de validación'. If 'si', it leads to 'Confirmación de ver examen'.</p>	<p>Warning</p>
	<p>UML Use Case Diagram for 'Mensaje de alerta' and 'Mensaje de satisfación'. A decision diamond asks '¿error?'. If 'no', it leads to 'Mensaje de alerta'. If 'si', it leads to 'Mensaje de satisfación'.</p>	<p>System Status Feedback</p>
	<p>UML Use Case Diagram for 'Confirmación de Reporte'. It shows a user action 'Confirmación de Reporte'. A decision diamond asks '¿cancelar?'. If 'no', it leads to a display action 'Visualización de Examen'.</p>	<p>Abort Operation</p>
<p><b>DFP Publicar Notas</b></p>	<p>UML Use Case Diagram for 'Mensaje de alerta' and 'Mensaje de confirmación'. A decision diamond asks '¿datos validos?'. If 'no', it leads to 'Mensaje de alerta'. If 'si', it leads to 'Mensaje de confirmación'.</p>	<p>Warning</p>
	<p>UML Use Case Diagram for 'Mensaje de alerta' and 'Mensaje de satisfación'. A decision diamond asks '¿error?'. If 'no', it leads to 'Mensaje de alerta'. If 'si', it leads to 'Mensaje de satisfación'.</p>	<p>System Status Feedback</p>
	<p>UML Use Case Diagram for 'Mensaje de confirmación' and 'proceso de publicación'. A decision diamond asks '¿confirmo?'. If 'si', it leads to a system action 'proceso de publicación'. If 'no', it leads to 'Mensaje de confirmación'.</p>	<p>Interaction Feedback</p>
	<p>UML Use Case Diagram for 'Datos Input Examen' and 'Validar datos'. It shows a user action 'Datos Input Examen' with parameters: curso, examen. A decision diamond asks '¿cancelar?'. If 'no', it leads to 'Validar datos'. If 'si', it leads to 'publicar: Examen'.</p>	<p>Abort Operation</p>
<p><b>DFP Logearse</b></p>	<p>UML Use Case Diagram for 'error de validación' and 'ir a la opción de acuerdo al tipo de persona'. A decision diamond asks '¿datos validos?'. If 'no', it leads to 'error de validación'. If 'si', it leads to 'ir a la opción de acuerdo al tipo de persona'.</p>	<p>Warning</p>





#### 5.4.4. ANALISIS DEL SISTEMA

##### A. ANALISIS DEL SISTEMA, NDT

La fase de análisis contiene los productos resultantes de analizar, definir y estructurar los requisitos establecidos en la fase anterior.

##### 1. DIAGRAMA DE CLASES DE CONTENIDO

(IWT2, 2016, pág. 92) “El diagrama de contenido permite modelar cómo se estructura la información que se maneja en el sistema”. Dependiendo del modelo de trabajo que se pretenda seguir en el proyecto, el diagrama de clases incluirá un tipo de clases u otro. Las clases representadas en este modelo se derivan de los requisitos de almacenamiento, por ende las encontramos los requerimientos de almacenamiento de información que se consideraron para responder a los mecanismos de usabilidad User Profile, Help, warning, System Status feedback, convertidos en clases de contenido



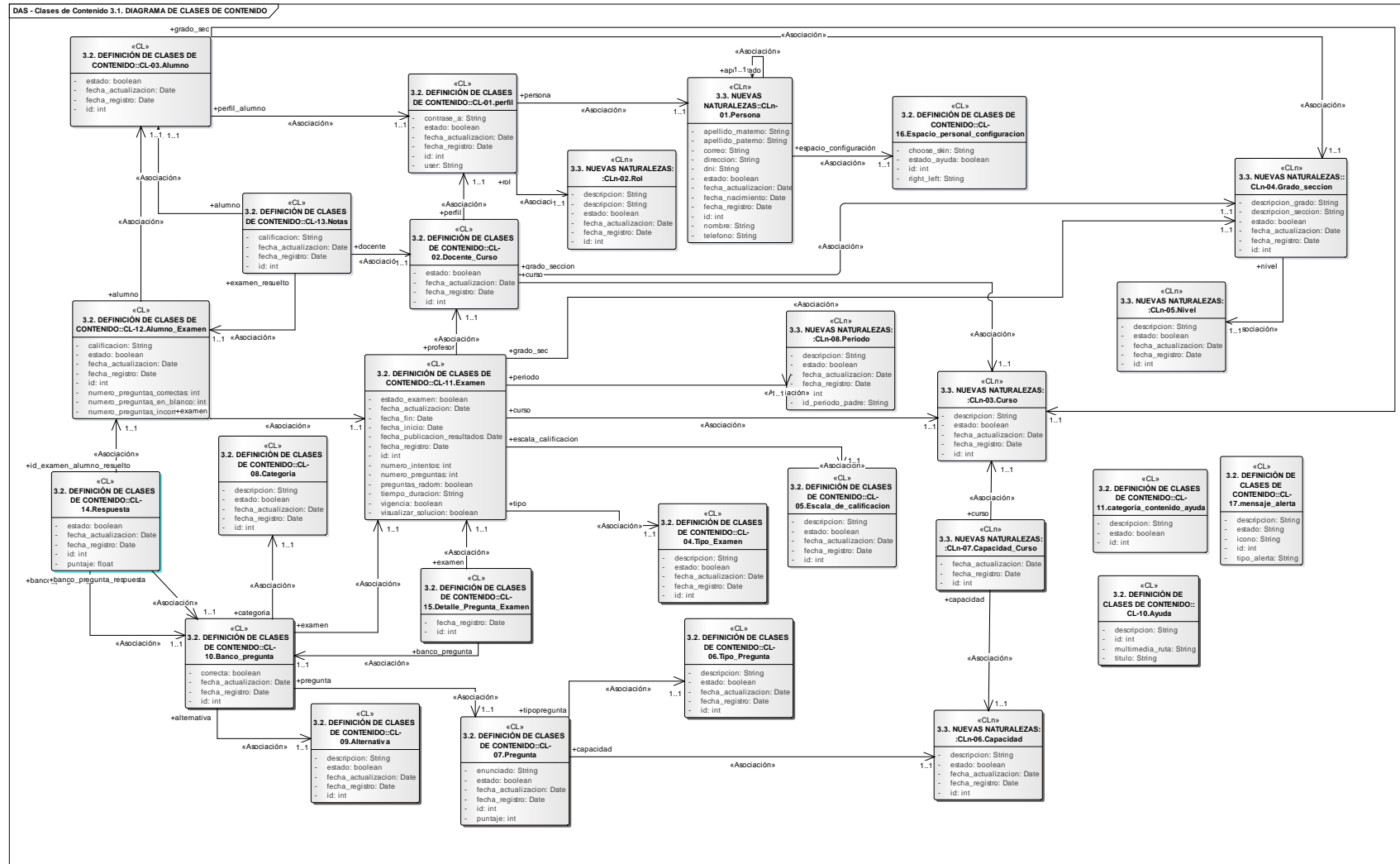


Diagrama 30 Clases de Contenido, NDT.



## 2. MODELO DE CLASES DE PROCESO

(IWT2, 2016, pág. 55) En el modelo de proceso se definen todos los procesos u operaciones que se realiza el sistema, el Diagrama 31 es generado a través de transformaciones, parte de los diferentes diagramas de casos de usos realizados en la fase de definición de los requerimientos funcionales del sistema.

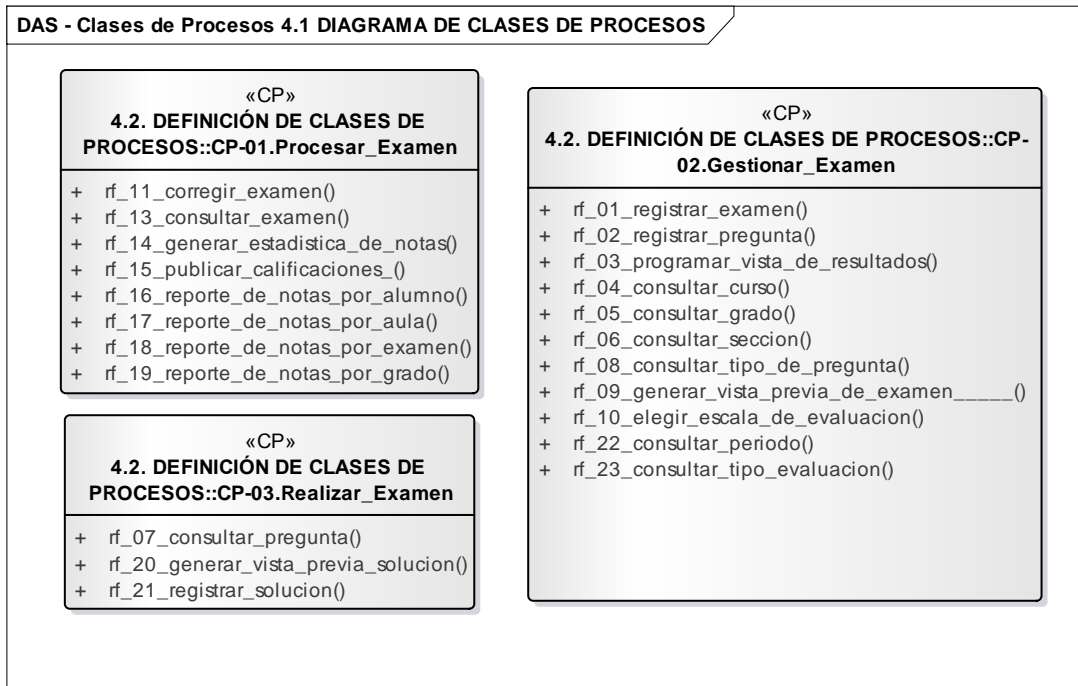


Diagrama 31 Requisitos no funcionales, NDT.

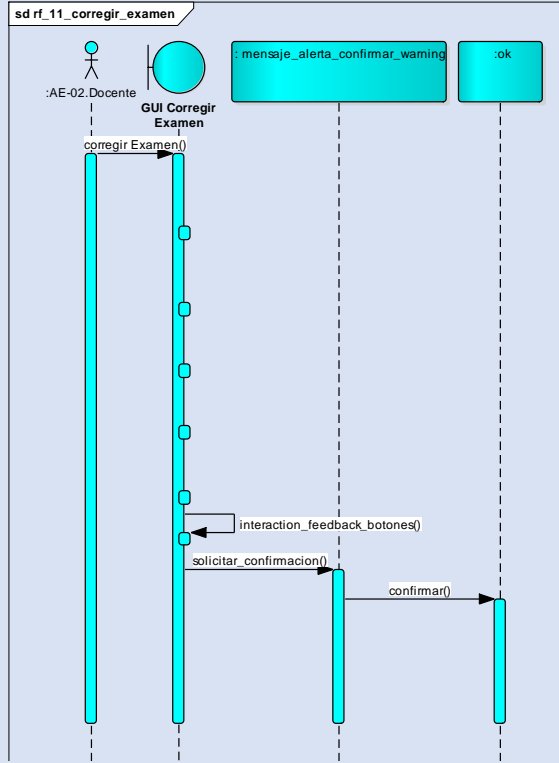
En esta actividad del modelado de clases de proceso también se realizó la representación estructurada de comportamientos, como una serie de pasos secuenciales en el tiempo, por medio de Diagramas de Secuencia, que representan el flujo de trabajo, paso de mensajes y cómo los elementos en general cooperan con el tiempo para lograr un resultado.

En los diagramas de secuencia se incluyó los mecanismo de usabilidad Interaction Feedback, Warnig, Step by step, Structured Text Entry y progress Feedback. En la siguiente tabla presenta los elementos representativos de los mecanismos de usabilidad en los artefactos generados.



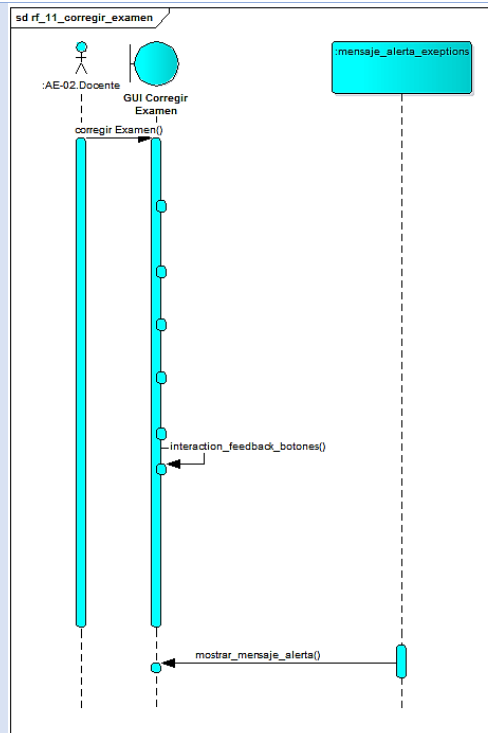
<p><b>MECANISMO DE USABILIDAD</b></p>	<p><b>ELEMENTO REPRESENTATIVO EN EL ARTEFACTO</b></p>
<p style="writing-mode: vertical-rl; transform: rotate(180deg);"><b>Interaction Feedback</b></p>	<div style="display: flex; justify-content: space-around;"> <div data-bbox="598 470 885 1131"> </div> <div data-bbox="885 470 1220 1131"> </div> </div> <p>Elemento incorporado se activa cada vez que hay una interacción del usuario en el sistema, como por ejemplo la pulsación de una tecla, al realizar un clic del ratón, la cual se debe de informar que la interacción del usuario ha sido recibida.</p>

Warnig



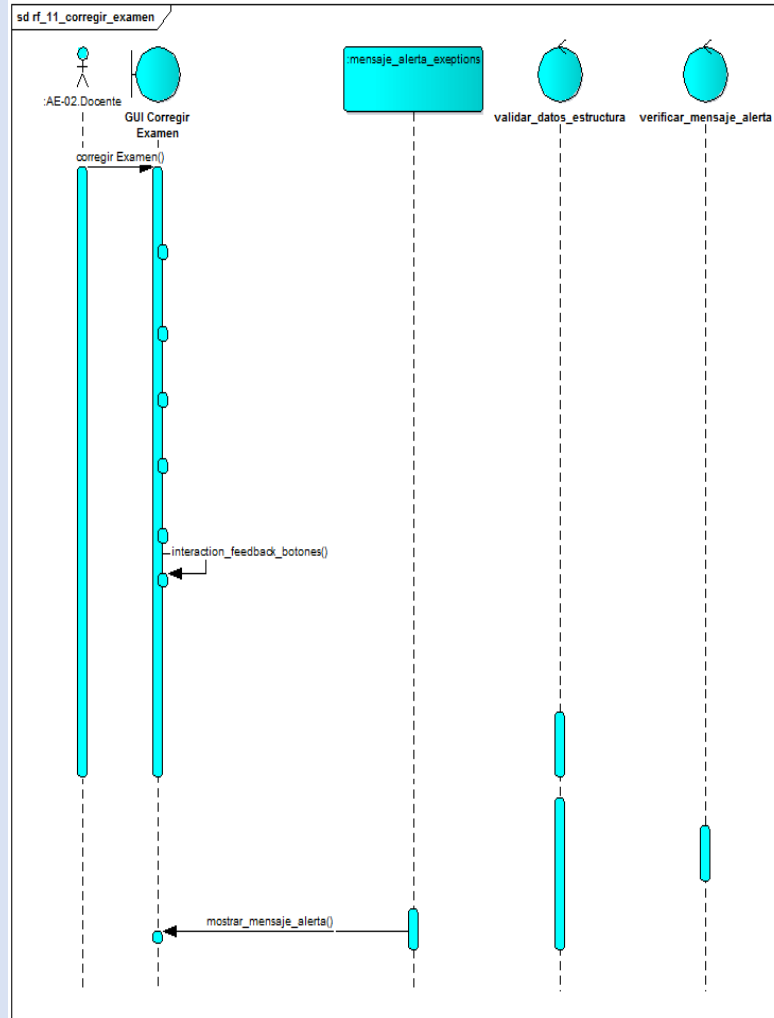
Elemento incorporado para representar las solicitudes de confirmación al usuario para realizar acciones principales.

Warning



Elemento incorporado para representar el envío de mensajes al usuario en caso suceda algún tipo de error mientras se ejecutan procesos en el sistema.

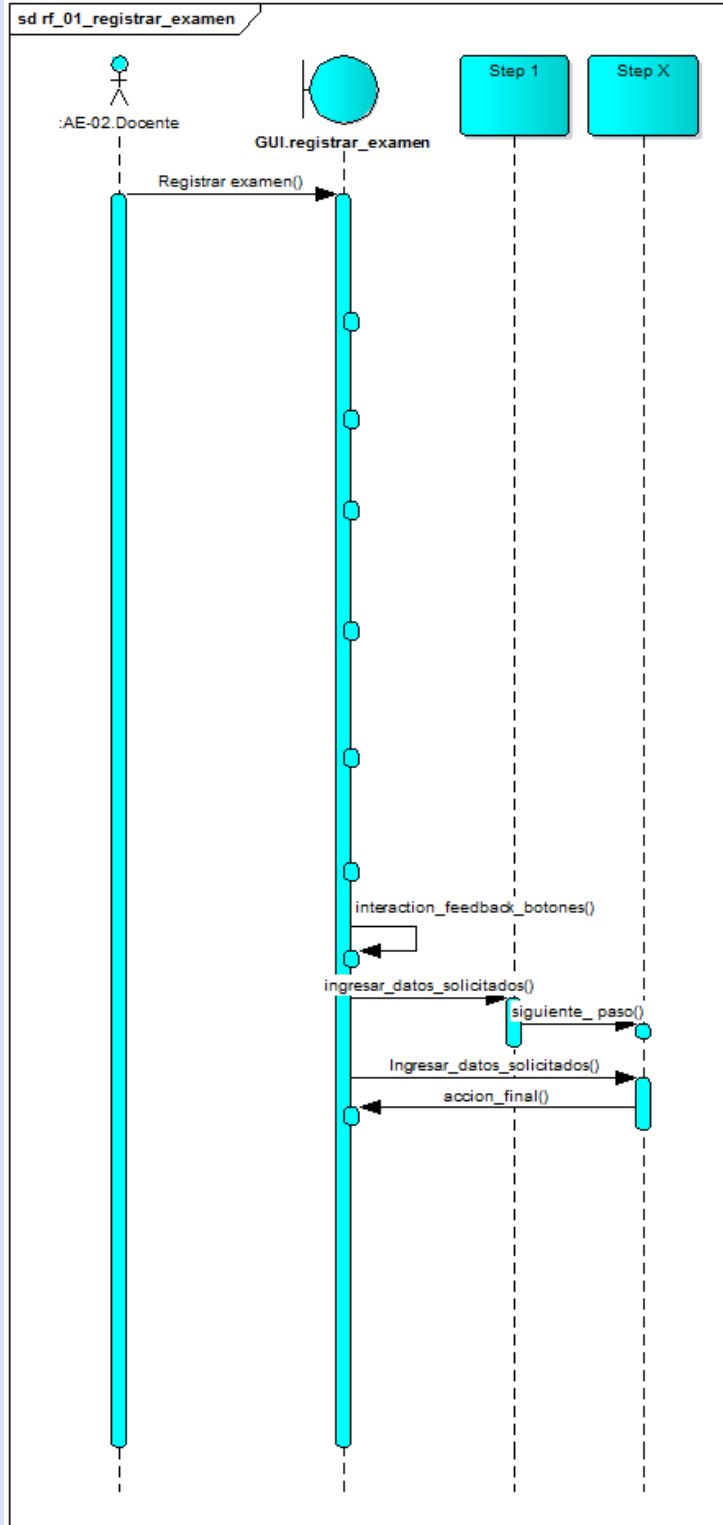
Warning &&Structured Text Entry



Elemento incorporado para representar las validaciones que tiene que hacer el sistema en cuanto a estructura de datos ingresados y los datos que son requeridos obligatoriamente, estas validaciones que realiza el sistema si son de carácter negativo que se hace la búsqueda del mensaje de alerta almacenado en la entidad mensaje alerta, que es enviada a “mensaje\_alerta\_exception”, quien da formato a las presentaciones de las mismas, y seguido de ello son llegadas a los usuarios finales.

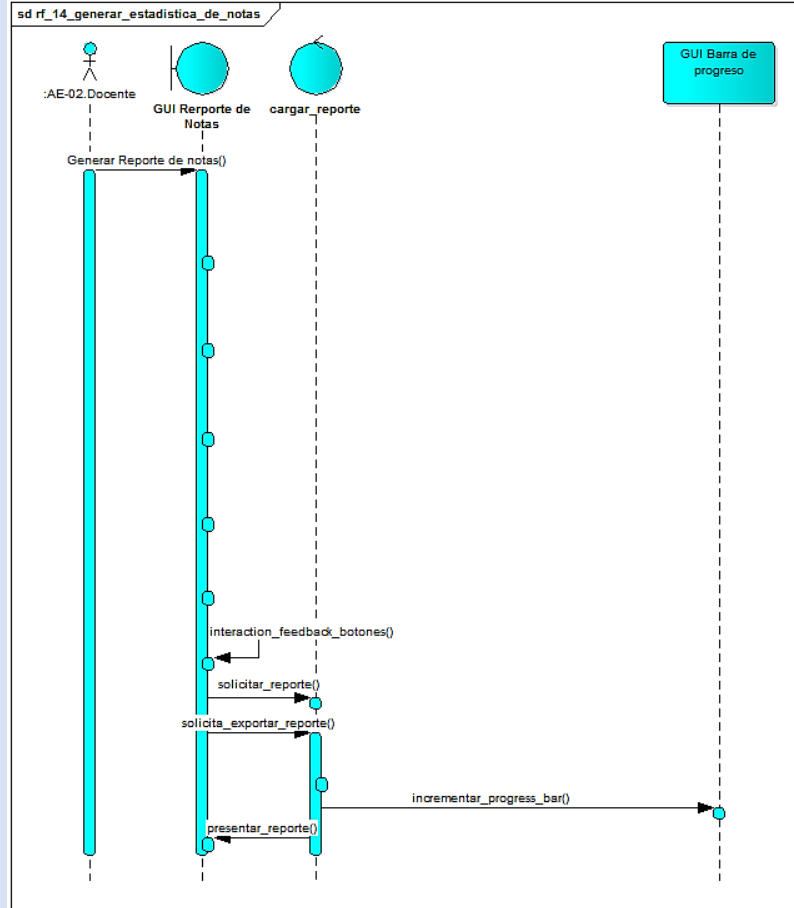


Step by step



Step 1 y Step X son elementos considerados en aquellos procesos que son largos, y es complicado para los usuario con poca experiencia.

Progress Bar



GUI Barra de Procesos, es un elemento con siderado en aquellas acciones que requieren mas de 1 minuto para ser concretada.

A continuación se presenta los elementos incorporados en los diferentes diagramas de secuencias realizado en los principales procesos del sistema.

