



UNA UNIVERSIDAD CON ALMA DE GUERRERO

ANÁLISIS Y EVALUACIÓN DE LAS ARQUITECTURAS REST Y SOAP,
PARA EL DESARROLLO DE SERVICIOS WEB APLICADOS
AL ERP "ADRISERP" Y SU VERSIÓN MÓVIL EN ANDROID

Tesis para optar por el Título de Ingeniero
de Sistemas, que presenta el bachiller.

AUTOR

LUCIANA ESTEFANIA BURGOS SUERO

CHICLAYO - PERÚ 2017



PRESENTACIÓN DE LA APROBACIÓN

“Análisis y evaluación de las arquitecturas REST y SOAP para el desarrollo de servicios web aplicados al ERP "AdrisERP" y su versión móvil en Android.”

Aprobación de la tesis

Bach. Burgos Suero Luciana Estefanía.

Mg. Ing. Bravo Ruiz Jaime Arturo
Presidente del Jurado de Tesis

Ing. Mejía Cabrera, Heber Iván
Secretario del Jurado de Tesis

Mg. Ing. Aquino Trujillo Yuri
Vocal de Jurado de Tesis

DEDICATORIA

Este trabajo está dedicado a mi familia, por apoyarme en el tiempo que ha durado el desarrollo del mismo, se lo dedico a ellos por la paciencia que me tuvieron; a mi madre por el apoyo y motivación para dar inicio a esta lucha de desarrollarme como profesional, a mi esposo por darme la seguridad, fortaleza y herramientas para poder continuar en este camino.

Y sobre todo a DIOS por la oportunidad de cumplir con un objetivo más en mi vida y porque sin él no lo hubiera podido lograr.

AGRADECIMIENTO

Agradezco a Dios por darme la fuerza y motivos para alcanzar mis objetivos, a mi familia por su gran apoyo, amor, y por guiarme en cada momento.

Y le agradezco a mi esposo por todo el apoyo, el compromiso y todo la paciencia que tubo para poder guiarme, acompañarme en este proceso, y por estar ahí siempre para verme superarme.

INDICE

PRESENTACIÓN DE LA APROBACIÓN	2
DEDICATORIA.....	3
AGRADECIMIENTO	4
INDICE	5
RESUMEN.....	10
ABSTRACT	11
INTRODUCCIÓN	12
CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN	13
1.1. Situación problemática.....	13
1.2. Formulación del problema.....	15
1.3. Delimitación de la investigación.....	15
1.4. Justificación e importancia de la Investigación	16
1.4.1. Justificación Científica	16
1.4.2. Justificación Económica	16
1.4.3. Justificación Social	17
1.5. Limitaciones de la Investigación	17
1.6. Objetivos	18
1.6.1. Objetivo general.....	18
1.6.2. Objetivos específicos	18
CAPÍTULO II: MARCO TEÓRICO	19
2.1. Antecedentes de Estudios.....	19
2.1.1. Nivel Internacional.....	19
2.1.2. Nivel Nacional	21
2.1.3. Nivel Regional.....	24
2.2. Estado del arte.....	24
2.2.1. Plataforma web	25
2.2.1.1. Web 2.0.....	25
2.2.1.2. Servicios Web.....	26
2.2.1.3. Arquitecturas de un sistema	27
2.3. Bases teórico científicas.....	29
2.3.1. Aplicaciones web	29
2.3.2. Web 2.0	30
2.3.3. Sistema Operativo Android.....	31
2.3.4. Arquitectura SOAP y REST	32
2.3.4.1. SOAP.....	32