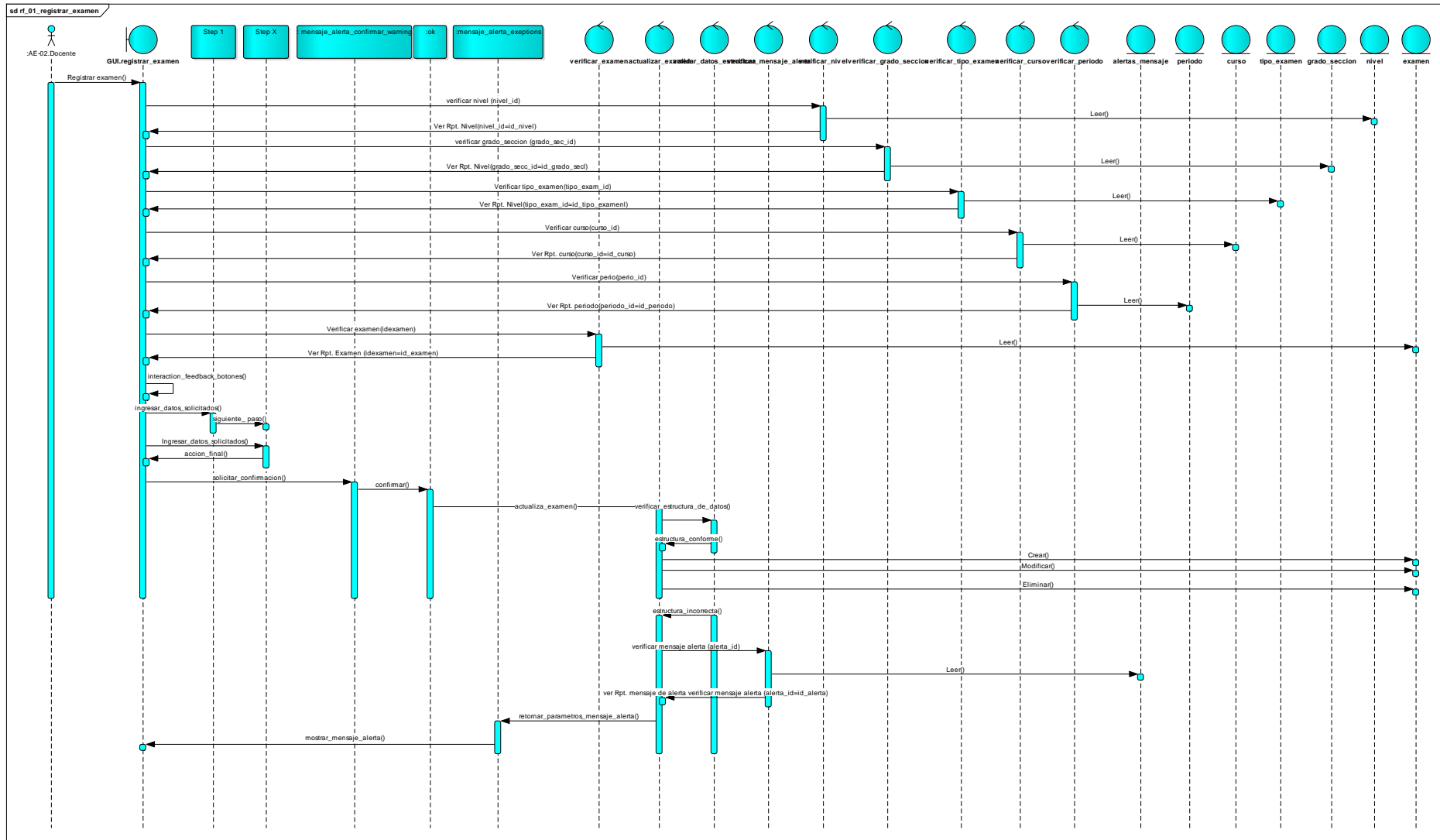


Diagrama 32 Diagrama de Secuencia Registrar Examen NDT.



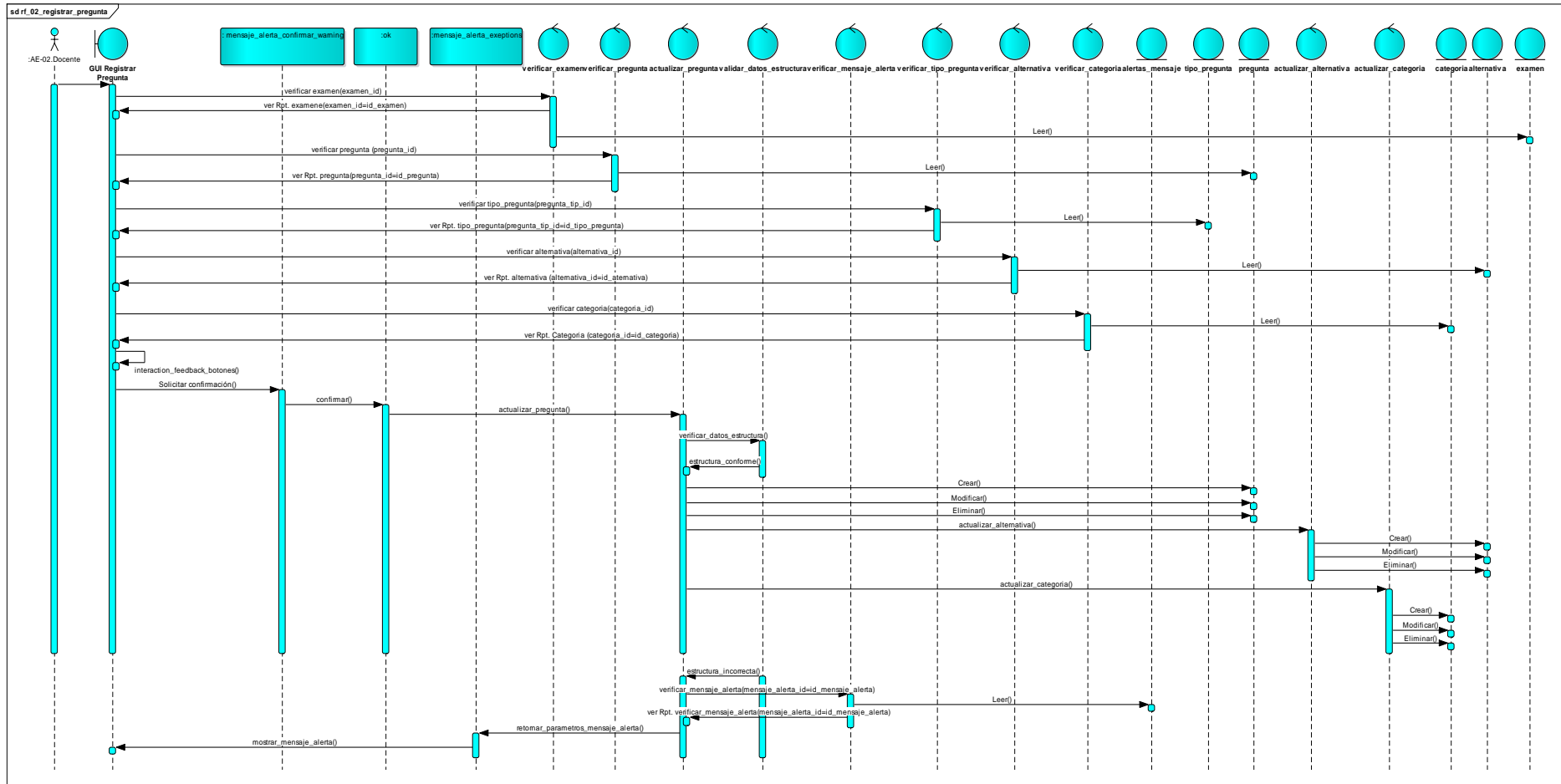


Diagrama 33 Diagrama de Secuencia Registrar Examen NDT.



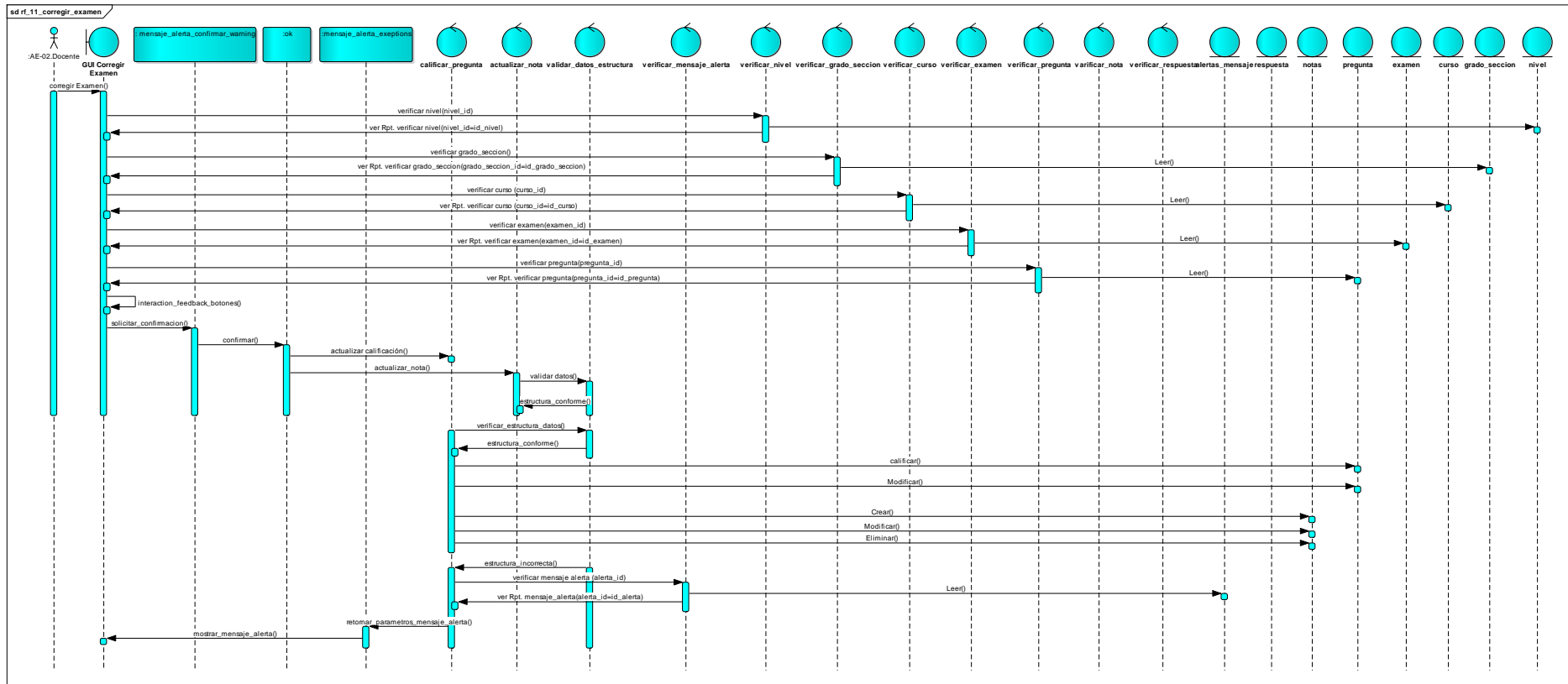


Diagrama 34 Diagrama de Secuencia Registrar Examen NDT.



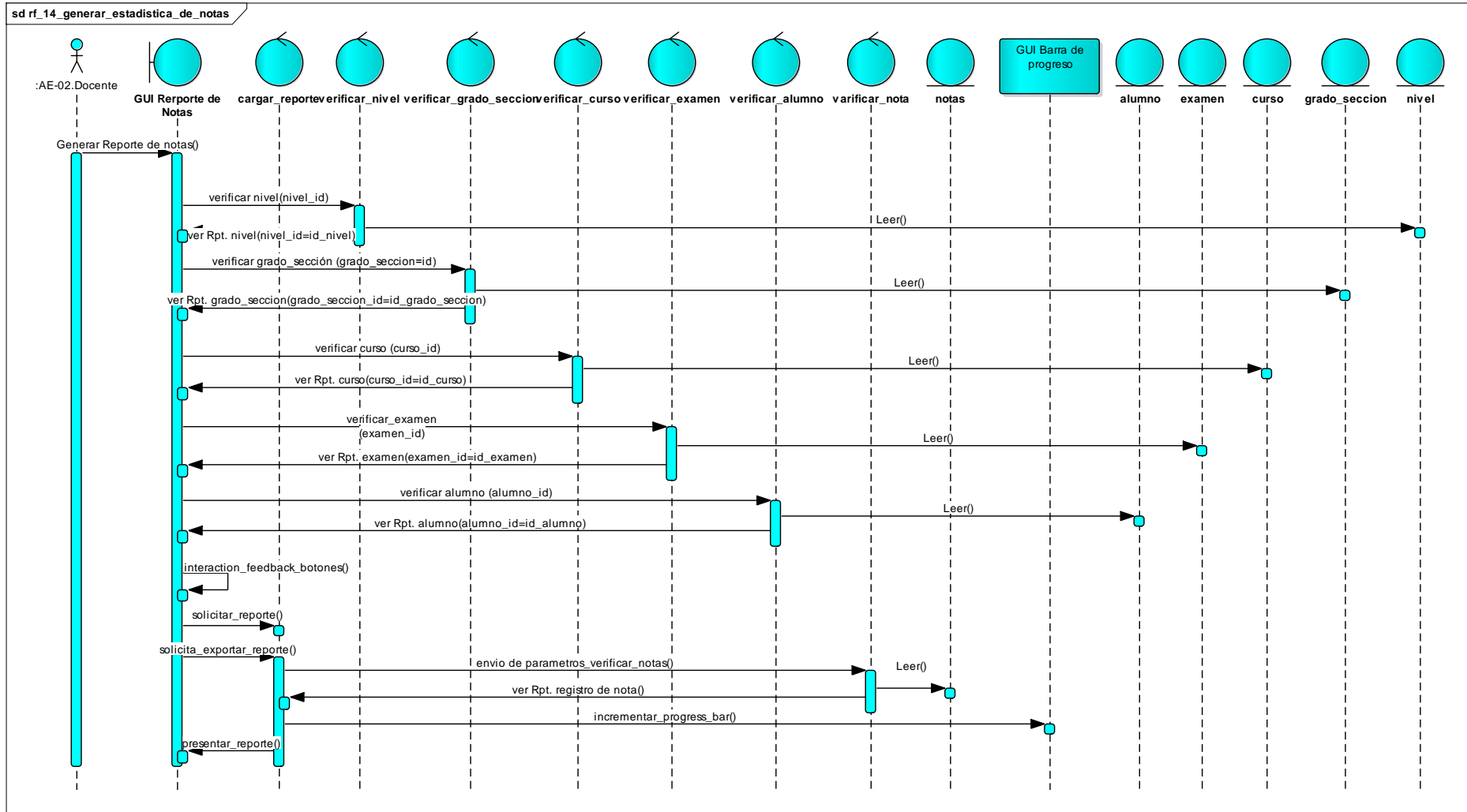


Diagrama 35 Diagrama de Secuencia Registrar Examen NDT.



### 3. MODELO DE NAVEGACIÓN PARA ACTORES DE ESTUDIO

(Grupo de investigación Ingeniería Web y Testing Temprano (IWT2),, 2010) “El modelo de navegación puede variar sustancialmente dependiendo del actor que en cada momento en que interactúe con el sistema. Por ello se definen los actores la etapa de captura y definición de requisitos”. A través de este modelo se puede representar como el actor va poder navegar a través del aplicativo web, y así poder tener una mejor navegabilidad para que el usuario no se pierda durante el manejo del sistema, ofreciendo así la garantía de que el sistema desarrollado es de calidad.

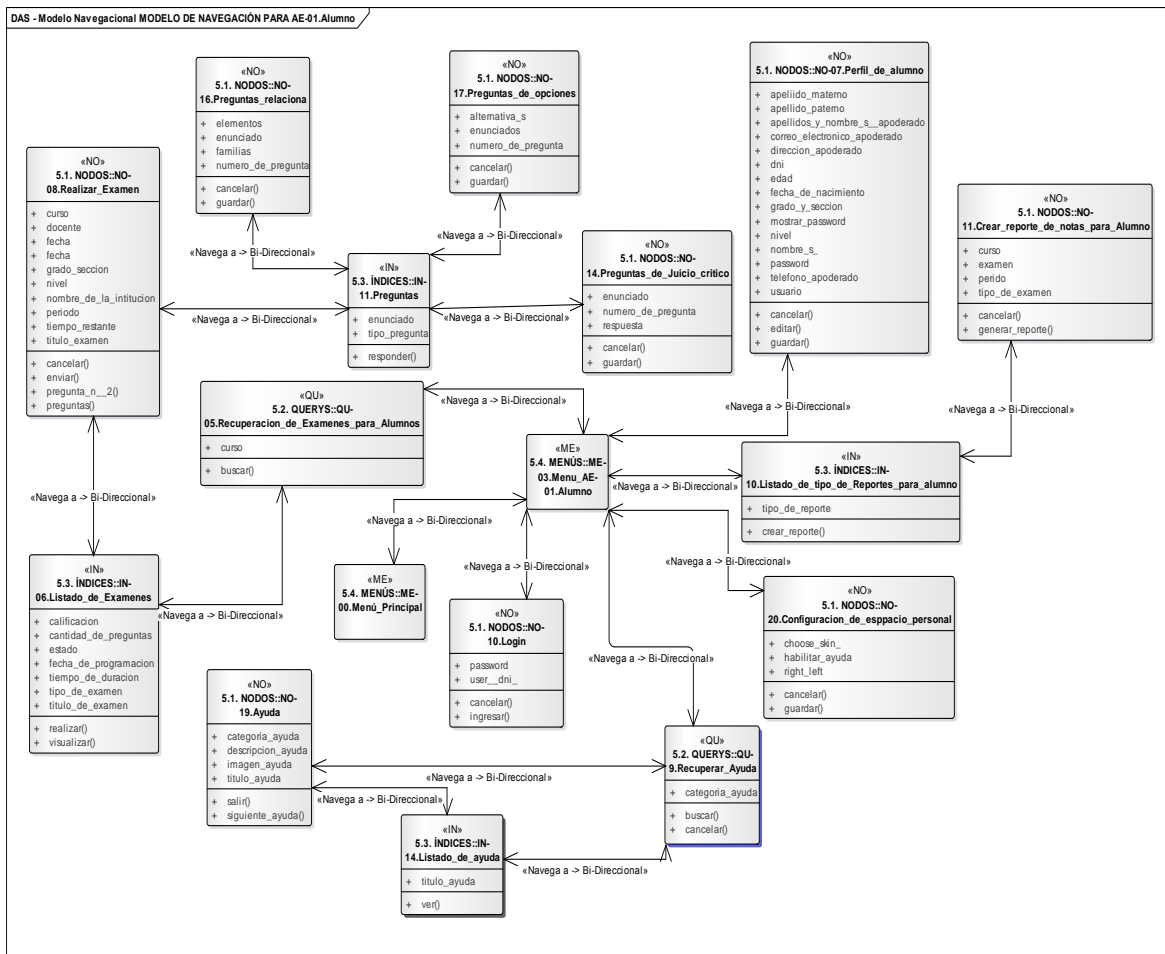


Diagrama 36 Modelo de Navegación para Alumnos, NDT.



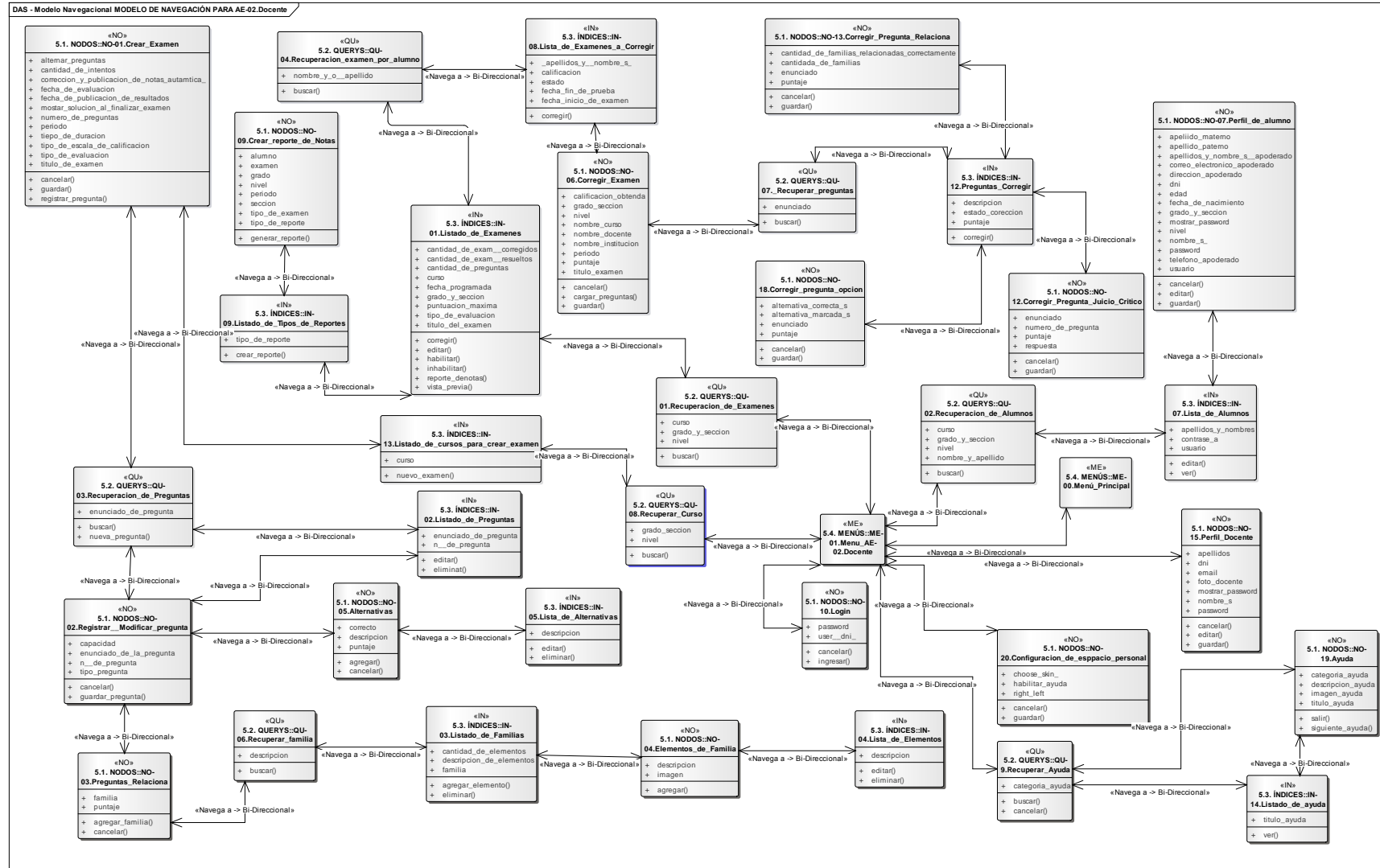


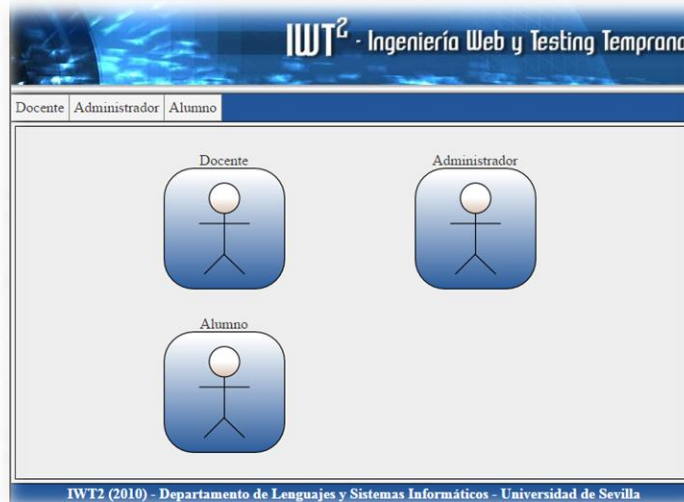
Diagrama 37Modelo de Navegación para Docentes, NDT.