2.3.4.2. REST.....	34
2.3.5. Herramienta de Insomnia.....	35
2.3.6. Herramienta de Valid WsdL.....	35
2.4. Definición de términos básicos.....	36
CAPÍTULO III: MARCO METODOLÓGICO	40
3.1. Tipo y diseño de investigación.....	40
3.1.1. Tipo de investigación.....	40
3.1.2. Diseño de la investigación.....	40
3.2. Población y muestra.....	41
3.2.1. Población.....	41
3.2.2. Muestra.....	42
3.3. Hipótesis.....	42
3.4. Operacionalización	42
3.4.1. Variable independiente.....	42
3.4.2. Variable dependiente.....	44
3.5. Métodos, técnicas e instrumentos de recolección de datos	45
3.5.1. Métodos de recolección de datos	45
3.5.2. Técnicas	45
3.5.3. Instrumentos de recolección de datos	46
3.6. Procedimiento para la recolección de datos.....	47
3.7. Análisis estadístico e interpretación de datos.....	48
3.7.1. Herramienta.....	48
3.7.2. Medida de datos estadísticos	48
3.8. Criterios éticos.....	49
3.9. Criterios de rigor científico.....	50
CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS	51
4.1. Resultados en tablas y gráficos.....	51
4.1.1. Selección de arquitecturas a utilizar.....	51
4.1.2. Selección de herramientas: medición en rendimiento y escalabilidad	52
4.1.3. Resultados de las mediciones en rendimiento y escalabilidad	53
4.1.3.1. Resultados de las mediciones de Rendimiento	53
4.1.3.2. Resultados de las mediciones de Escalabilidad	55
4.1.3.2.1. Arquitecturas SOAP.....	55
4.1.3.2.2. Arquitectura REST.....	58
4.1.3.2.3. Promedio de medición: características de calidad de las arquitecturas.....	61
4.1.4. Curva De Aprendizaje	62
4.1.4.1. Promedio para la curva de aprendizaje de acuerdo a los expertos:.....	67
4.1.5. Tiempo del desarrollo.....	68
4.1.5.1. Promedio para el tiempo de desarrollo de las arquitecturas por cada experto	73
4.1.6. Evaluación económica de la propuesta.....	74
4.1.6.1. Costos	74
4.1.6.1.1. Costos de la inversion.....	74
4.1.6.1.2. Costos de operacion.....	75
4.1.6.2. Beneficios.....	76



4.1.6.2.1. Beneficios Tangibles.....	76
4.1.6.2.2. Beneficios Intangibles.....	76
4.1.6.3. Valor presente neto	77
4.2. Contrastación de hipótesis	78
4.3. Discusión de resultados	78
4.3.1. Selección de arquitecturas	78
4.3.2. Selección de herramientas.....	79
4.3.2.1. Insomnia.....	79
4.3.2.2. Valid WDSL.....	79
4.3.3. Análisis de Rendimiento y Escalabilidad	80
4.3.3.1. Rendimiento	80
4.3.3.2. Escalabilidad	81
4.3.4. Evaluación económica de la propuesta.....	82
4.3.5. Curva De Aprendizaje	83
4.3.5.1. Arquitectura SOAP	83
4.3.5.2. Arquitectura REST.....	83
4.3.6. Tiempo Desarrollo.....	84
CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN	85
5.1. Documentación.....	85
5.1.1. Servicios de la Arquitectura REST.....	85
5.1.1.1. Código de los Servicios	85
5.1.1.1.1. Login	85
5.1.1.1.2. Listar Clientes.....	86
5.1.1.1.3. Buscar Clientes	86
5.1.1.1.4. Registrar Venta.....	87
5.1.1. Servicios de la Arquitectura SOAP	88
5.1.1.1. Código de los Servicios	88
5.1.2.1.1. Login.....	88
5.1.2.1.2. Listar Clientes	89
5.1.2.1.3. Buscar Clientes.....	90
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES.....	91
6.1. Conclusiones.....	91
6.1.1. Primera conclusión.....	91
6.1.2. Segunda conclusión.....	91
6.1.3. Tercera conclusión.....	91
6.1.4. Cuarta conclusión	91
6.2. Recomendaciones	92
6.2.1. Primera recomendación	92
6.2.2. Segunda recomendación.....	92
6.2.3. Tercera recomendación.....	92
BIBLIOGRAFÍA	93
ANEXOS.....	96
Anexo 01.- Matriz de consistencia	96



Anexo 02.- Instrumento de recolección de datos	97
Anexo 03: Estructura de la Propuesta de Solución	98
Anexo 04: Sistematización de rendimiento y escalabilidad	99
Anexo 05: Matriz de consistencia de información	100
Anexo 06: Carta de compromiso de los expertos	101
Anexo 07: Tabla de medición por experto – Variable independiente	105
Anexo 08: Juicio de experto - Variable dependiente	109
Anexo 09: Resultados de pruebas de rendimiento de arquitecturas REST y SOAP... ..	116
Anexo 10: Consumo del servicio desde app en Android con la arquitectura más óptima “REST”	121