#### 4. PROTOTIPOS NAVEGACIONALES

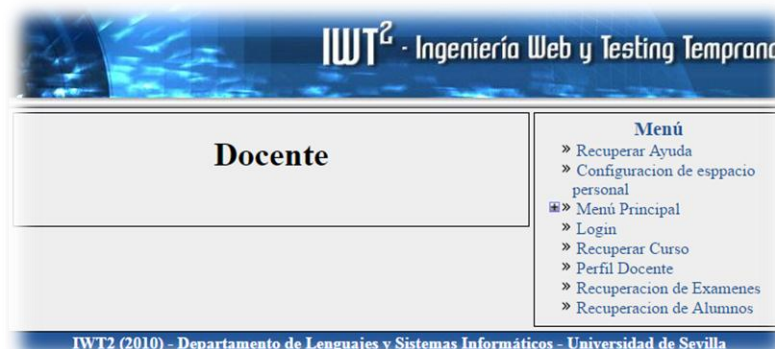
(IWT2, 2016) El Objetivo de los prototipos generados herramienta NDT –Prototypes, es mostrar la navegación del sistema y otorgar una primera vista del aplicativo. En los prototipos generados se puede ver la inclusión de los mecanismos de usabilidad Help y User Profile.



GUI 1 Modelo de Navegación para Alumnos, NDT



GUI 2 Login del Sistema de Exámenes Online, NDT.



GUI 3 Menú del Docente, NDT.



GUI 4Menú del Alumno, NDT.



GUI 5Listado de Cursos del docente a cargo, disponibles para Crear Nuevo Examen, NDT.



GUI 6Crear Exámenes, NDT

### Recuperacion de Exámenes

Grado y Sección: -- ▾

Nivel: -- ▾

Curso: -- ▾

#### Menú

- » Recuperar Ayuda
- » Configuración de espacio personal
- » Menú Principal
- » Login
- » Recuperar Curso
- » Perfil Docente
- » Recuperacion de Exámenes
- » Recuperacion de Alumnos

---

### Listado de Exámenes

Resultado

**Fecha programada:** valor Fecha programada

**Cantidad de Exam. Corregidos:** valor Cantidad de Exam. Cor

Curso: valor Curso

**Título del Examen:** valor Título del Examen

**Tipo de Evaluación:** valor Tipo de Evaluación

**Puntuación máxima:** valor Puntuación máxima

**Cantidad de preguntas:** valor Cantidad de preguntas

**Cantidad de Exam. Resueltos:** valor Cantidad de Exam. Res

**Grado y Sección:** valor Grado y Sección

GUI 7 Recuperación y Listado de Exámenes, NDT

## IWT<sup>2</sup> - Ingeniería Web y Testing Temprano

### Listado de Tipos de Reportes

Resultado

**Tipo de Reporte:** valor Tipo de Reporte

Resultado

**Tipo de Reporte:** valor Tipo de Reporte

Resultado

**Tipo de Reporte:** valor Tipo de Reporte

Resultado

**Tipo de Reporte:** valor Tipo de Reporte

[Atrás](#)

#### Menú

- » Recuperar Ayuda
- » Configuración de espacio personal
- » Menú Principal
- » Login
- » Recuperar Curso
- » Perfil Docente
- » Recuperacion de Exámenes
- » Recuperacion de Alumnos

IWT2 (2010) - Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla

GUI 8 Recuperación y Listado de Exámenes, NDT

## IWT<sup>2</sup> - Ingeniería Web y Testing Temprano

### Crear reporte de Notas

Sección: -- ▾

Examen: -- ▾

Periodo: -- ▾

Alumno: -- ▾

Nivel: -- ▾

Grado: -- ▾

Tipo de Reporte: -- ▾

Tipo de Examen: -- ▾

[Atrás](#)

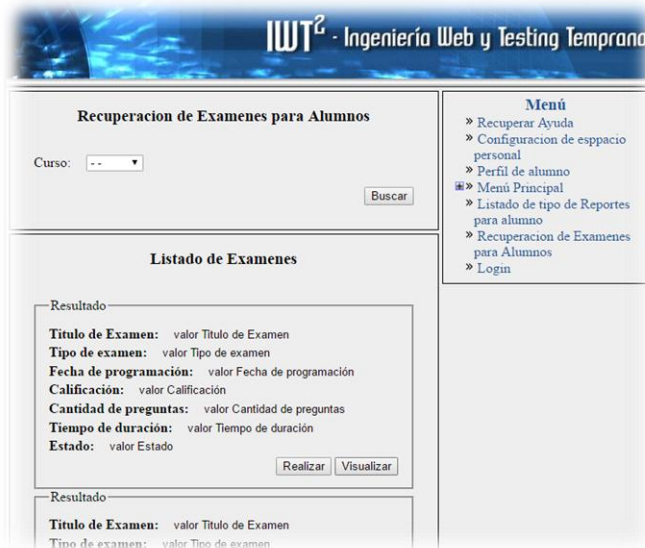
#### Menú

- » Recuperar Ayuda
- » Configuración de espacio personal
- » Menú Principal
- » Login
- » Recuperar Curso
- » Perfil Docente
- » Recuperacion de Exámenes
- » Recuperacion de Alumnos

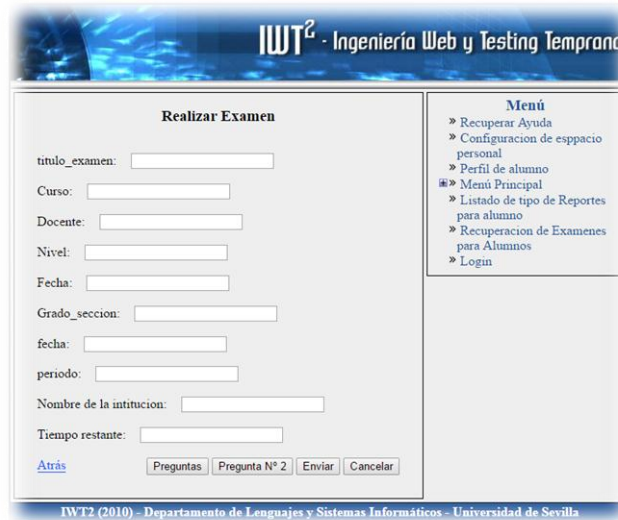
IWT2 (2010) - Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla

GUI 9 Filtros para Generar reporte de Notas, NDT.





GUI 10 Recuperación y Listado de Exámenes para Alumnos, NDT



IWT2 (2010) - Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla

GUI 11 Realizar Examen, NDT.

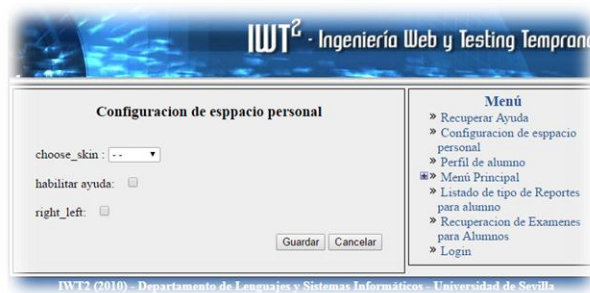


GUI 12 Recuperar y Listar Ayuda, NDT





GUI 13 Mostrar Ayuda, NDT



GUI 14 Registrar configuración de espacio personal, NDT.

## B. ANALISIS DEL SISTEMA,UWE

Según la UWE la etapa de análisis está conformado por tres tipos de modelos que son: el modelo de contenido, el modelo navegacional y el modelo de estructura de procesos.

### 1. MODELO DE CONTENIDO:

Según (Urvina Barrionuevo & Gavilanes Torres, 2013, pág. 58) el modelo de contenido define la estructura de los datos, basándose en el análisis realizado en la etapa anterior para luego determinar la información más importante que se requerirá en el sistema web.

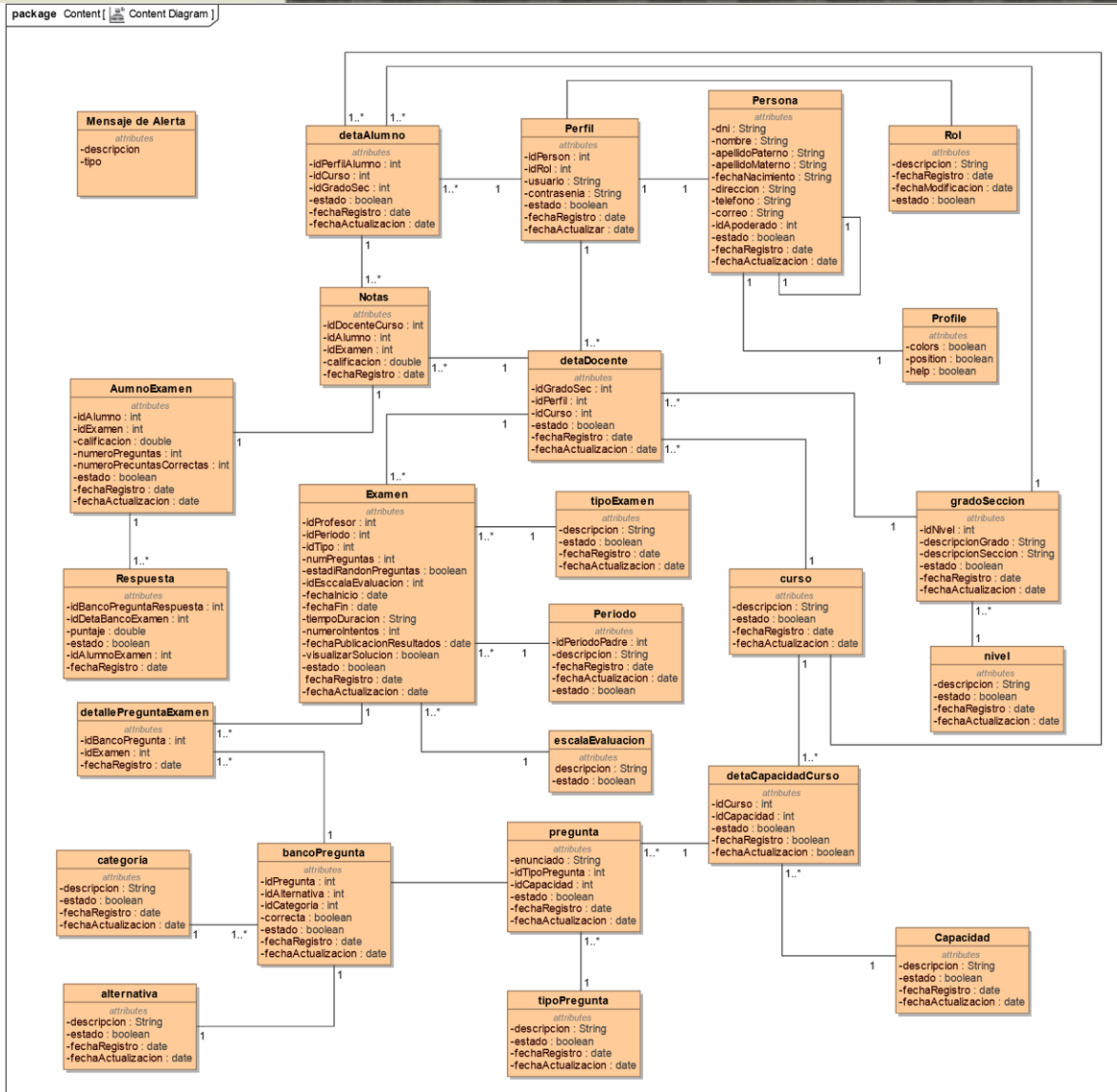


Diagrama 38 Modelo de Contenido, UWE.

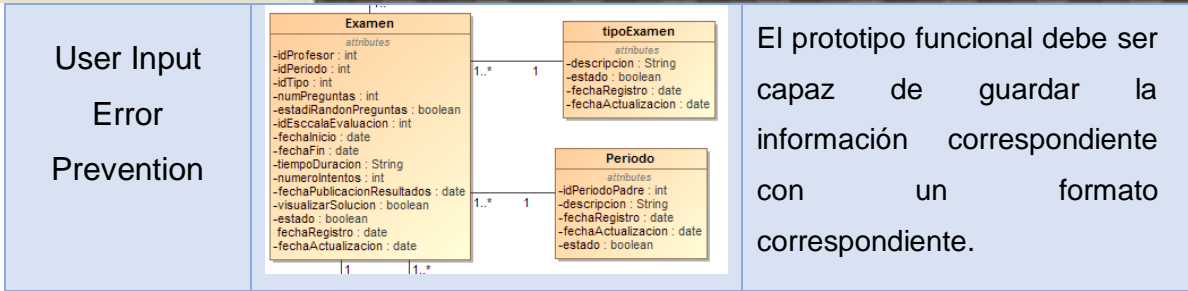
En el diagrama de contenido se han incluido los requerimientos funcionales de usabilidad generados a partir de la aplicación de UPSEPS, que hacen referencia a los mecanismos de usabilidad:

- a) Help
- b) User Profile
- c) Warning
- d) System Status Feedback



MECANISMO DE USABILIDAD	REQUERIMIENTO DE ALMACENAMIENTO DE INFORMACIÓN	DESCRIPCIÓN
<p>System Status feedback</p> <p>Warning</p>	<pre> classDiagram     class Mensaje de Alerta {         -descripcion : String         -idTipoAlerta : int         -icono : byte         -estado : boolean     }     class TipoAlerta {         -descripcion : String     }     Mensaje de Alerta "1..*" -- "1" TipoAlerta         </pre>	<p>El prototipo funcional debe ser capaz de guardar la información correspondiente de acuerdo a los diferentes mensajes de alertas que se puedan mostrar en el sistema.</p>
<p>Help</p>	<pre> classDiagram     class Ayuda {         -descripcion : String         -icono : byte         -estado : boolean     }         </pre>	<p>El prototipo funcional debe ser capaz de guardar la información correspondiente a la ayuda que el sistema puede dar al usuario que haga uso del sistema.</p>
<p>User Profile</p>	<pre> classDiagram     class Persona {         -dni : String         -nombre : String         -apellidoPaterno : String         -apellidoMaterno : String         -fechaNacimiento : String         -direccion : String         -telefono : String         -correo : String         -idApoderado : int         -estado : boolean         -fechaRegistro : date         -fechaActualizacion : date     }     class Profile {         -colors : boolean         -position : boolean         -help : boolean     }     Persona "1" -- "1" Profile         </pre>	<p>El prototipo funcional debe ser capaz de guardar la información correspondiente a la personalización el usuario le pueda dar al sistema según su gusto, en cuanto al Choose Skin (tema o color) según su preferencia así como la locación del menú, ya sea izquierda o derecha.</p>





### 5.4.2.1.1. MODELO DE NAVEGACIÓN

(Urvina Barrionuevo & Gavilanes Torres, 2013, pág. 59) El modelo de navegación hace uso de dos tipos de meta-clases abstractas “nodos” y “link”, a través de ellos se define la estructura navegacional que una aplicación web puede tener.

En los nodos de los diferentes diagramas de navegación que veremos a continuación, veremos el uso de diferentes estereotipos como son:

- a) «navigationClass» (□): Una clase navegacional es la representación de un nodo navegable de la estructura de hipertexto para así poder establecer conexión entre el modelo de contenido y el modelo de navegación.
- b) «index» (≡): Permite la selección de una instancia de clase de contenido a partir de un conjunto de instancias que ha sido recopilada durante la navegación anterior.
- c) «menu» (☰): Se utiliza para controlar rutas de navegación alternativos.
- d) «query» (⌘): Se utiliza cuando se va a recuperar el contenido de una fuente de datos.
- e) «processclass» (Σ): Sirve para ilustrar si los datos persistentes de la aplicación son modificados o no.
- f) «processLink»: Te permite conectar dos nodos
- g) «NavigationLink»: Es un enlace que conecta a cualquier tipo de nodos.





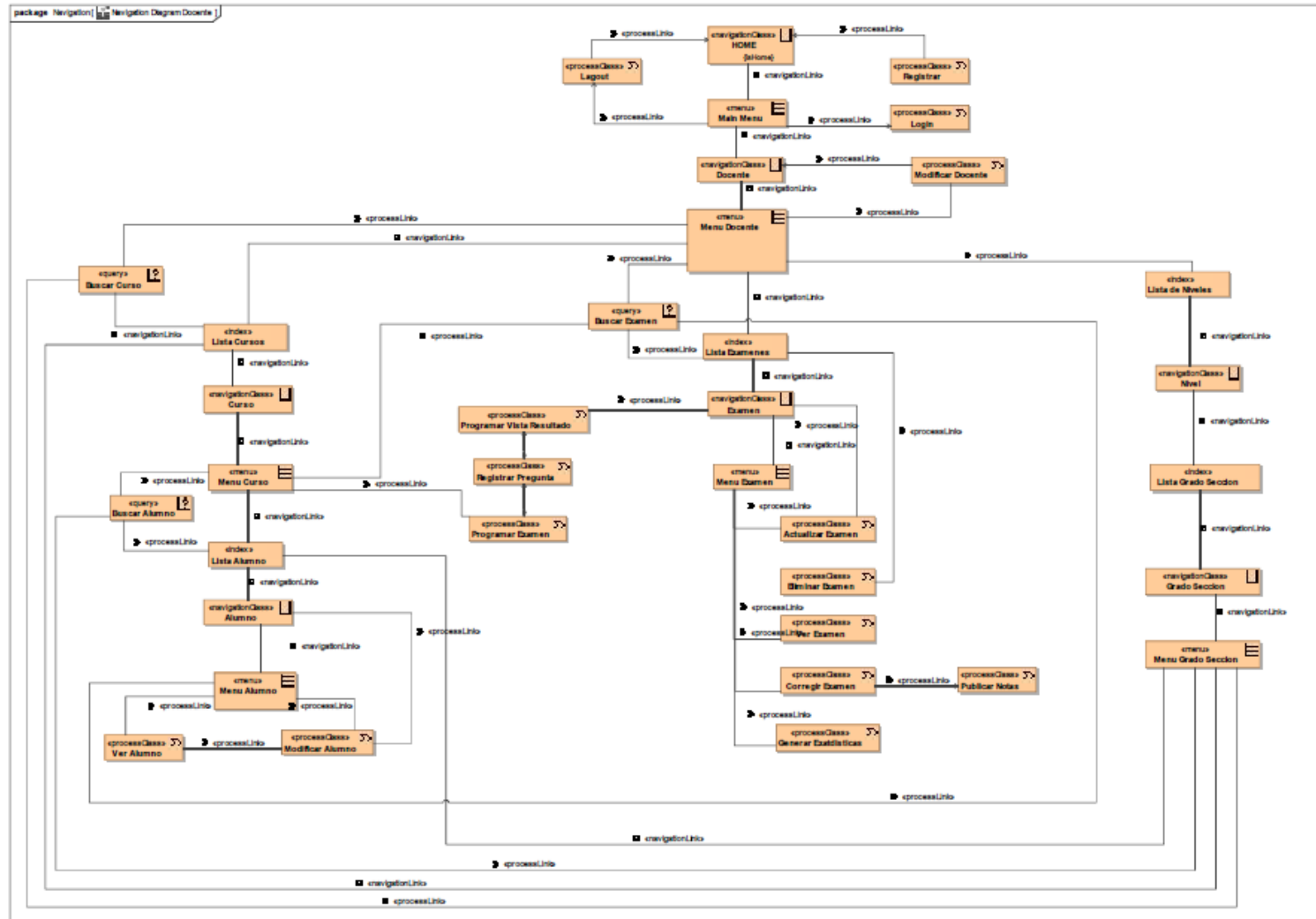


Diagrama 39 Modelo Navegacional del Docente, UWE.



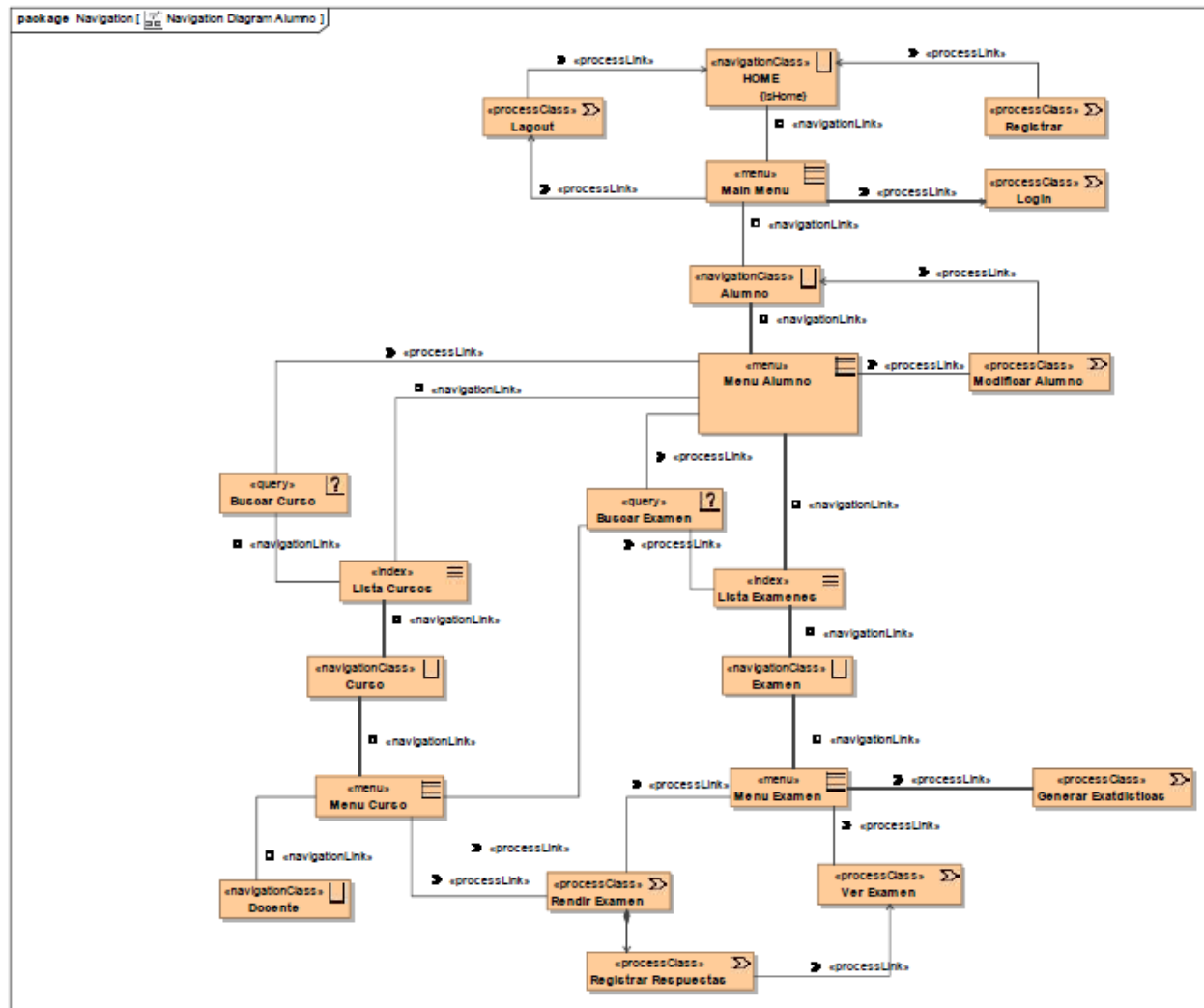


Diagrama 40 Modelo Navegacional del Alumno



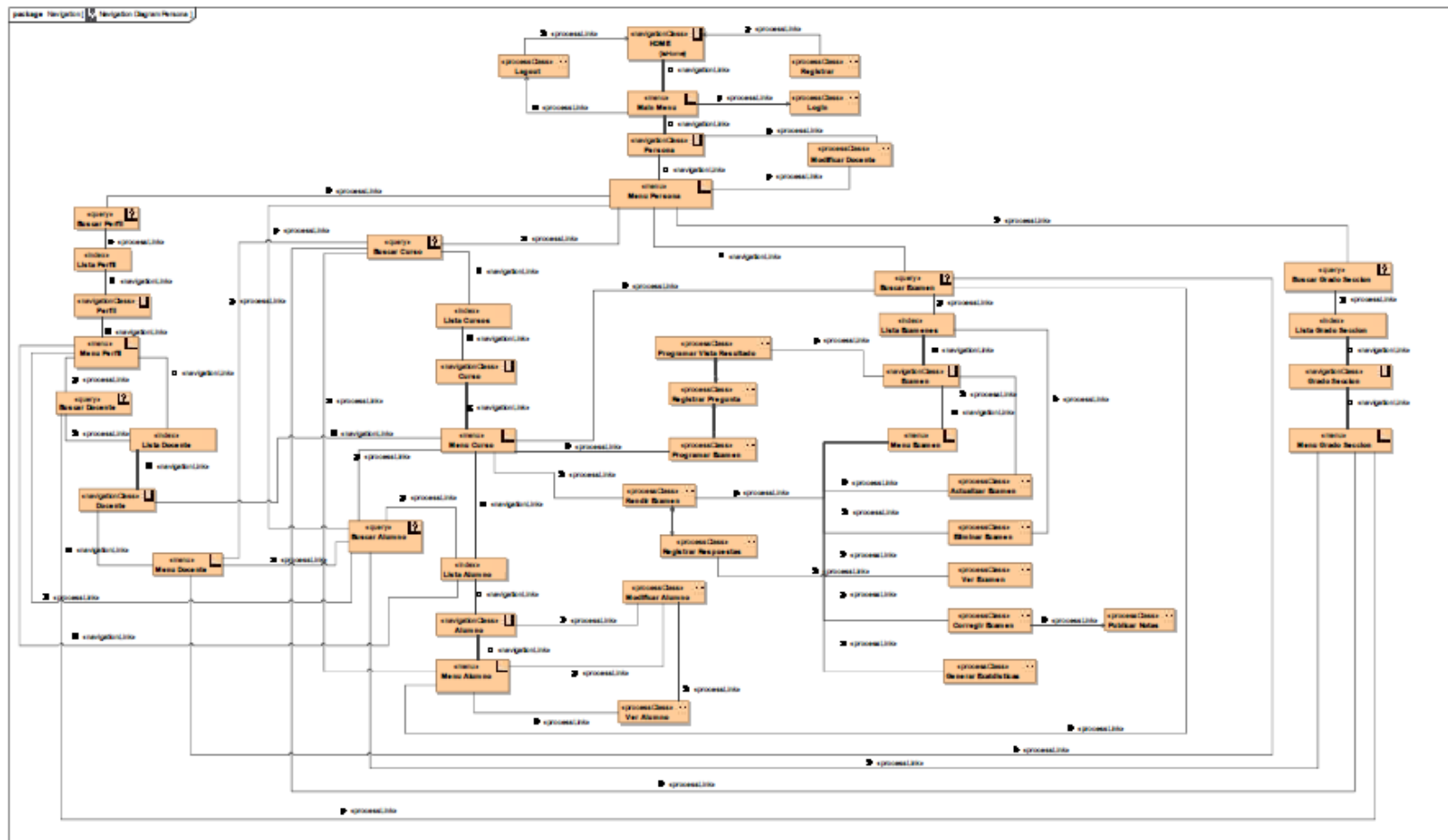


Diagrama 41 Modelo Navegacional del Administrador, UWE.



En el diagrama navegacional, no se ha incorporado ningún mecanismo de usabilidad, debido a que aquí solo se plasma la interacción que el usuario deberá tener al usar el sistema.

## 2. ESTRUCTURA DEL DIAGRAMA DE PROCESOS

En este diagrama se define las relaciones entre las diferentes clases de proceso.

En los diferentes diagramas de navegación que veremos a continuación, veremos el uso de diferentes estereotipos como son:

- a) «processclass» ( $\Sigma$ ): Sirve para identificar los diferentes procesos que se plasman en el modelo Navegacional y así poder definir los datos que se solicitara al usuario durante el proceso.

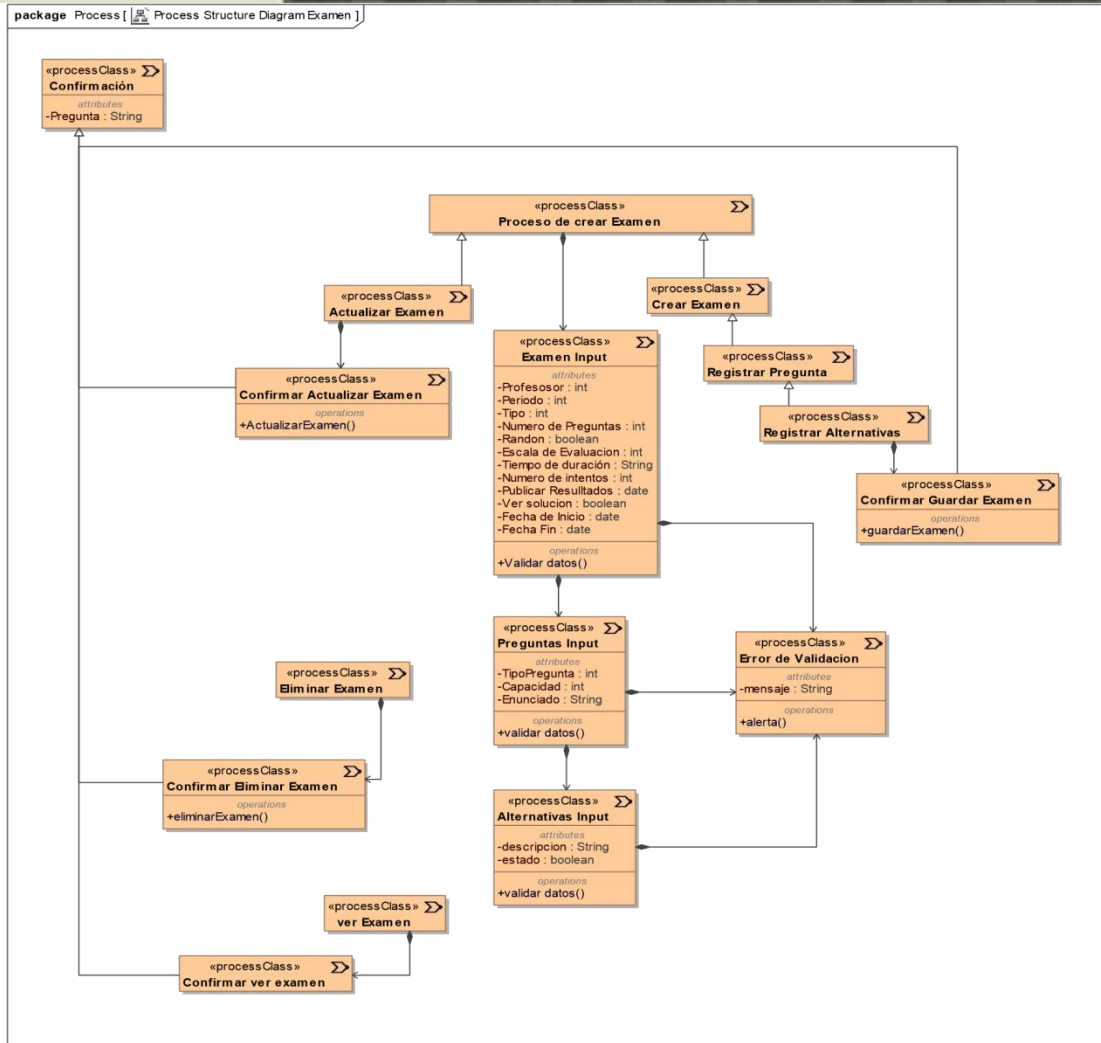


Diagrama 42 Estructura del Proceso de Crear Examen, UWE.

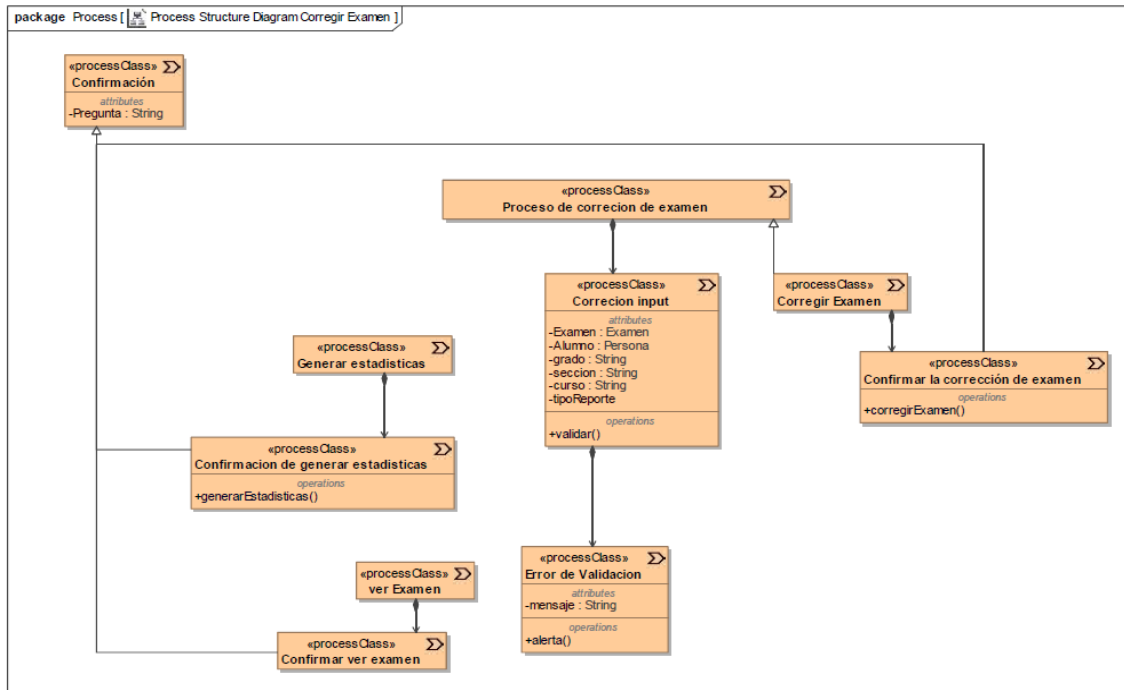


Diagrama 43 Estructura del Proceso Corregir Examen, UWE.

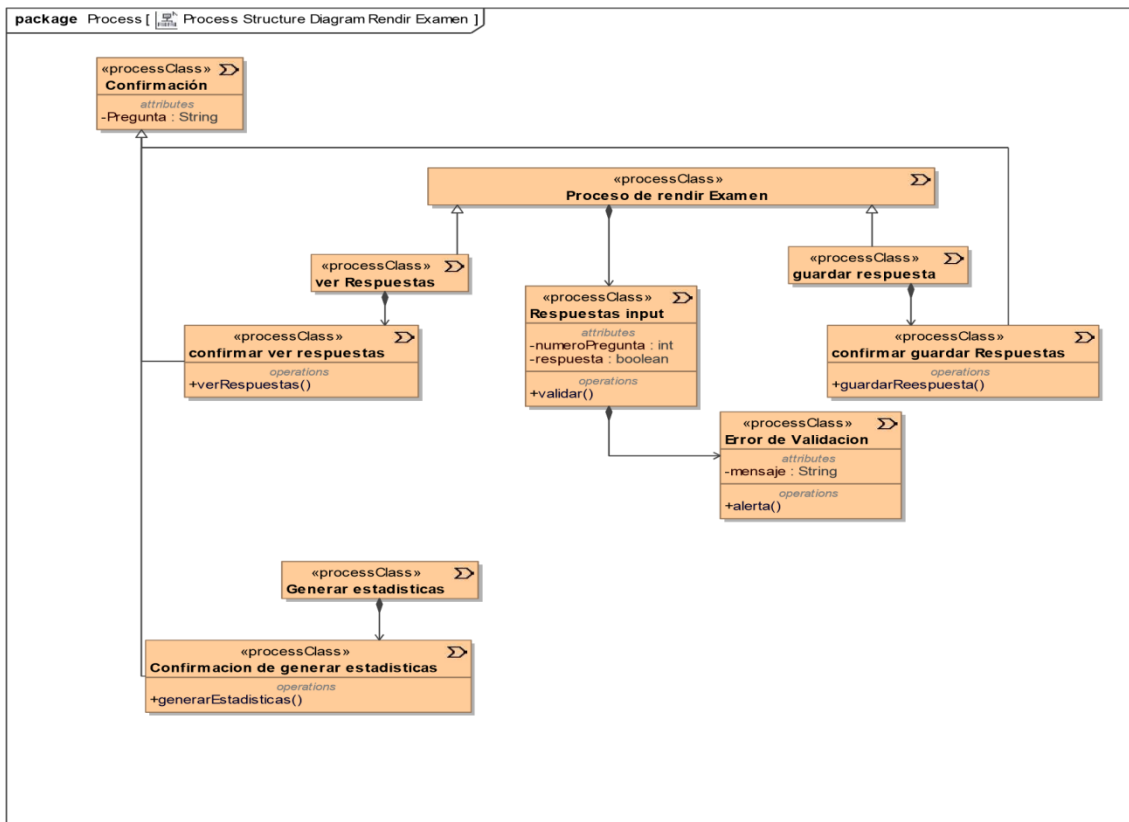


Diagrama 44 Estructura del Proceso de Rendir Examen, UWE.



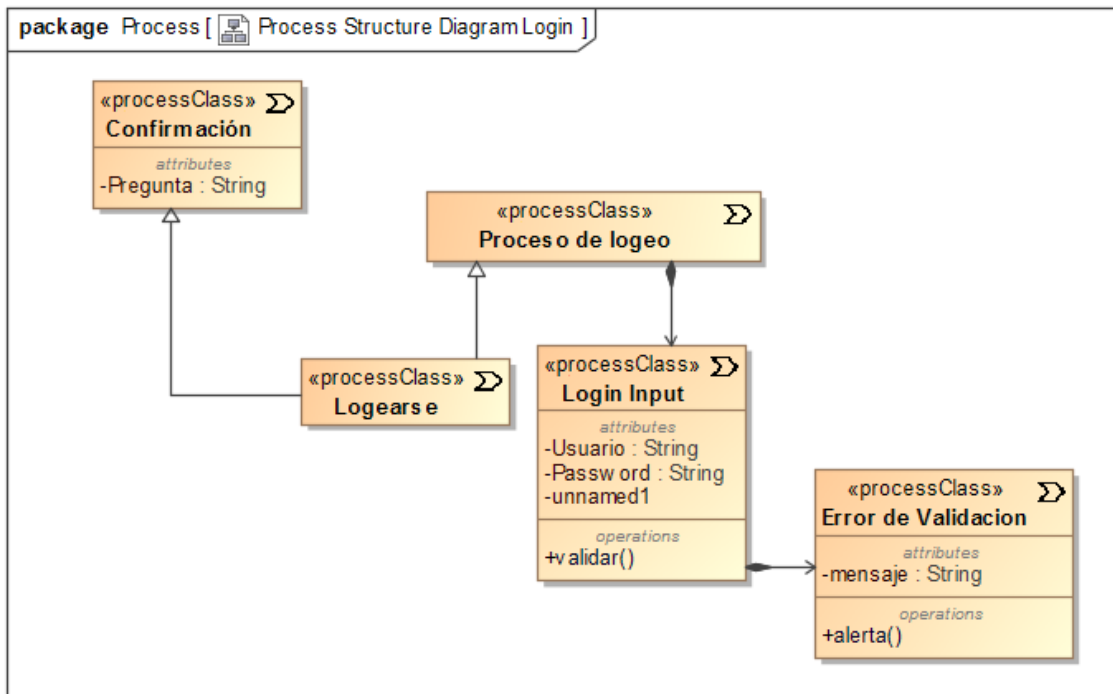


Diagrama 45 Estructura del proceso de Login, UWE

En los diagramas de estructura de proceso se han incluido los requerimientos funcionales de usabilidad generados a partir de la aplicación de UPSEPS, que hacen referencia a los mecanismos de usabilidad:

- a) Warning.
- b) System Status Feedback:
- c) User Input Error Prevention

Nombre del diagrama	Componente representativo	Mecanismo de Usabilidad
Estructura de procesos Crear Examen		Warning
		System Status Feedback

	<pre> «processClass» &gt;&gt; Examen input attributes -Profesor : Int -Periodo : Int -Tipo : Int -Numero de Preguntas : Int -Randon : boolean -Escala de Evaluacion : Int -Tiempo de duracion : String -Numero de Intentos : Int -Publicar Resultados : date -Ver solucion : boolean -Fecha de Inicio : date -Fecha Fin : date operations +Validar datos()         </pre>	<p>User Input Error Prevention</p>
<p><b>Estructura de procesos Rendir Examen</b></p>	<pre> «processClass» &gt;&gt; confirmar guardar Respuestas operations +guardarReespuesta()         </pre>	<p>Warning</p>
	<pre> «processClass» &gt;&gt; Error de Validacion attributes -mensaje : String operations +alerta()         </pre>	<p>System Status Feedback</p>
	<pre> «processClass» &gt;&gt; Respuestas input attributes -numeroPregunta : int -respuesta : boolean operations +validar()         </pre>	<p>User Input Error Prevention</p>
<p><b>Estructura de procesos Corregir Examen</b></p>	<pre> «processClass» &gt;&gt; Confirmar la corrección de examen operations +corregirExamen()         </pre>	<p>Warning</p>
	<pre> «processClass» &gt;&gt; Error de Validacion attributes -mensaje : String operations +alerta()         </pre>	<p>System Status Feedback</p>
	<pre> «processClass» &gt;&gt; Correccion input attributes -Examen : Examen -Alumno : Persona -grado : String -seccion : String -curso : String -tipoReporte operations +validar()         </pre>	<p>User Input Error Prevention</p>





<b>Estructura de procesos de Login</b>	<pre> «processClass» &gt;&gt; Confirmación   attributes   -Pregunta : String         </pre>	Warning
	<pre> «processClass» &gt;&gt; Error de Validacion   attributes   -mensaje : String   operations   +alerta()         </pre>	System Status Feedback
	<pre> «processClass» &gt;&gt; Login Input   attributes   -Usuario : String   -Passw ord : String   -tipoUsuario : boolean   operations   +validar()         </pre>	User Input Error Prevention

### 5.4.5. DISEÑO DEL SISTEMA

#### A. DISEÑO DEL SISTEMA: NDT

En esta fase se engloban aspectos concretos de cómo lo analizado anteriormente se implementará en la máquina. Está orientado a la plataforma concreta con la que se vaya a trabajar y debe corresponder con la estructura del futuro código.

#### 1. DIAGRAMA DE CLASES DE DISEÑO

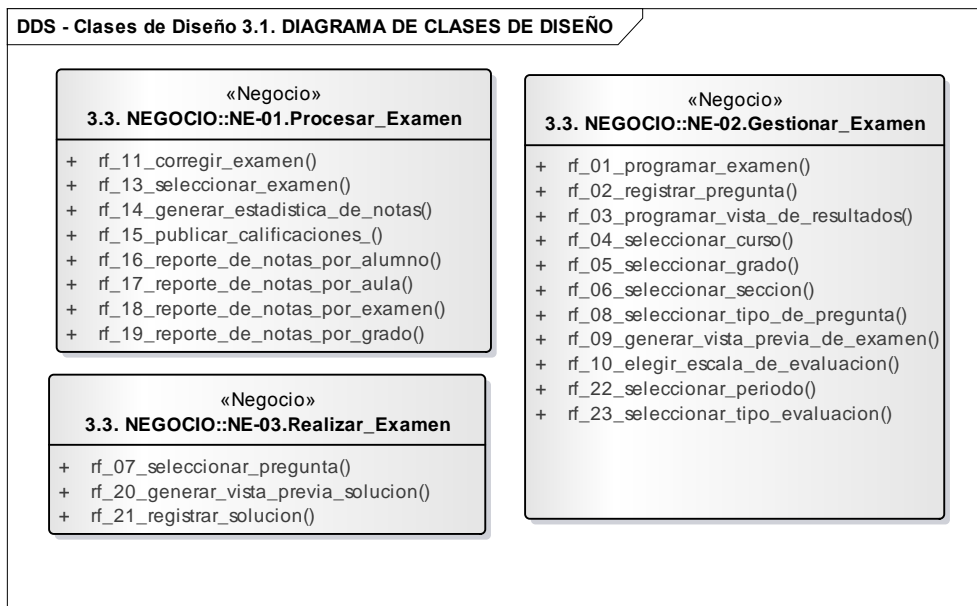


Diagrama 46 Diagrama de clases de diseño, negocio, NDT.



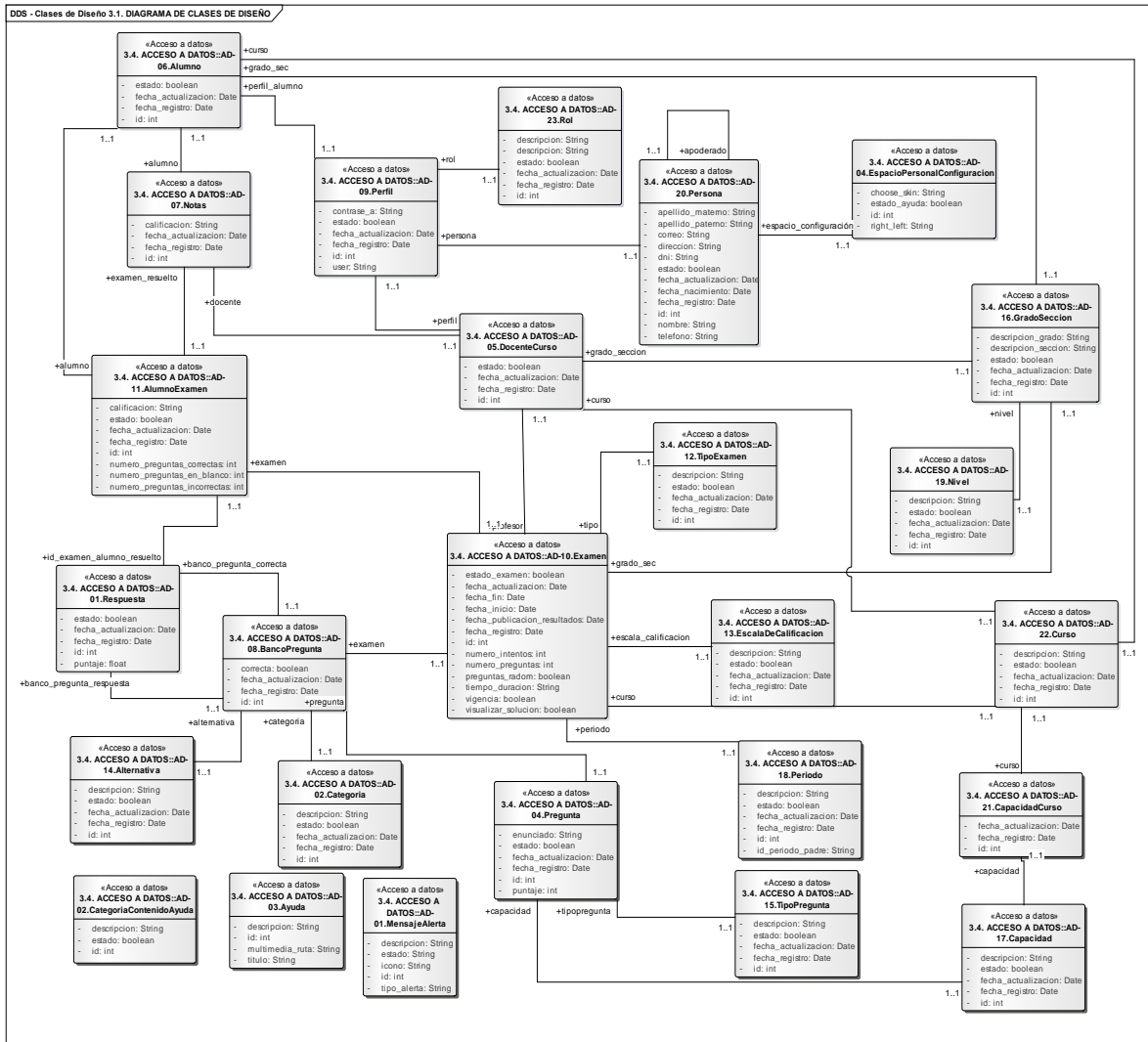


Diagrama 47 Diagrama de clases de diseño, Acceso a datos, NDT



DDS - Clases de Diseño 3.1. DIAGRAMA DE CLASES DE DISEÑO

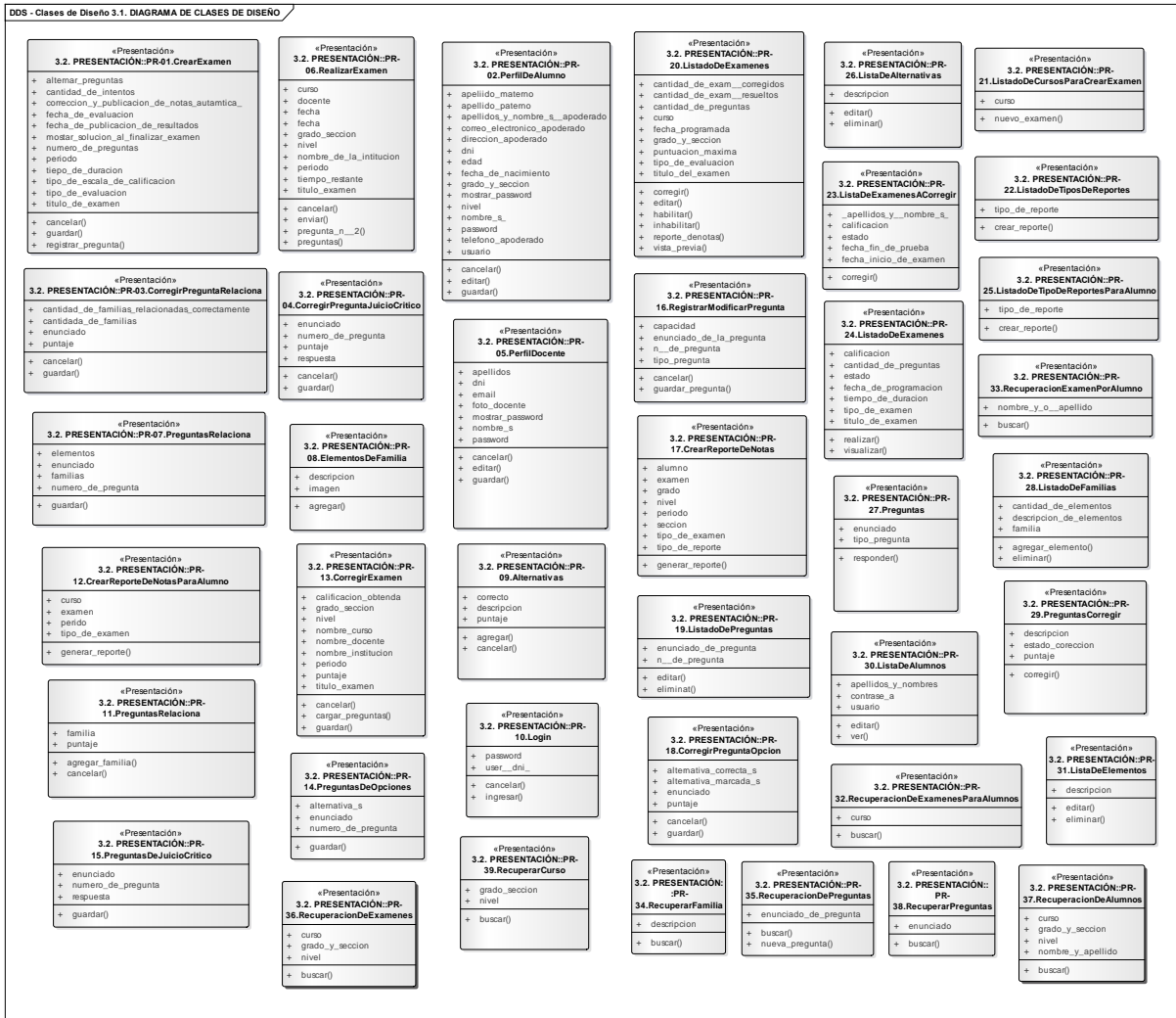


Diagrama 48 Diagrama de clases, Presentación, NDT

## B. DISEÑO DEL SISTEMA: UWE

En esta etapa haremos énfasis en el modelo de presentación que es parte de la metodología UWE.

### 1. DIAGRAMA DE PRESENTACION

A través del diagrama de presentación, podemos encontrar dos diferentes modelos, el modelo de navegación y el modelo que identifica los diferentes procesos que pertenecen a un sistema web.

En los diferentes diagramas de presentación que veremos a continuación, veremos el uso de diferentes estereotipos como son:

a) <<presentacionAlternatives>> (📄)



- b) <<presentationGroup>> (📁)
- c) <<iteratedPresentationGroup>> (📁)
- d) <<inputForm>> (📁)
- e) <<presentationPage>> (📄)
- f) <<<tab>> (⇐)
- g) <<button>> (●)
- h) <<anchor>> (—)
- i) <<text>> (≈)
- j) <<image>> (🖼️)
- k) <<selection>> (👉)
- l) <<textInput>> (abl)

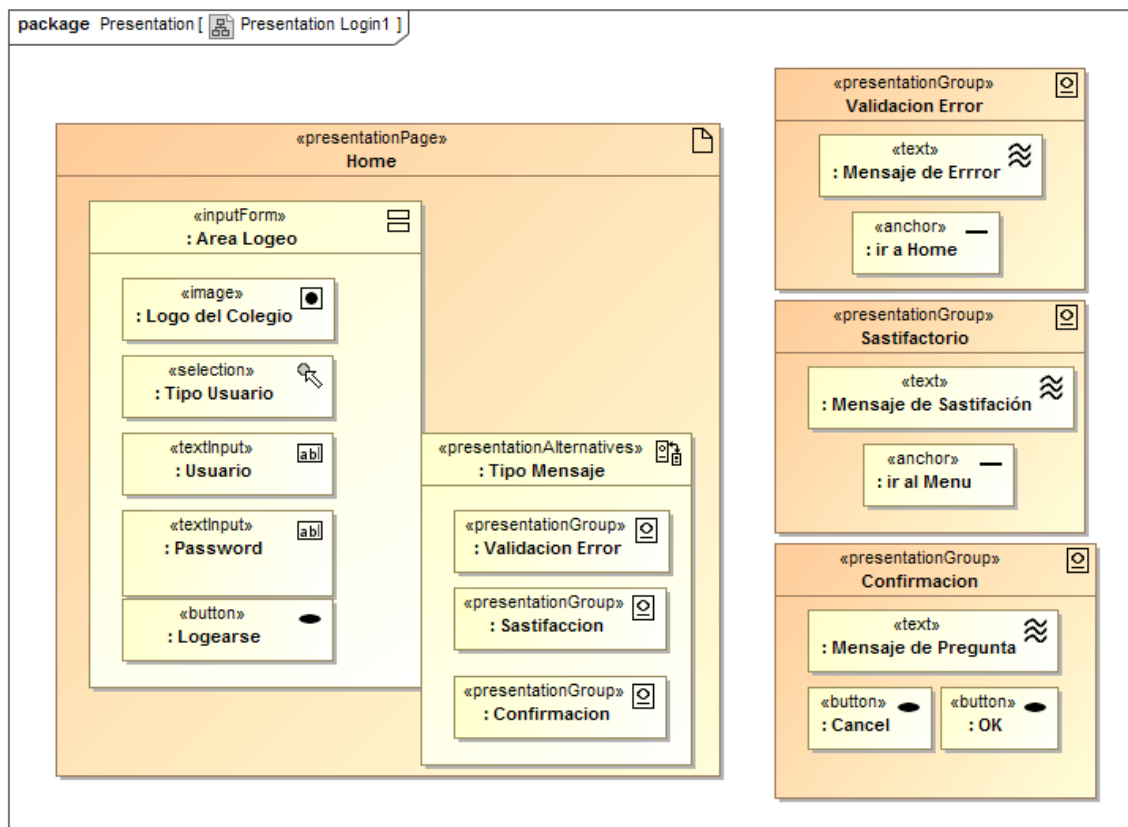


Diagrama 49 Modelo de presentación del Login, UWE

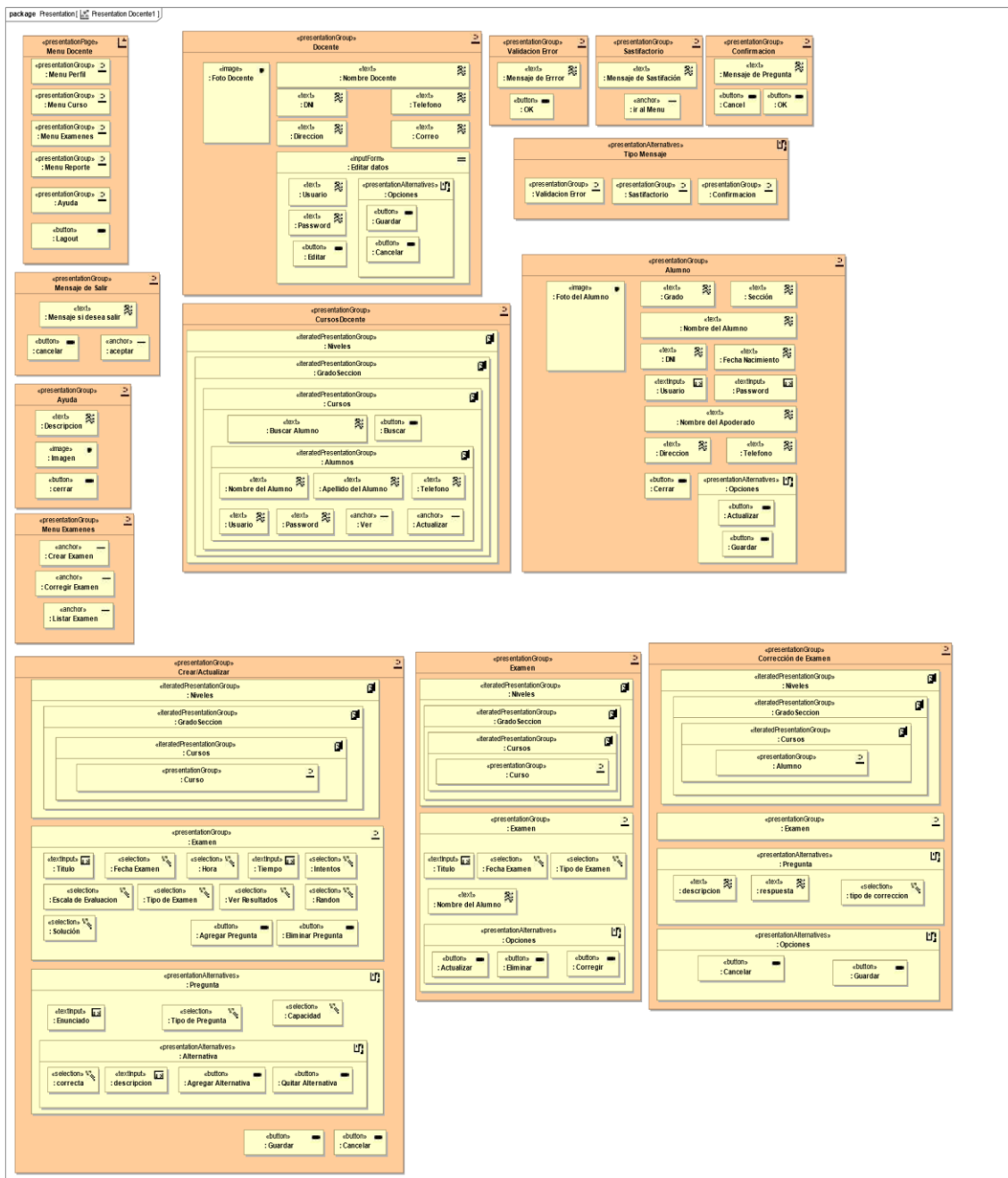


Diagrama 50 Modelo de presentación del docente, UWE.



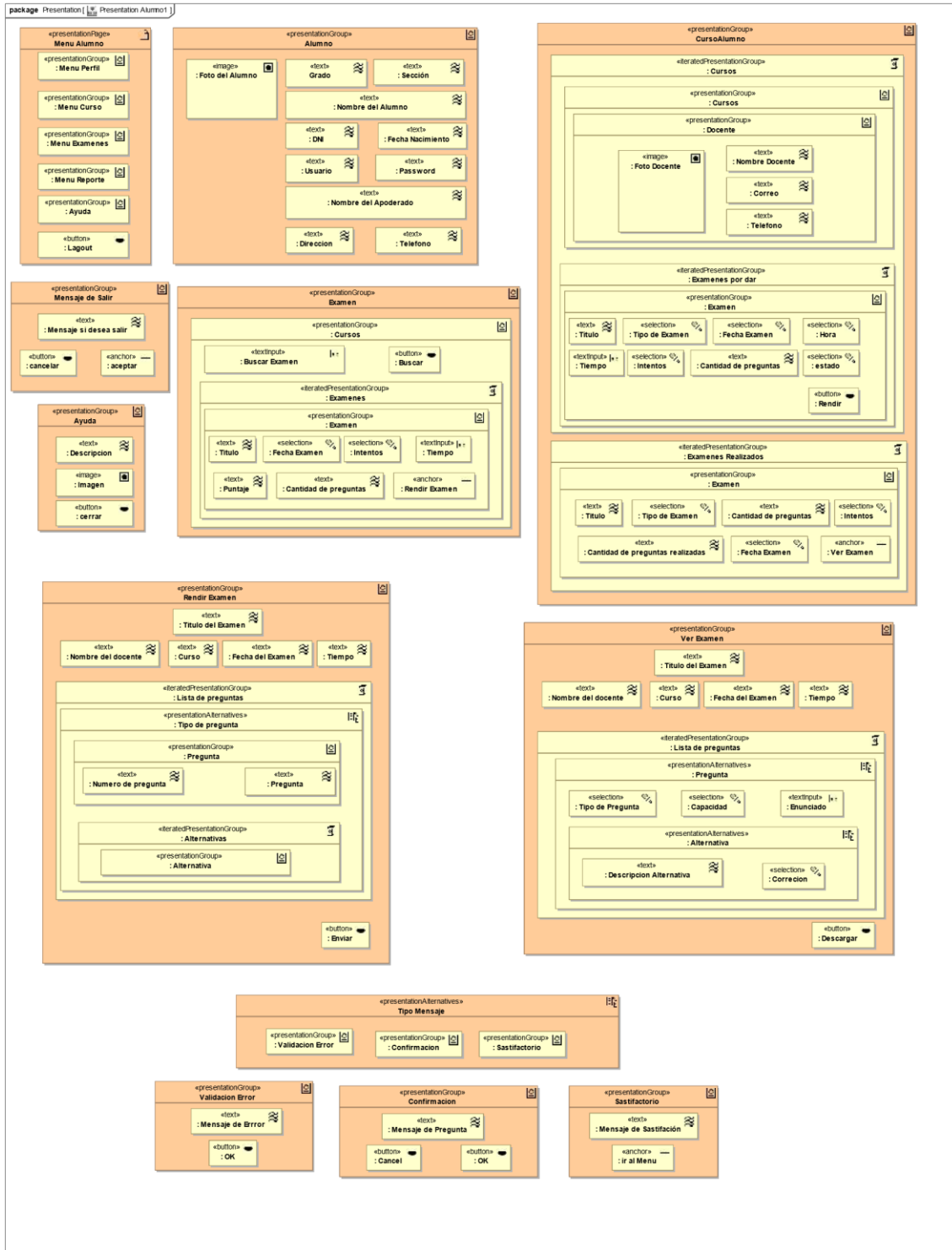


Diagrama 51 Modelo de presentación del Alumno, UWE.

En los diagramas de presentación se han incluido los requerimientos funcionales de usabilidad generados a partir de la aplicación de USEP's, que hacen referencia a los mecanismos de



usabilidad: Warning, System Status Feedback, Interaction Feedback, User Input Error Prevention, Help y User Profile.

Mecanismo de usabilidad	Componente representativo	Nombre del Diagrama	Descripción
<b>Warning</b>		Modelo de Presentación del Logeo	Mecanismo de usabilidad incorporado en cada uno de los modelos de presentación para el aviso de mensajes, sobre acciones que afectaran al sistema.
		Modelo de Presentación del Docente	
		Modelo de Presentación del Alumno.	
<b>System Status Feedback</b>		Modelo de Presentación del Logeo	Mecanismo de usabilidad incorporado en todos los modelos de presentación para el aviso de mensajes, sobre errores al realizar una determinada acción.
		Modelo de Presentación del Docente	
		Modelo de Presentación del Alumno	
<b>User Input Error Prevention</b>		Modelo de Presentación del Logeo	La incorporación de User Input Error Prevention en los modelos de presentación sirve

		<p>Modelo de Presentación del Docente</p>	<p>para especificar la entrada de texto en cada uno de los diferentes diagramas.</p>
		<p>Modelo de Presentación del Alumno</p>	
<p><b>Help</b></p>		<p>Modelo de Presentación del Logeo</p>	<p>La incorporación del Help en los modelos de presentación, hace referencia a una ventana de ayuda cuando el usuario solicite como realizar una actividad.</p>
		<p>Modelo de Presentación del Docente</p>	
		<p>Modelo de Presentación del Alumno</p>	
<p><b>User Profile</b></p>		<p>Modelo de Presentación del Logeo</p>	<p>La incorporación de User Profile en el pódolo de presentación, hace referencia a cambios que el usuario podría realizar el diseño del sistema.</p>
		<p>Modelo de Presentación del Docente</p>	
		<p>Modelo de Presentación del Alumno</p>	

La NDT (Navigational Development Techniques) y UWE (UML-BASE Web Engineering) son metodologías que están enfocadas





exclusivamente en el análisis y diseño de la aplicación, estas no toman en cuenta la fase de implementación del sistema, hace referencia (Vilariño De Almeida, 2010), es por ello que en este punto de la investigación se llegó hasta el alcance de dichas metodologías, para proceder a la etapa o desarrollo del sistema se seleccionó la metodología más óptima.

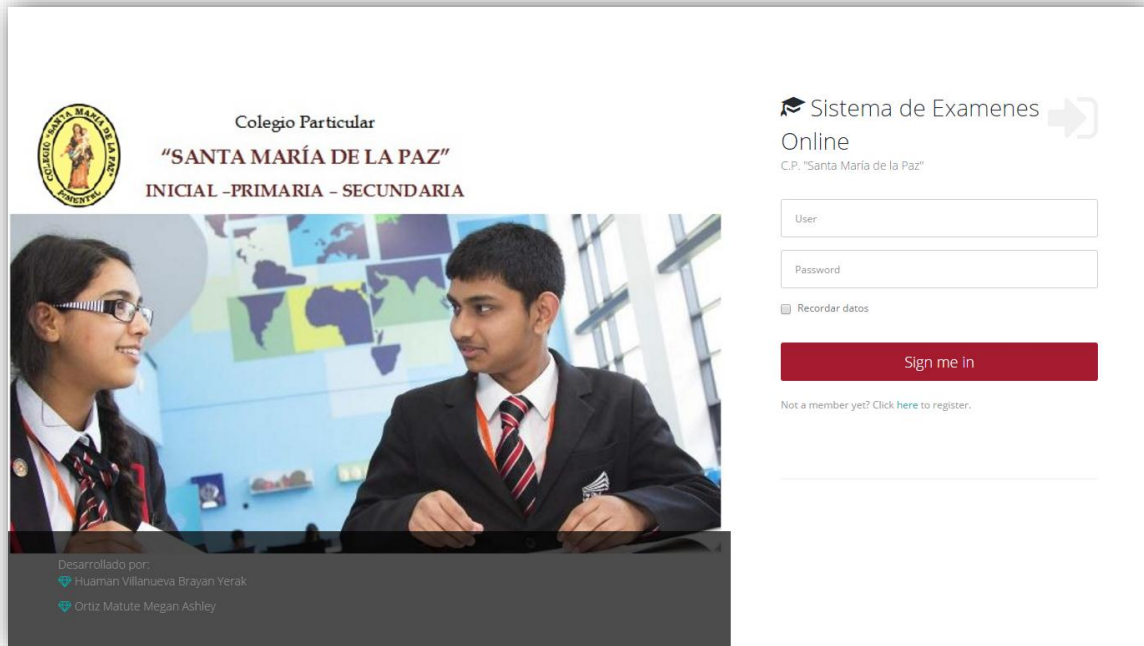
Para ello se realizó la evaluación de ambas metodologías bajo los siguientes puntos: cantidad de artefactos generados que incluye a la usabilidad, tiempo que tomo desarrollar cada metodología de acuerdo a su alcance, así como también el número de presentaciones que se realizó por cada metodología estudiada. La cual se presenta al detalle en el capítulo IV: análisis de resultados.

#### 5.4.6. CODIFICACIÓN DEL SISTEMA

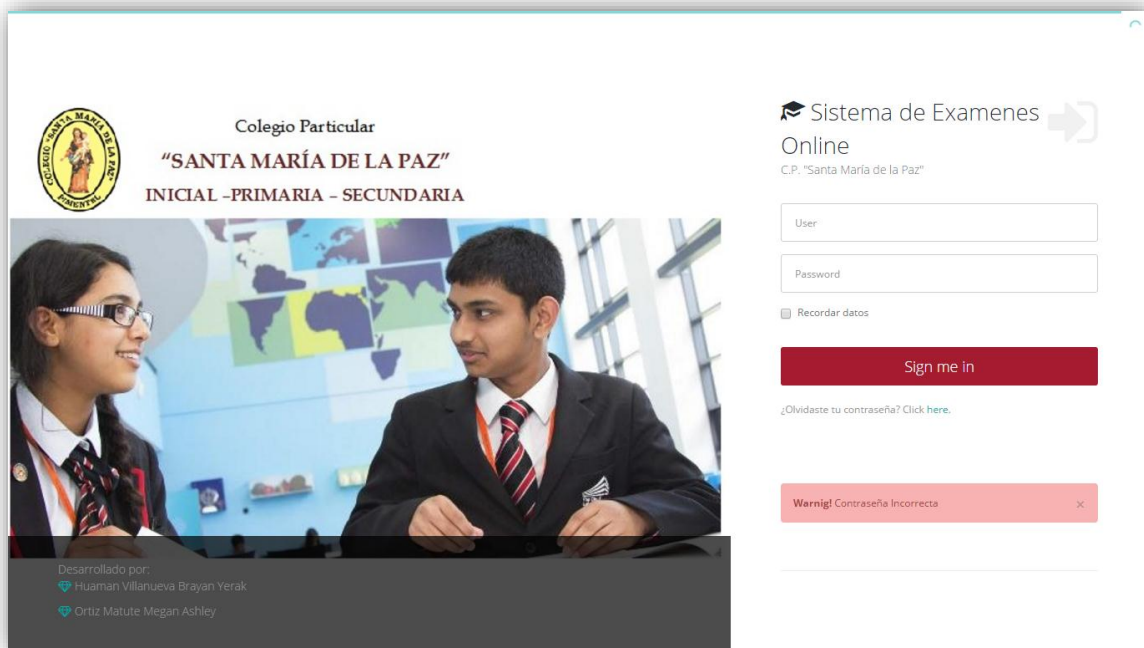
Bajo el análisis realizado en el Capítulo IV: Resultados de la investigación, se determinó que la metodología más eficiente es la NDT, dado a que da más énfasis a la etapa de definición de requerimientos, el tiempo que tomo en realizar la captura de requerimientos, definición de requerimientos, análisis y diseño del sistema con la integración de las directrices fue más corto frente a los resultados obtenidos de la UWE, también se pudo observar que la NDT es la metodología que integro un mayor número de mecanismos de usabilidad.

Una de las característica diferenciadora entre la UWE y la NDT, es que esta última trabaja con matrices de Trazabilidad, la cual relaciona los elementos generados por cada artefacto, con los elementos consecuentes del modelo generado producto de transformaciones, además de que cada elemento está alineado con los objetivos del proyectos, de tal forma que se mantiene la consistencia e integridad de los requerimientos a través de una fase a otra. Es por ello que se selección a la metodología NDT para continuar a la etapa de codificación del caso de estudio, el cual consistió realizar una sistema Web de Exámenes Online, donde nuestro principal stakeholder fue la Institución Educativa particular “Santa María de la Paz”.

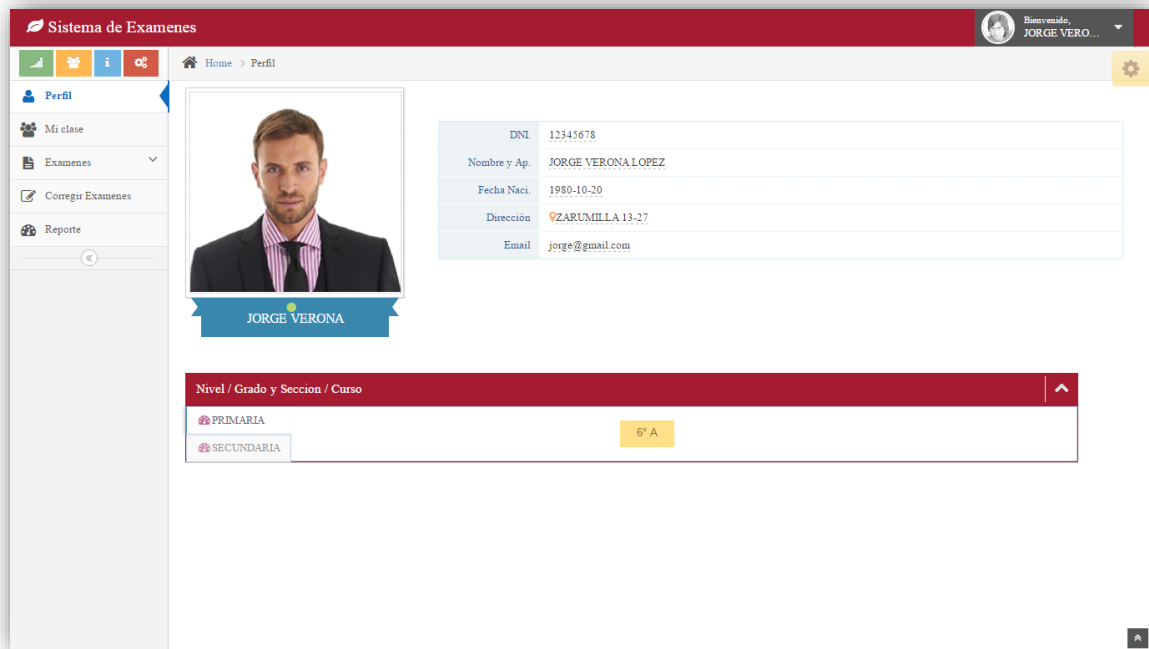
A continuación presentaremos el sistema Web de Exámenes Online evidenciando los puntos donde se resalta la inclusión de los mecanismos de usabilidad en el producto final.



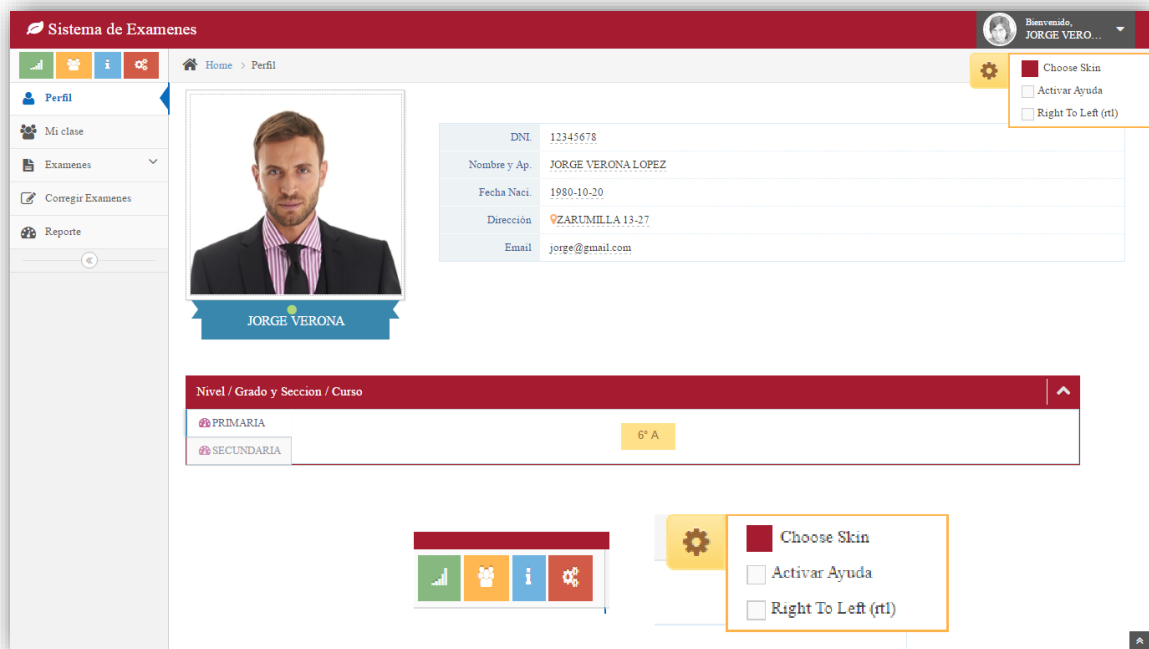
GUI 15 Login del Sistema de Exámenes Online



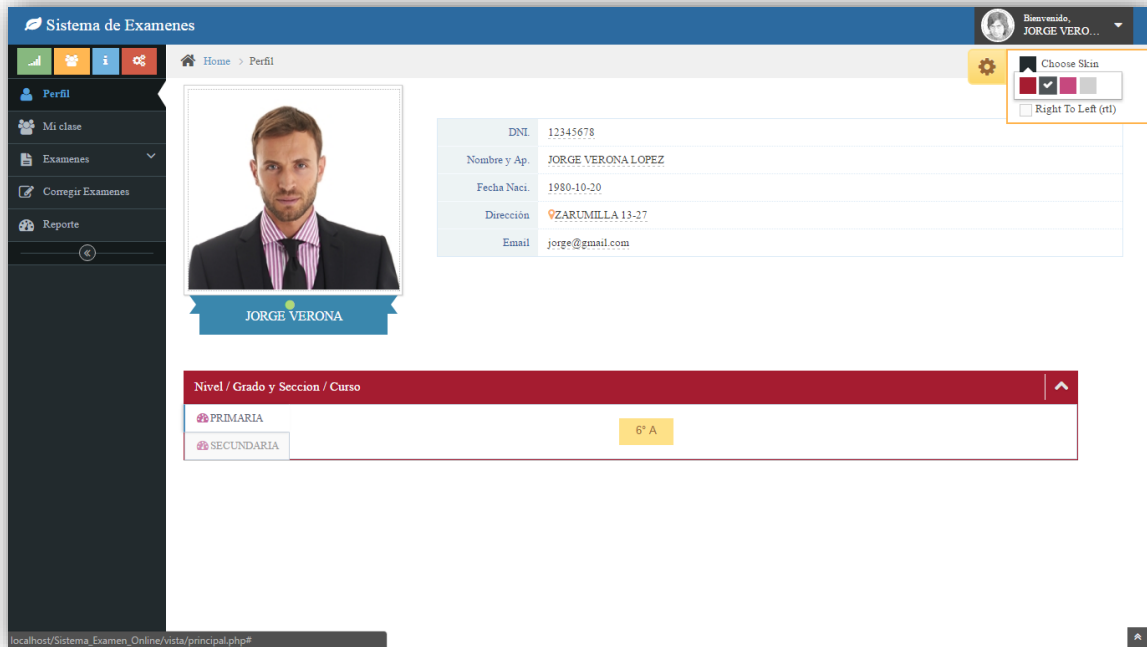
GUI 16 Mecanismo de Usabilidad Warning, Sistema de Exámenes Online



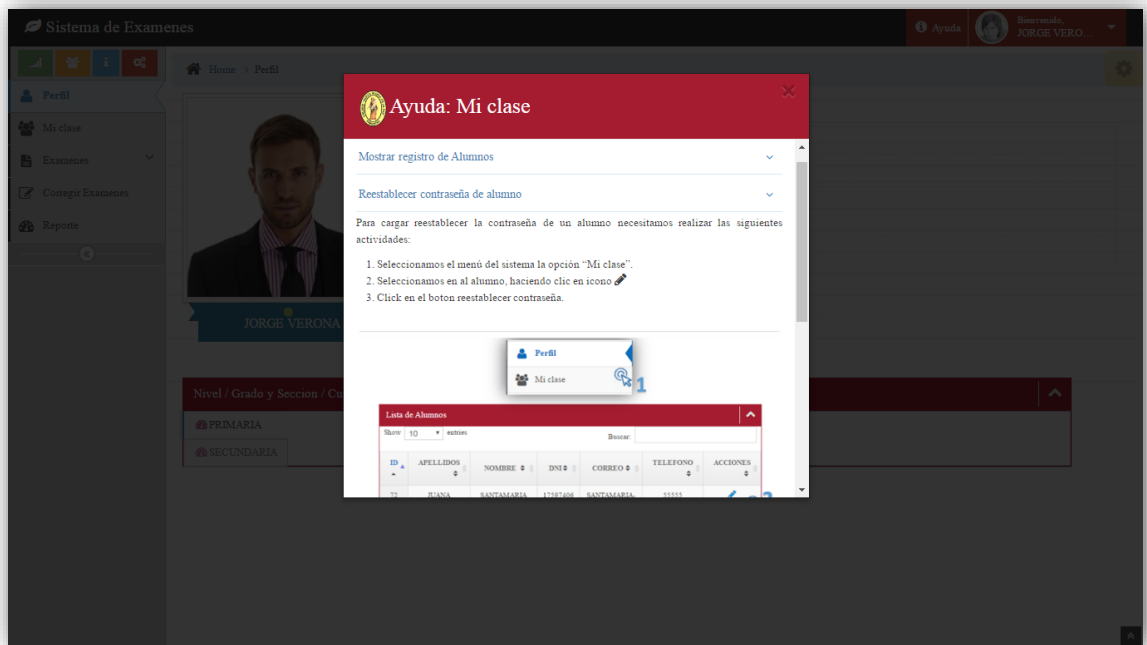
GUI 17 Interfaz principal del sistema, perfil del usuario



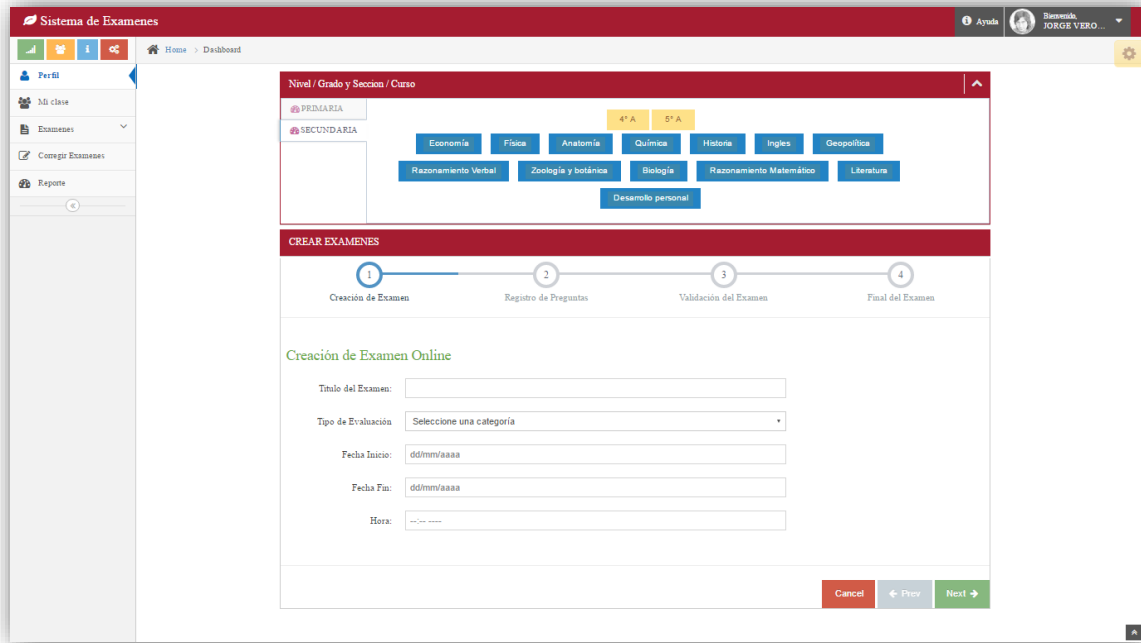
GUI 18 Mecanismo de Usabilidad, User Profile.



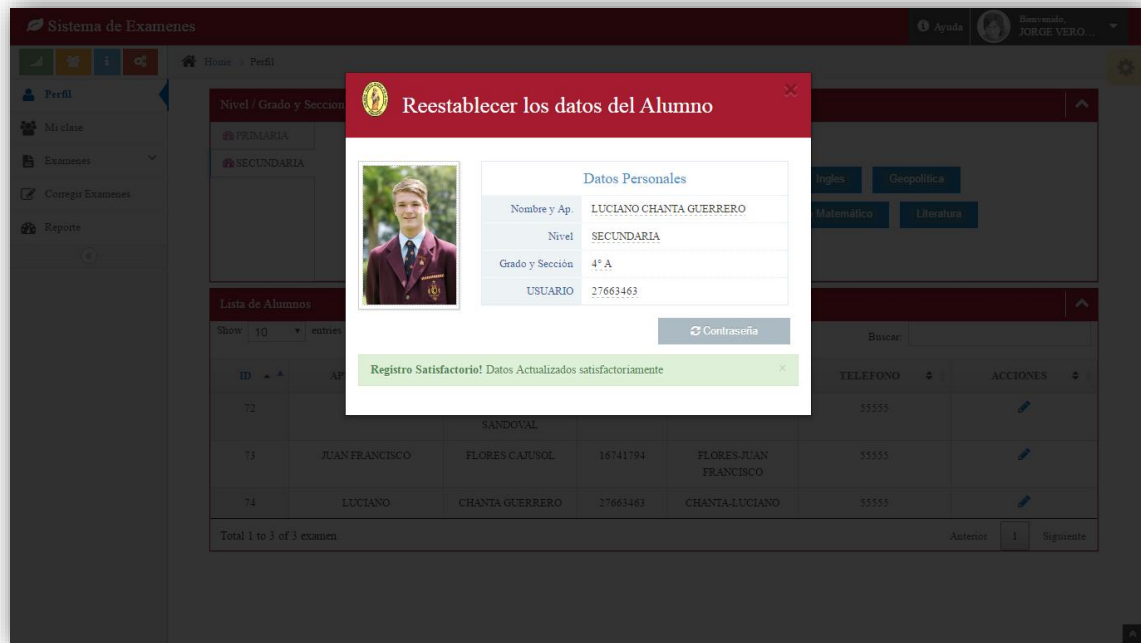
GUI 19 Personalización del espacio del usuario, Choose Skin.



GUI 20 Mecanismo de Usabilidad, Help.

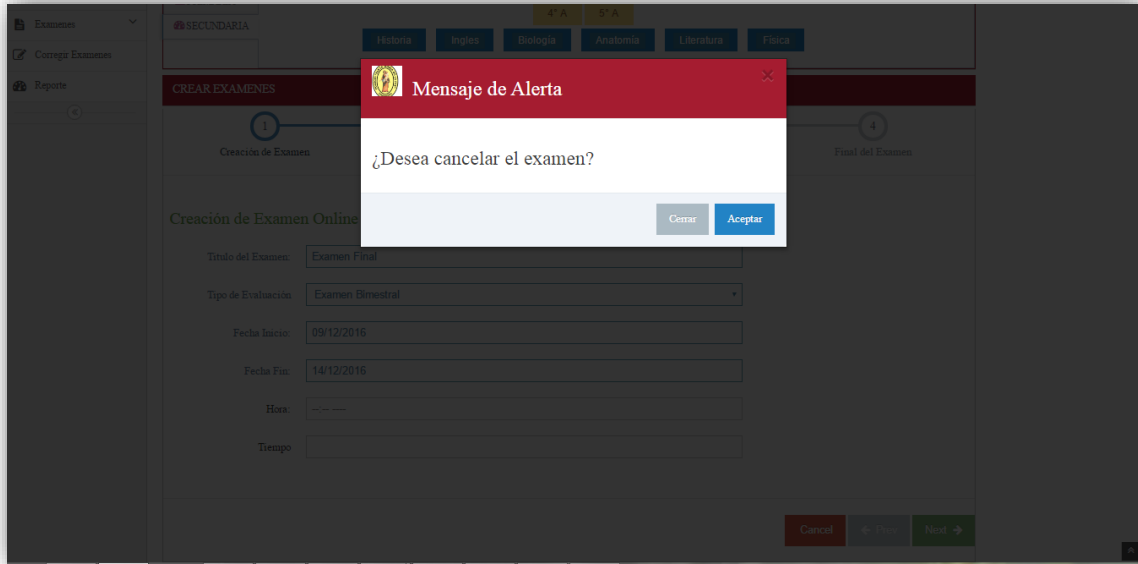


GUI 21 Mecanismo de Usabilidad, Wizard

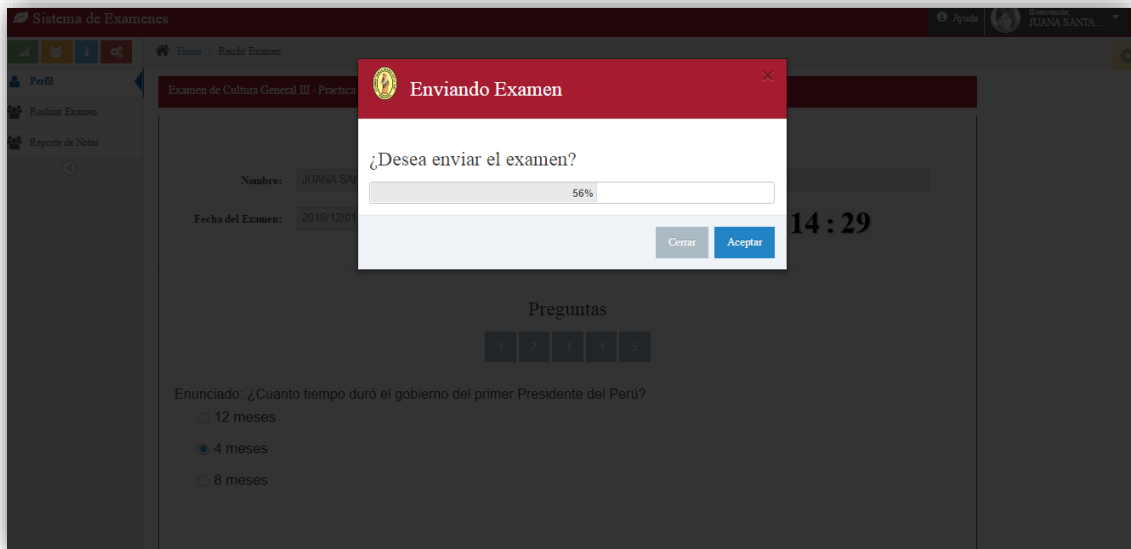


GUI 22: Mecanismo de usabilidad, Warning, confirma el estado de registro correcto.





GUI 23 Mecanismo de usabilidad Undo cancel para cancelar el examen



GUI 24 Mecanismo de usabilidad Progress Feedback para el envió del examen

En el Capítulo IV: Resultados de la investigación, se procedió a evaluar al producto software con el fin de medir la usabilidad como característica de calidad externa, bajo la ISO/IEC 9126-2.



## CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES

### 6.1 Conclusiones

Según (Ormeño Y., Panach J., Pastor O, 2012), “*La usabilidad es considerada en las últimas etapas del proceso de desarrollo de software, cuando las interfaces han sido ya diseñadas*”. En esta etapa, la inclusión de características de usabilidad puede afectar a la arquitectura del sistema, con la participación tanto de la reanudación de procesos, actividades que ya han sido concluidas, más trabajo. Para Chian Wen Too & Sa’adah Hassan (2015) la falta de atención a los requisitos de usabilidad durante la primera etapa del proceso de desarrollo de software, significa el aumento del costo de la solución de problemas de usabilidad y hace que los productos de software finales sean pobres en calidad, es por ello que esta investigación parte de la interrogante ¿Cómo capturar los requerimientos de usabilidad en etapas tempranas del desarrollo de software?, para dar solución a esta interrogante, el presente trabajo de investigación tuvo como objetivo principal la captura de los requisitos no funcionales, principalmente el de la usabilidad, evaluando las diferentes metodologías de desarrollo web que existen bajo el paradigma de Desarrollo basado en Modelos (MDD), que tratan los requisitos no funcionales desde la etapa de captura de requisitos, con la integración de Usability Elicitation Patterns (USEPs) para la mejora de la identificación de requisitos de Usabilidad.

Como punto de partida de este trabajo se procedió a la identificación de los diferentes factores de calidad de usabilidad de un producto software propuestos por los diferentes estándares como lo son la ISO/IEC 9126, ISO/IEC 9241 y la ISO/IEC 25000 para poder evaluar los productos software, en la cual se concluyó que la ISO/IEC 9126 es la más eficiente para esta investigación, dado a que es una ISO que en el Perú es aplicada para la evaluación del producto software tanto en entidades públicas como privadas. Seguido de ello se identificó y seleccionó las metodologías de desarrollo web por la cual se tuvo como

referencia a la investigación de ESCALONA y KOCH (2004), donde se hace énfasis a los tipos de requerimientos tratados por cada enfoque metodológico de desarrollo de software Web, bajo las técnicas empleadas y las fases que cubren cada enfoque en términos de su proceso de desarrollo, las técnicas aplicadas y los resultados obtenidos. Seleccionando las metodologías NDT y UWE con las cuales se desarrolló un caso de estudio aplicando las directrices USEP's en la etapa de captura y definición de requerimientos, lográndose definir requerimientos funcionales de usabilidad e integrarlas en cada uno de los artefactos generados tanto en la metodología de desarrollo web NDT y UWE.

Realizada la integración se procedió a evaluar y elegir la metodología más factible de acuerdo al tiempo que requirió desarrollar cada metodología integrando las directrices de acuerdo a su alcance de las mismas, así como también se analizó la cantidad de artefactos generados por cada una de las metodologías segregando los artefactos que integran a la usabilidad y a los artefactos que no plasman un requerimiento funcional de usabilidad. También se evaluó a las metodologías bajo el número de mecanismos de usabilidad que estas integraron en cada una de sus fases, así como el total de ellas, determinando que la metodología más eficiente es la NDT frente a los resultados recolectado de la UWE, debido a que da más énfasis a la etapa de definición de requerimientos, el tiempo que tomo el realizar la captura de requerimientos, definición de requerimientos, análisis y diseño del sistema con la integración de las directrices fue más corto frente a los resultados obtenidos de la UWE, también se concluyó que la NDT es la metodología que integro un mayor número de mecanismos de usabilidad.

Cabe resaltar que la metodología NDT trabaja con matrices de Trazabilidad, la cual relaciona los elementos generados por cada artefacto, con los elementos consecuentes del modelo generado producto de transformaciones, además de que cada elemento está alineado con los objetivos del proyectos, de tal forma que mantuvo la



consistencia e integridad de los requerimientos a través de una fase a otra.

Aplicada la evaluación con la ISO/IEC 9126-02 para medir la calidad externa de usabilidad al prototipo funcional, realizado bajo metodología de desarrollo web NDT y la integración de directrices USEP's para la captura de requerimientos de usabilidad en etapas tempranas de desarrollo de software, los resultados indicaron que se alcanzó la satisfacción del cliente. Logrando obtener:

- Que el tiempo promedio del aprendizaje del funcionamiento del sistema fue de 5.85 minutos.
- El 20 % de los usuarios que manipularon el sistema indicaron que el sistema le fue muy fácil de aprender, mientras que el 40% de encuestados indico que el sistema es fácil de aprender, y otro grupo que representa el 40% de los usuarios consideran que el sistema es tiene un grado regular de facilidad de aprender.
- Otro punto a destacar de la evaluación del software se obtuvo que el 95 % de los usuarios que utilizaron el sistema comprendieron de una manera efectiva las funcionalidades del sistema. Se pudo observar que los mensajes de error presentados en el sistema, fueron de gran utilidad para que el usuario pueda corregir los errores presentados.
- Cabe resaltar que el 88% de los usuarios que utilizaron el sistema indican que el sistema fue fácil de operar dado a que concluyeron de forma exitosa las actividades designadas.
- El 90% de los usuarios manifestó que el sistema les pareció atractivo, mientras que el 10% de los usuarios que utilizaron el sistema indicaron que este tiene un grado regular de atractividad.

## 6.2 Recomendaciones:

Se plantea a continuación algunas recomendaciones, con el fin de que estas sean aspectos a considerar para trabajos futuros que se deseen realizar en campo de la Ingeniería de software, en temas relacionados con la usabilidad.

Se recomienda establecer un manual de buenas prácticas para la definición e integración de los mecanismos de usabilidad con los artefactos propuestos a generar por las metodologías UWE y NDT, para darle más énfasis a la usabilidad en cada uno de los artefactos generados.

Se recomienda seleccionar y evaluar, el sistema más adecuado de transformación de modelos a código para la metodología UWE, de tal forma que podamos definir prototipos en la fase de análisis que ayuden al analista a tener una visión general del sistema con la inclusión de los mecanismos de usabilidad en los proyectos de desarrollo de software.

Definir métricas que, basadas en los artefactos generados, de tal forma que nos permitan determinar el nivel de usabilidad que tiene un producto software, a partir de sus modelados, para así lograr el mejor valor de usabilidad.

## REFERENCIAS

- Apache Friends. (2016). *XAMPP*. Obtenido de XAMPP:  
<https://www.apachefriends.org/index.html>
- Brandon, D. (2008). *Software Engineering for Modern Web Applications: Methodologies and Technologies*. New York: IGI Global.
- Braude, E. j. (2003). *Ingeniería del Software. Una perspectiva Orientado a Objetos*. RA-MA.
- Benigni G., Zabala S. (2013). Diseño de interfaces en los medios visitados por los venezolanos ¿Los más usados son usables?. Revisado del repositorio Digital de tesis de la Universidad Politécnica de Valencia. [http://sedici.unlp.edu.ar/bitstream/handle/10915/50691/Documento\\_completo.pdf-PDFA.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/50691/Documento_completo.pdf-PDFA.pdf?sequence=1)
- Calidad Integral* (2017.). Retrieved from ISO 9001 | Calidad Integral:  
[https://www.calidadintegral.com/iso\\_9000.php](https://www.calidadintegral.com/iso_9000.php)
- Carpi, A., & Anne E., E. (2008). *Visionlearning*. Obtenido de Comparación en la Investigación Científica:  
<http://www.visionlearning.com/es/library/Proceso-de-la-Ciencia/49/Comparaci%C3%B3n-en-la-Investigacion-Cient%C3%ADfica/152>
- Colegio de Ingenieros del Perú. (30 de Julio de 2012). *CÓDIGO DEONTOLÓGICO DEL COLEGIO DE INGENIEROS DEL PERÚ 2011*. Obtenido de Colegio de Ingenieros del Perú:  
[http://www.cip.org.pe/Cvista/publicaciones/documentos/CODIGO\\_DEONTOLOGICO\\_2011.pdf](http://www.cip.org.pe/Cvista/publicaciones/documentos/CODIGO_DEONTOLOGICO_2011.pdf)
- Condor, E. (15 de Enero de 2013). *Pezi*. Recuperado el 4 de Diciembre de 2016, de <https://prezi.com/bbv5cko3mmp1/metodologias-de-ingenieria-de-software/>
- Cortés Morales, R. (2006). *Introducción Al Análisis de Sistemas Y la Ingeniería de Software*. San José: EUNED.
- Daniel Ramos, R. N. (2015). Curso de Ingeniería de Software. En R. N. Daniel Ramos, *Curso de Ingeniería de Software* (pág. 341). Copyright.
- Del Valle Rojo S. (2012). Requerimientos no Funcionales para Aplicaciones Web. Revisado del SEDICI Repositorio Digital Institucional de la

Universidad Nacional de la Plata.

[http://sedici.unlp.edu.ar/bitstream/handle/10915/33033/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/33033/Documento_completo.pdf?sequence=1)

Del Valle Rojo S., Oliveros A. (2013). Elicitación y Especificación de Requerimientos No Funcionales para aplicaciones Web. Revisado del SEDICI Repositorio Digital Institucional de la Universidad Nacional de la Plata. [http://sedici.unlp.edu.ar/bitstream/handle/10915/33039/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/33039/Documento_completo.pdf?sequence=1)

Del Valle Rojo, S., & Oliveros, A. (2014). Elicitación y Especificación de Requerimientos No Funcionales para aplicaciones Web. *ASSE 15º Simposio Argentino de Ingeniería de Software*, 218-231. Obtenido de [https://www.researchgate.net/publication/265759472\\_Elicitacion\\_y\\_Especificac](https://www.researchgate.net/publication/265759472_Elicitacion_y_Especificac)

Escalona, J., Koch, N. (2004). Requirements Engineering For Web Applications– A Comparative Study. Rinton Press. *Journal of Web Engineering*, Vol.2, No.3, pg. 193-212. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.153.5974&rep=rep1&type=pdf>

Fernández, J. P. (2014). *Aplicación de técnicas de usabilidad y accesibilidad en el entorno cliente*. IC. Antequera, Málaga. Obtenido de <http://www.tierradelazaro.com/wp-content/uploads/2016/08/UF1843-Aplicaci%C3%B3n-de-t%C3%A9cnicas-de-usabilidad-y-accesibilidad-en-el-entor.pdf>

Ferré Grau, X. (2001). Principios Básicos de Usabilidad para Ingenieros Software. Madrid, Madrid, España. Obtenido de <http://lucio.ls.fi.upm.es/miembros/xavier/papers/usabilidad.pdf>

Garcia, X. G. (2015). *Aplicaciones de usabilidad y accesibilidad en el entorno cliente*. Parainfo:Madrid, España. Obtenido de: [https://books.google.com.pe/books?id=JgDsBgAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.pe/books?id=JgDsBgAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

Garrido Carrillo, A. (2005). *Fundamentos de programación en C++*. Madrid: Delta.

- Gomez Fuentes, M. d. (2011). Notas del curso: Analisis de Requerimientos. En M. d. Gomez Fuentes, Notas del curso: Analisis de Requerimientos. Mexico. Obtenido de la UAM Cuajimalpa: [http://web.cua.uam.mx/publicaciones/Notas\\_Analisis\\_Requerimiento.pdf](http://web.cua.uam.mx/publicaciones/Notas_Analisis_Requerimiento.pdf)
- Grupo de investigación Ingeniería Web y Testing Temprano (IWT2),. (2010). *User guide and best practices for NDT-Profile 2.X*. Retrieved from <http://iwt2.org/actividad-grupo/investigacion/resultados/ndt/aprendiendo-ndt/manuales-de-ndt-y-ejemplos/>
- Grupo de investigación Ingeniería Web y Testing Temprano (IWT2). (2016). *IWT2*. Obtenido de IWT2: <http://iwt2.org/wp-content/uploads/2015/05/09-DRS.pdf>
- H. Kaisler, S. (2005). *Software Paradigms*. New Jersey: John Wiley & Sons, Inc. Anthony Carpi, P. A. (s.f.).
- Hari Prasad N. y Ramesh G. (2015). Usability through Design and its Impact on Reusability. *International Journal of Advanced Research in Computer Science*, Volume 6, No. 6, ISSN N° 0976-5697. <http://ezproxy.concytec.gob.pe:2087/eds/pdfviewer/pdfviewer?sid=dcf3841a-f757-4eb8-a675-4d6974a4a6ae%40sessionmgr107&vid=23&hid=114>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. (2010). *Metodología de la investigación*. México D.F.: McGRAW-HILL.
- Hernández, R., & Ander. (19 de Noviembre de 2011). *Tesis de Investigacion*. Obtenido de Tipos de Investigacion: <http://tesisdeinvestig.blogspot.pe/2011/11/tipos-de-investigacion-segun-ander-egg.html>
- Ignacio Panach, J., Juristo, N., Valverde, F., & Pastor, Ó. (2015). A framework to identify primitives that represent usability within model-driven development method. "Information And Software Technology", v.58, pp. 338-354  
[http://personales.upv.es/jopana/Files/Journals/A\\_Framework\\_to\\_Identify\\_Primitives.pdf](http://personales.upv.es/jopana/Files/Journals/A_Framework_to_Identify_Primitives.pdf)
- Isela Ormeño, Y., & Ignacio Panach, J. (2013). Mapping Study about Usability Requirements Elicitation. Obtenido de

- [http://personales.upv.es/jopana/Files/Conferences/Mapping\\_Study\\_About.pdf](http://personales.upv.es/jopana/Files/Conferences/Mapping_Study_About.pdf)
- Isela Ormeño, Y., Ignacio Panach , J., & Pastor, Ó. (2012). *Elicitación de Requisitos de Usabilidad: Un estudio sistemático*.
- Iso.org. (2015). Retrieved from La International Standards Organization: <https://www.iso.org/about-us.html>
- ISO/IEC 25000. (2014). Obtenido de Portal ISO/IEC 25000: <http://iso25000.com/>
- iso9001calidad.com. (2013). *ISO 9001 calidad. Sistema de Gestion de Calidad segun la ISO 9000*. Obtenido de <http://iso9001calidad.com/definicion-de-terminos-586.html>
- iso9001calidad.com. (2013). *ISO 9001 calidad. Sistemas de Gestion de Calidad segun ISO 9000*. Obtenido de ISO 9001 calidad. Sistemas de Gestion de Calidad segun ISO 9000: <http://iso9001calidad.com/medir-satisfaccion-del-cliente-55.html>
- ITI-UPV. (2016). *Instituto Tecnológico de Informática de la Universitat Politècnica de València*. Recuperado el Septiembre de 2016, de <http://www.iti.es/es/tecnologias-y-servicios/servicios-tecnologicos-avanzados/calidad-del-software>
- Ivar, J., Grady, B., & James, R. (2000). El proceso Unificado de Desarrollo de Software. En J. Ivar, B. Grady, & R. James, *El proceso Unificado de Desarrollo de Software*. Addison Wesley.
- IWT2. (11 de 02 de 2016). *IWT2, Ingeniería Web y Testing temprano*. Obtenido de IWT2, Ingeniería Web y Testing temprano: <http://iwt2.org/actividad-grupo/investigacion/resultados/ndt/>
- La International Standards Organization. (2015). Obtenido de ISO 9000 - Quality management: [http://www.iso.org/iso/home/standards/management-standards/iso\\_9000.htm](http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm)
- La International Standards Organization. (2015). Obtenido de ISO 9000 - Quality management: [http://www.iso.org/iso/home/standards/management-standards/iso\\_9000.htm](http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm)

- Laínez Fuentes, J. R., Noriega Martínez, R., Ramos, D., & Durango, A. (2015). *Curso de Ingeniería de Software*. IT Campus Academy.
- LMU – Ludwig-Maximilians-Universität München, Institute for Informatics. (1 de Enero de 2015). *UWE – UML-based Web Engineering*. Obtenido de UWE – UML-based Web Engineering: <http://uwe.pst.ifi.lmu.de/index.html>
- Loic Martinez, F. A. (2005). Introducción a la Ingeniería del software: Modelos de Desarrollo de Programas. En F. A. Loic Martinez, *Introducción a la Ingeniería del software: Modelos de Desarrollo de Programas*. Delta.
- Lopez Echeverry, A. M., Cabrera, C., & Valencia Ayala, L. E. (2008, Septiembre 39). Introducción a la Calidad de Software. *Scientia Et Technica 2008, XIV (39)*. Retrieved from <http://www.redalyc.org/articulo.oa?id=84920503058>
- Lynch J. (Octubre 04, 2015) Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch. Revisado de <http://www.infoq.com/articles/standish-chaos-2015>
- Macconnel , S. (1996). *Desarrollo y Gestion de proyectos Informaticos*. Madrid: Macgraw-Hill.
- Maria J. Escalona ; University of Seville, Seville ; Gustavo Aragón. (2008). NDT. A Model-Driven Approach for Web Requirements. *IEEE Transactions on Software Engineering, 34*, 377 - 390. doi:10.1109/TSE.2008.27
- MARTÍNEZ PÉREZ , L. (4 de Septiembre de 2012). *Ingenieria de software*. Obtenido de Ingenieria de software: <http://softwareverde.blogspot.pe/2012/09/definicion-de-modelo.html>
- Martinez, R. (2013). *PosgreSQL-es*. Obtenido de PostgreSQL-es: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- Moraga, A., Piattini, M. G., & Calero, C. (2010). *Calidad del Producto y Proceso Software*. Ra-Ma.
- Moreno J., Marciszack M. (2014). La Usabilidad Desde La Perspectiva De La Validación de Requerimientos No Funcionales Para Aplicaciones Web. WICC 2014 XVI Workshop de Investigadores en Ciencias de la Computación. <http://conaiisi.unsl.edu.ar/ingles/2013/231-655-1-DR.pdf>.
- Moya de Madrigal, L. (2005). *Introduccion Ala Estadistica de la Salud*. San José: Editorial de la Universidad de Costa Rica.

- Navarro, C., Molina Dias, A., & Redondo, M. (2015, Febrero 13). Towards a Model for Evaluating the Usability of M-learning Systems: from a Mapping Study to an Approach. *IEEE LATIN AMERICA TRANSACTIONS*, 13(2). Retrieved from [https://www.researchgate.net/publication/273391272\\_Towards\\_a\\_Model\\_for\\_Evaluating\\_the\\_Usability\\_of\\_M-learning\\_Systems\\_from\\_a\\_Mapping\\_Study\\_to\\_an\\_Approach](https://www.researchgate.net/publication/273391272_Towards_a_Model_for_Evaluating_the_Usability_of_M-learning_Systems_from_a_Mapping_Study_to_an_Approach)
- Oracle Corporation and/or its affiliates. (01 de 05 de 2016). MySQL. Obtenido de MySQL: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- Ormeño Y., Panach J.(2013). Mapping Study about Usability Requirements Elicitation. *Advanced Information Systems Engineering Volume 7908 of the series Lecture Notes in Computer Science* pp 672-687.<http://link.springer.com/book/10.1007/978-3-642-38709-8>
- Panach Navarrete, J. I. (2010). Incorporación de Mecanismos de Usabilidad en un Entorno de Producción de Software Dirigido por Modelos. Revisado del repositorio Digital de tesis de la Universidad Politécnica de Valencia. Recuperado el Septiembre de 2015, de <https://riunet.upv.es/bitstream/handle/10251/8379/tesisUPV3285.pdf>
- Paz F. (2013). Heurísticas De Usabilidad Para Sitios Web Transaccionales. Revisado del Repositorio Digital de Tesis PUCP [http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5399/PAZ\\_FREDDY\\_HEURISTICA\\_USABILIDAD\\_SITIOS\\_WEB\\_TRANSACCIONALES.pdf?sequence=1&isAllowed=y](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5399/PAZ_FREDDY_HEURISTICA_USABILIDAD_SITIOS_WEB_TRANSACCIONALES.pdf?sequence=1&isAllowed=y)
- Pons, C., Giandini, R., & Pérez, G. (2010). *Desarrollo de software dirigido por modelos* (1º Edición ed.). Buenos Aires , Argentina: Mac Graw Hill. Obtenido de DOCPLAYER <http://docplayer.es/3391898-Desarrollo-de-software-dirigido-por-modelos-conceptos-teoricos-y-su-aplicacion-practica.html>
- Portal Iso. (2015). Obtenido de ISO 25000 Calidad del Producto Software: <http://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad>
- Pressman S., R. (2010). *Ingeniería del Software: Un enfoque practico* (7ª ed.). México, D. F., México: The McGraw-Hill Companies.



- Pressman, R., & Lowe, D. (2009). *Web Engineering. A practitioner's Approach*. New York: McGraw-Hill.
- Real Academia Española. (Octubre de 2014). *Diccionario de la Lengua Española*. Obtenido de Diccionario de la Lengua Española: <http://dle.rae.es/?id=Om9ZDVF>
- Ruiz, F. (4 de Diciembre de 2016). *Alarcos Research Group*. Obtenido de Alarcos Research Group: [http://alarcos.esi.uclm.es/per/fruiz/curs/ips/trans/ips\\_spem.pdf](http://alarcos.esi.uclm.es/per/fruiz/curs/ips/trans/ips_spem.pdf)
- Sánchez, S., Sicilia, M. Á., & Rodríguez, D. (2012). *Ingeniería del Software un enfoque desde la guía SWEBOK*. Madrid: Alfaomega.
- Sanchez, W. (2 Agosto del 2011). *La usabilidad en Ingeniería de Software: Definición y características*.
- Sanchez-Segura, M.-I., Moreno, A., & Juristo, N. (2007). Guidelines for Eliciting Usability Functionalities. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 3, 33.
- Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. DF Mexico: Prentice Hall.
- SOMMERVILLE, I. (2005). *INGENIERIA DE SOFTWARE (7º ed.)*. Madrid: Pearson Education.
- Sommerville, I. (2005). *Ingeniería del Software. 7ª Edición*. Pearson. Madrid: PEARSON EDUCACIÓN. S.A..
- Tahuiton Mora, J. (s.f.). *Heurísticas De Usabilidad Para Sitios Web Transaccionales*. Obtenido de Repositorio Digital de Tesis PUCP: [http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5399/PAZ\\_FREDDY\\_HEURISTICA\\_USABILIDAD\\_SITIOS\\_WEB\\_TRANSACCIONALES.pdf?sequence=1&isAllowed=y](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5399/PAZ_FREDDY_HEURISTICA_USABILIDAD_SITIOS_WEB_TRANSACCIONALES.pdf?sequence=1&isAllowed=y)
- Técnicas Avanzadas de Diseño Web*. (septiembre de 2009). España: Editorial Vertice. Obtenido de <https://books.google.com.pe/books?id=8-HfD0pjAg0C&pg=PA121&dq=La+usabilidad&hl=es&sa=X&ved=0ahUKEwj55OL8q9PMAhWEQCYPKHC3LApA4ChDoAQgmMAI#v=onepage&q=La%20usabilidad&f=false>
- The PHP Group. (13 de Mayo de 2016). *PHP*. Obtenido de <http://php.net/manual/es/intro-what-is.php>
- Thomas Pablo, J. (2006). Definición de un proceso de elicitación de objetivos. Revisado del Repositorio Digital Institucional de la Universidad

Nacional de la Plata, Postgrado Facultad de Informática.  
[http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Ingenieria\\_de\\_Software/Tesis/Thomas\\_Pablo.pdf](http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Ingenieria_de_Software/Tesis/Thomas_Pablo.pdf)

- Travé Gonzales, G. (1996). *Consideraciones sobre la utilización de técnicas e instrumentos de investigación educativa para la evaluación de unidades didácticas de contenido social*. Obtenido de Repositorio Institucional de la Universidad de Huelva: [http://rabida.uhu.es/dspace/bitstream/handle/10272/10654/Consideraciones\\_sobre\\_la\\_utilizacion.pdf?sequence=2](http://rabida.uhu.es/dspace/bitstream/handle/10272/10654/Consideraciones_sobre_la_utilizacion.pdf?sequence=2)
- Urvina Barrionuevo, K. R., & Gavilanes Torres, E. D. (2013). *Sistema web utilizando Software Libre para la administración y control del registro acumulativo de alumnos para el departamento de orientación y bienestar estudiantil de la Unidad Educativa Experimental "Pedro Fermín Cevallos" De La Ciudad De Ambato*. Ecuador. Retrieved from <http://repositorio.uta.edu.ec/handle/123456789/4952>
- Vilariño De Almeida, J. C. (2010). *Modelo para la selección de la metodología de desarrollo web de una aplicación según sus características funcionales*. Caracas .
- Vinca. (2011). *Normas ISO 9000*. Obtenido de Normas ISO 9000: <http://www.normas9000.com/que-es-iso-9000.html>
- Wen Too, C., & Hassan, S. (2015). *The Adoptions of USEPs in Identifying Usability Requirements in SSM-Based Framework: A Case Study*. Obtenido de Intelligent Software Methodologies, Tools and Techniques Volume 513 of the series Communications in Computer and Information Science pg. 71-82.: [http://link.springer.com/chapter/10.1007%2F978-3-319-17530-0\\_6#page-1](http://link.springer.com/chapter/10.1007%2F978-3-319-17530-0_6#page-1)

## ANEXO I



Colegio Particular  
"SANTA MARÍA DE LA PAZ"  
INICIAL -PRIMARIA - SECUNDARIA  
-Pimentel-



UNIVERSIDAD  
SEÑOR DE SIPÁN

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	Huaman Villanueva B. Ortiz Matute M.	Huaman Villanueva B.	Melgar Neciosup Alex	09-08-16	Versión Original

### ACTA DE ACEPTACIÓN DEL PROYECTO

NOMBRE DEL PROYECTO	SIGLAS DEL PROYECTO
Desarrollo de una aplicación Web Exámenes Online para la Institución Educativa Particular "Santa María de la Paz"- Pimentel.	<b>DAWEOIESMPP</b>
NOMBRE DEL CLIENTE O SPONSOR	
DIR. YOVANA LOAYZA NECIOSUP	

### DECLARACIÓN DE LA ACEPTACIÓN FORMAL

Por medio de la presente acta se hace una constancia de la aceptación del proyecto **"Desarrollo de una aplicación Web de Exámenes Online para la Institución Educativa Particular Santa María de la Paz, Pimentel"** a cargo de HUAMAN VILLANUEVA BRAYAN YERAK con N° DNI 73251931 y ORTIZ MATUTE MEGAN ASHLEY con N° DNI 72744269, estudiantes de la Escuela de Ingeniería de Sistemas, de la Universidad Señor de Sipán, Iniciado el 6 de Agosto del 2016 y culminando el 15 de Diciembre del 2016, según lo acordado en la reunión de presentación del proyecto.

Sin más preámbulos yo, YOVANA LOAYZA NECIOSUP, directora de la institución educativa, doy la autorización para que se lleve a cabo el proyecto de **"Desarrollo de una aplicación Web de Exámenes Online para la Institución Educativa Particular Santa María de la Paz, Pimentel"**

### OBSERVACIONES ADICIONALES

N/A





Colegio Particular  
"SANTA MARÍA DE LA PAZ"  
INICIAL - PRIMARIA - SECUNDARIA  
-Pimentel-



UNIVERSIDAD  
SEÑOR DE SIPÁN

ACEPTADO POR:	
NOMBRE DEL CLIENTE, SPONSOR, REPRESENTANTES DEL PROYECTO U OTRO FUNCIONARIO	FIRMA
DIR. YOVANA LOAYZA NECIOSUP	
ALEX MELGAR NECIOSUP	
HUAMAN VILLANUEVA BRAYAN YERAK	
ORTIZ MATUTE MEGAN ASHLEY	
DISTRIBUIDO Y ACEPTADO	
REPRESENTANTE DEL PROYECTO	FECHA
HUAMAN VILLANUEVA BRAYAN YERAK	09-08-16



## ANEXO II



Colegio Particular  
"SANTA MARÍA DE LA PAZ"  
INICIAL - PRIMARIA - SECUNDARIA  
-Pimentel-



UNIVERSIDAD SEÑOR DE SIPÁN

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
2.0	Ortiz Matute M., Huaman Villanueva B.	Ortiz Matute M.	Querevalú Paiva Jorge	09/08/2016	Versión Original

NOMBRE DEL PROYECTO	SIGLAS DEL PROYECTO
Desarrollo de una aplicación Web Exámenes Online para la Institución Educativa Particular "Santa María de la Paz"- Pimentel.	<b>DAWEOIESMPP</b>

## DESCRIPCION DE LOS REQUERIMIENTOS

REQUERIMIENTOS FUNCIONAL DOCUMENTADO	
<b>Requerimiento A: Logueo</b>	<b>RA</b>
<ol style="list-style-type: none"> <li>1. El docente se identificará en el sistema con el número de su DNI y una contraseña.</li> <li>2. El alumno se identificará en el sistema con el número de su DNI y una contraseña.</li> <li>3. Las contraseñas serán otorgadas en secretaría, las cuales pueden ser modificadas.</li> <li>4. Las contraseñas son otorgadas a los alumnos por el tutor del grado y sección.</li> <li>5. Para recuperación de contraseñas, el alumno solicita su contraseña al tutor del aula.</li> <li>6. Las contraseñas pueden incluir letras, números y símbolos.</li> </ol>	
<b>Requerimiento B: sistema en General</b>	<b>RB</b>
<ol style="list-style-type: none"> <li>1. Alerta de conexión a internet.</li> <li>2. Alerta de conexión con el servidor.</li> <li>3. Alertas para los exceptions de los try catch.</li> </ol>	
<b>Requerimiento C: Crear Exámenes</b>	<b>RC</b>
<p>Una vez que el usuario se ha autenticado en el sistema, este podrá crear evaluaciones, para ello debe: elegir el grado y sección el cual tiene a cargo, seguido de seleccionar la asignatura a evaluar. Seleccionado la asignatura a evaluar, da clic en el menú de crear Exámenes, para ello deberá seguir los siguientes pasos.</p> <ol style="list-style-type: none"> <li>1. <b>Programación del examen</b> <ol style="list-style-type: none"> <li>1.1. Ingresar un título para evaluación</li> <li>1.2. Selecciona el tipo de evaluación</li> <li>1.3. seleccionar la fecha en la que se llevará a cabo el examen.</li> <li>1.4. Ingresar la hora de inicio y la hora de fin de la evaluación.</li> <li>1.5. Cantidad de preguntas.</li> <li>1.6. Habilitar o deshabilitar el modo random para asignación de preguntas</li> <li>1.7. Habilitar la opción de publicación automática de notas.</li> </ol> </li> <li>2. <b>Registro de preguntas</b> <ol style="list-style-type: none"> <li>2.1. Seleccionar el tipo de pregunta</li> <li>2.2. Ingresar el enunciado de la pregunta</li> <li>2.3. Asignar la capacidad a la que pertenece dicha pregunta, opcional</li> </ol> </li> </ol>	





Colegio Particular  
"SANTA MARÍA DE LA PAZ"  
INICIAL - PRIMARIA - SECUNDARIA  
-Pimentel-



UNIVERSIDAD  
SEÑOR DE SIPÁN

- 2.4. Asignar el puntaje que tendrá la pregunta
- 2.5. Guardar las preguntas
- 2.6. Vista previa de las preguntas registradas
- 3. Programación de la vista de resultados**
  - 3.1. El docente programa la fecha de visualización de la resolución de la evaluación.
  - 3.2. El deberá seleccionar el tipo de calificativo a mostrar (0 a 20 vigesimal) o por calificación por letras.
- 4. Guardar Examen.**

<b>Requerimiento D: Visualización de Reportes</b>	<b>RD</b>
<ol style="list-style-type: none"> <li>1. El docente puede visualizar el reporte de logros por cada alumno</li> <li>2. Docente puede visualizar el reporte de logros de la clase por cada examen</li> <li>3. El docente puede visualizar el progreso de cada sección en un determinado curso</li> <li>4. Los Reportes dados a los docentes deben de ser exportados a Excel. (por cada reporte)</li> <li>5. El docente puede visualizar el reporte de logros por cada alumno ante una evaluación considerando, preguntas correctas, preguntas erróneas, preguntas en blanco, así como la nota obtenida. (dentro de las opciones de arriba)</li> <li>6. El docente puede acceder a una vista previa del examen realizado por el alumno. (dentro de las opciones de arriba)</li> <li>7. El docente puede visualizar el reporte de logros obtenidos la clase en un examen en específico, así como el progreso general de la clase teniendo en cuenta, número de preguntas respondidas correctamente, número de preguntas incorrectas, número de preguntas en blanco y promedio general. (Dentro de las opciones de arriba).</li> <li>8. Los reportes en caso de usar colores para mostrar el estado de un alumno o la nota lograda, es necesario manejar una leyenda de colores</li> </ol>	
<b>Requerimiento E: Corrección de exámenes</b>	<b>RE</b>
<ol style="list-style-type: none"> <li>1. El usuario visualiza los exámenes a corregir en una especie de bandeja, donde puede visualizar la resolución de la evaluación por cada alumno.                     <ol style="list-style-type: none"> <li>1.1. Al abrir el examen del alumno se le carga las preguntas con su determina respuesta, la cual el docente podrá otorgar un calificativo a las preguntas que son de tipo argumentativo.</li> <li>1.2. Finalizado la corrección de la evaluación confirma el guardar la evaluación.</li> <li>1.3. Una vez corregido todos los exámenes este podría dar por finalizado o cerrado la evaluación.</li> </ol> </li> <li>2. El docente podrá habilitar y deshabilitar el publicar las respuestas correctas de la evaluación.</li> </ol>	
<b>Requerimiento F: Publicación de notas</b>	<b>RF</b>
<ol style="list-style-type: none"> <li>1. El docente visualiza todos los exámenes ya corregidos y sus estados de publicación, en caso de que no esté habilitado la opción de publicación de notas este podrá programar la fecha publicación de las notas.</li> </ol>	
<b>Requerimiento G: Dar Exámenes</b>	<b>RG</b>
<p>Una vez que el usuario se ha autenticado en el sistema, este podrá rendir evaluaciones, para ello debe: seleccionar la asignatura, seleccionar entre los exámenes habilitados el examen a realizar.</p> <ol style="list-style-type: none"> <li>1. El alumno podrá visualizar el tiempo designado para la prueba, así como un cronómetro que le indicara el tiempo que le queda para finalizar la prueba.</li> </ol>	





Colegio Particular  
 "SANTA MARÍA DE LA PAZ"  
 INICIAL - PRIMARIA - SECUNDARIA  
 -Pimentel-



UNIVERSIDAD SEÑOR DE SIPÁN

**2. Resolver el examen**

Para dar respuesta al examen este podrá navegar entre las preguntas y seleccionar la pregunta a desarrollar, si la pregunta es de tipo:

- 2.1. Opción única, debe seleccionar entre las alternativas de respuesta, una única opción
- 2.2. Opción múltiple, debe seleccionar entre las alternativas de respuesta, las opciones que respondan o guarden relación a la pregunta
- 2.3. Juicio Crítico, el alumno debe de ingresar su opinión o respuesta argumentativa.
- 2.4. Relacionar.

**3. Enviar el examen**

- 3.1. El alumno selecciona el botón de enviar examen.
- 3.2. Una vez enviado el examen el alumno solo podrá visualizar la evaluación con las respuestas dadas.
- 3.3. Para los exámenes que tengan preguntas de juicio crítico, estos deben ser corregidos, evaluados por los docentes, para ello se deben de cargar en una especie de bandeja en espera de ser calificados.
- 3.4. El calificativo que le mostrará al alumno dependerá a la configuración dada por el docente.

**Requerimiento H: Visualización de Reportes**

**RH**

1. El alumno puede visualizar su progreso por curso
2. El alumno puede visualizar un reporte de los resultados obtenidos por cada examen
3. Una vez rendido el examen el alumno pueda visualizar sus respuestas y pueda descargar en pdf.
4. Los reportes en caso de usar colores para mostrar el estado de un alumno o la nota lograda, es necesario manejar una leyenda de colores.

**REQUERIMIENTOS NO FUNCIONAL DOCUMENTADO**

**RNFA**

- Que el sistema sea tolerante a fallos.
- Capacidad de recuperación de datos.
- Producto web se adapte a los distintos navegadores Web.
- Dar soporte a todos los usuarios en horarios de alta concurrencia.
- Que sea fácil de usar por todos los usuario.
- Las interfaces de la aplicación deben ser amigables e intuitivas
- La aplicación debe de proporcionar tiempos de respuesta rápidos
- La aplicación debe de ser fácil de analizar y modificar para corregir posibles fallas
- La aplicación debe de mantener los datos de forma segura y protegida.

**ACEPTADO POR:**

NOMBRE DEL CLIENTE, SPONSOR U OTRO FUNCIONARIO	FECHA Y HORA	FIRMA
DOCENTE DE LA I.E.P. MELGAR NECIOSUP ALEX	12/09/16 2:30 P.M.	
DOCENTE DE LA I.E.P. QUEREVALÚ PAIVA JORGE	12/09/16 02:30 PM	



### ANEXO III

- Los requerimientos no funcionales como requerimientos de calidad según el uso de cada metodología web orientado bajo el paradigma MDD

	Atributos de calidad ISO/IEC 9126-1							Otros			
	Usabilidad	Performance (1)	Seguridad (2)	Confibilidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Reutilización	Fiabilidad	Disponibilidad	Escalabilidad
WSDM	✓		✓		✓						
HFBM	✓	✓						✓			
UWE	✓		✓								
WebML	✓	✓	✓		✓					✓	✓
NDT	✓					✓			✓		
DDDP	✓	✓	✓				✓				

- (1) Subcaracterística de la Eficiencia
- (2) Subcaracterística de la Funcionalidad
- (3) Subcaracterística de la Portabilidad

- Alcance de cada una de las metodologías en la etapa de requerimientos no funcionales.

	Restricciones										
	Restricciones de proceso de desarrollo				Otras						
	Alcance del proceso		Detalle de proceso		Lenguajes		Técnicas				
	Parcial	Completo	Parcial	Completo	Lenguajes de modelado	Lenguaje de implementación	Representación interna de datos	Control de versiones,	Protocolos de interface	Comunicación	De ambiente de desarrollo
WSDM		✓	✓			✓					
HFBM		✓	✓		✓						
UWE		✓	✓		✓						
WebML		✓		✓	✓	✓					
NDT	✓			✓	✓					✓	✓
DDDP		✓	✓		✓	✓	✓	✓	✓		

TABLA 5.2: Requerimientos No Funcionales – Restricciones y las metodologías Web.

- Requerimientos identificados por cada una de las metodologías





	Req. Datos	Req. Interfaz usuario	Req. Navegacionales	Req. Personalización	Req. Transaccionales	Req. No Funcionales
VSDM	✓			✓		✓
OHDM	✓	✓			✓	
INA	✓	✓	✓		✓	
FPM	✓	✓	✓			✓
OHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
V2000			✓	✓	✓	
WebML	✓	✓		✓		✓
IDT	✓	✓	✓	✓	✓	✓
DDP	✓	✓	✓	✓	✓	✓

TABLA 5.3: Requerimientos identificados por cada metodología

4. Requerimientos de calidad identificado en cada una de las metodologías

	Atributos de calidad						Restricciones				
	Atributos de calidad ISO/IEC 9126-1						Restricciones de proceso de desarrollo				
	Atributos de calidad ISO/IEC 9126-1						Alcance del proceso		Detalle de proceso		
	Usabilidad	Performance (1)	Seguridad (2)	Confibilidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Parcial	Completo	Parcial	Completo
WSDM	✓		✓		✓				✓	✓	
HFPM	✓	✓							✓	✓	
UWE	✓		✓						✓	✓	
WebML	✓	✓	✓		✓				✓		✓
NDT	✓					✓		✓			✓
DDDP	✓	✓	✓			✓			✓	✓	

(1) Subcaracterística de la Eficiencia  
 (2) Subcaracterística de la Funcionalidad  
 (3) Subcaracterística de la Portabilidad

5. Técnicas de Ingeniería de requerimientos y herramientas CASE aplicadas a los requerimiento son funcionales de acuerdo a cada una de las metodologías.



## ANEXO IV

Las referentes tablas muestran los atributos a considerar de las directrices Patrones de elicitación de usabilidad (Usability Elicitation Patterns, USEPs) basada en la investigación de (Juristo N., Moreno A., Sanchez M., 2007), con el propósito de poder capturar los requerimientos de usabilidad en etapas tempranas de desarrollo de software.

Tabla 42: Una lista preliminar de características de usabilidad funcionales

Usability features with design implications	Description	Usability benefits
<b>Feedback</b>	Keep the user informed of what is going on inside the system	Nielsen[41], Constantine[14], Shneiderman[48], Hix[23]
<b>Undo cancel</b>	Allow the user to cancel complete or ongoing operations	Nielsen[41], Constantine[14], Shneiderman[48]
<b>User input errors preventions/correction</b>	To improve data input for users and software corrections as soon as possible	Constantine[14], Shneiderman[48], Hix[23]
<b>Wizard</b>	To help users to do tasks involving more than one step	Constantine[14]
<b>User profile</b>	To allow the user to tailor the application to user features.	Hix[23]
<b>Help</b>	To provide useful help for users on how to do tasks	Nielsen[41], Constantine[14],
<b>Command aggregation</b>	To help the user to create commands to execute more	Nielsen[41], Constantine[14],



	than one task at a time	Hix[23]
<b>Shortcuts (Key and tasks)</b>	To allow users to activate a task with one quick gesture	Nielsen[41], Constantine[14], Shneiderman[48], Hix[23]
<b>Reuse information</b>	To allow the user to easily move data from one part of a system to another	Constantine[14],

Tabla 43 Los mecanismos de usabilidad para los que se han desarrollado Usabilidad Obtención y Guías de Especificaciones

Usability Feature	Usability Mechanism	HCI Authors' Label	Goal
<b>Feedback</b>	System Status	Modeless Feedback Area [15] Status Display[51]	To inform users about the internal status of the system
	Interaction	Interaction Feedback[53] Modeless Feedback Area [15] Let Users Know What is Going on[6]	To inform users that the system has registered a user interaction, i.e., that the system has heard users.
	Warning	Think Twice [53] Warning [54]	To inform users of any action with important consequences
	Long Action Feedback	Progress Indicator [51][52] Show Computer is Thinking[53] Time to Do Something Else[53] Progress[54] Modeless Feedback Area [15] Let Users Know What is Going On [6]	To inform users that the system is processing an action that will take some time to complete.
<b>Undo Cancel</b>	Global Undo	Multi-Level Undo [51][52] Undo[54]	To undo system actions at several



		Global Undo[31] Allow Undo[53] Go Back One Step[51]	levels
	Object – Specific Undo	Object-Specific Undo[31]	To undo several actions on an object
	Abort Operation	Go Back One Step[51] Emergency Exit[53] Cancellability[52]	To cancel the execution of an action or the whole application
	Go Back	Go Back to a Safe Place [51] Go Back One Step[51]	To go back to a particular state in a command executions sequence
<b>User Input Error Prevention/ Correction</b>	Structure Text Entry	Forms, Structure Text Entry [51] Structure Format[52] Structured Text Entry [53]	To help prevent the user from making data input errors
<b>Wizard</b>	Step-by-Step Execution	Step –by- Step[51] Wizard[52][54]	To help users to do tasks that require different steps with user input and correct such input
<b>User Profile</b>	Preferences	User Preferences[51] Preferences[54]	To record each user's options for using the system interface.
	Personal Object Space	Personal Object Space[51]	To record each user's options for using the system interface
	Favorites	Favorites[54] Bookmarks[51]	To record certain places of interest for the user
<b>Help</b>	Multilevel Help	Multilevel Help	To provide different help level for different users
<b>Command Aggregation</b>	Command Aggregation	Composed Command[51] Macros[52]	To express possible actions to be taken with the software thought commands that can be built from smaller parts.



## ANEXO V

NOMBRE DEL PROYECTO	SIGLAS DEL PROYECTO
Desarrollo de una aplicación Web Exámenes Online para la Institución Educativa Particular “Santa María de la Paz”- Pimentel.	<b>DAWEOIESMPP</b>

ENCUESTA					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	Ortiz Matute M., Huaman Villanueva B.	Ortiz Matute M.	Querevalú Paiva Jorge		Versión Original

INDICACIONES	
<p>En encuesta dirigida para profesores y alumnos de la Institución Educativa Particular “Santa María de la Paz”.</p> <p>Responder la encuesta de forma anónima y personal, marcando con una <i>Xo</i> en cerrando con un círculo, su respuesta.</p>	
TIPO DE USUARIO	
Docente	Alumno
<p>La presente encuesta es una modelo de encuesta SUS, usado extensivamente en evaluaciones de proyectos en Digital Equipment Co. Ltd (sistemas de oficina, dirección de sistemas, herramientas técnicas y sistemas de hardware), resultando como simple y fiable. La escala SUS es una escala de estilo Likert que genera un único número, representando una medida compuesta de la usabilidad del sistema global sometido a estudio.</p>	

Escala para la Usabilidad de los Sistemas (System Usability Scale). Sustituyo "sistema" por "website"				
1. Creo que me gustará visitar con frecuencia este website				
En completo				Completamente de acuerdo
Desacuerdo				
1	2	3	4	5
2. Encontré el websi te innecesariamente complejo				
En completo				Completamente de acuerdo
Desacuerdo				
1	2	3	4	5
3. Pensé que era fácil utilizar el website				
En completo				Completamente de acuerdo
Desacuerdo				
1	2	3	4	5
4. Creo que necesit aría del apoyo de un experto para recorrer el website				
En completo				Completamente de acuerdo
Desacuerdo				
1	2	3	4	5



5. Encontré las diversas posibilidades del website bastante bien integradas				
En completo				Completamente de acuerdo
Desacuerdo				
1	2	3	4	5
6. Pensé que había demasiada inconsistencia en el website				
En completo				Completamente de acuerdo
Desacuerdo				
1	2	3	4	5
7. Imagino que la mayoría de las personas aprenderían muy rápidamente a utilizar el website				
En completo				Completamente de acuerdo
Desacuerdo				
1	2	3	4	5
8. Encontré el website muy grande al recorrerlo				
En completo				Completamente de acuerdo
Desacuerdo				
1	2	3	4	5
9. Me sentí muy con fiado en el manejo del website				
En completo				Completamente de acuerdo
desacuerdo				
1	2	3	4	5
10. Necesito aprender muchas cosas antes de manejarlo en el website				
En completo				Completamente de acuerdo
desacuerdo				
1	2	3	4	5