INDICE DE ILUSTRACIONES

Ilustración 1. Web service “Descripción Grafica del funcionamiento de las arquitecturas” ..	25
Ilustración 2. Interacción entre un conjunto de Servicios Web Fuente: Arquitectura de servicios web (ws) / http://www.mad.es/	29
Ilustración 3. Aplicaciones web 1.0 y 2.0	31
Ilustración 4. Gráfico del Puntaje Promedio de la curva de aprendizaje.	68
Ilustración 6. Promedio del tiempo de desarrollo – Grafica	73

INDICE DE TABLAS

Tabla 1. Operacionalización de variable independiente.	43
Tabla 2. Operacionalización, variable dependiente.....	44
Tabla 3. Selección de la arquitectura SOAP.....	51
Tabla 4. Selección de la arquitectura REST	52
Tabla 5. Selección de Herramientas para medir tiempos.....	53
Tabla 6. Mediciones en rendimiento de las arquitecturas SOAP Y REST	53
Tabla 7 Medición de características de la arquitectura SOAP mediante la ISO/IEC 9126.....	56
Tabla 8 Medición de características de la arquitectura Rest mediante la ISO/IEC 9126	59
Tabla 9. Promedio de medición de características de calidad las arquitecturas	61
Tabla 10. Curva de Aprendizaje - Experto 1	63
Tabla 11. Curva de Aprendizaje - Experto 2.....	64
Tabla 12. Curva de Aprendizaje - Experto 3	65
Tabla 13. Curva de Aprendizaje - Experto 4	66
Tabla 14. Promedio de la curva de aprendizaje de los expertos.....	67
Tabla 15. Tiempos de desarrollo – Experto 1.	69
Tabla 16. Tiempos de desarrollo - Experto 2.....	70
Tabla 17. Tiempos de desarrollo - Experto 3.....	71
Tabla 18. Tiempos de desarrollo - Experto 4.....	72
Tabla 19. Promedio para el tiempo de desarrollo	73
Tabla 20. Inversión de Recurso Humano.....	74



Tabla 21. Inversión de Útiles de escritorio.....	74
Tabla 22. Inversión de Hardware.....	75
Tabla 23. Recursos Humanos	75
Tabla 24. Inversión de la depreciación.....	75
Tabla 25. Inversión de los materiales de escritorio	75
Tabla 26. Matriz de consistencia	96
Tabla 27. Instrumento de recolección de datos.....	97
Tabla 28. Datos Estructura de la Propuesta de Solución.....	98
Tabla 29. Tabla del Experto 01	105
Tabla 30. Tabla del Experto 02	106
Tabla 31. Tabla del Experto 03	107
Tabla 32. Tabla del Experto 04	108
Tabla 33. Planilla de la arquitectura SOAP del Experto 01	109
Tabla 34. Planilla de la arquitectura SOAP del Experto 02	110
Tabla 35. Planilla de la arquitectura SOAP del Experto 03	111
Tabla 36. Planilla de la arquitectura SOAP del Experto 03	112
Tabla 37. Planilla de la arquitectura REST del Experto 03.....	113
Tabla 38. Planilla de la arquitectura SOAP del Experto 04	114
Tabla 39. Planilla de la arquitectura REST del Experto 04.....	115

RESUMEN

La presente investigación pretende realizar el análisis y evaluación de dos arquitecturas, REST y SOAP, para el desarrollo de servicios web del módulo ADRISSMS del sistema ERP “AdrisERP”.

La problemática responde al hecho de que hoy en día existen diferentes arquitecturas para realizar el desarrollo de los servicios web, lo que hace que los programadores tengan dudas respecto a cuál de ellas es la mejor en ciertos puntos. Se debe tener en cuenta también que cada vez es más frecuente que las personas tengan uno o varios dispositivos móviles como Smartphones y Tablets. Esto produce que las personas tengan acceso a internet desde distintos aparatos electrónicos como PC, Smartphones, etc. (Fernández, VoCSDroid, 2012). Por lo cual la propuesta es realizar dos servicios web, una con la arquitectura REST y la otra con la arquitectura SOAP.

Para ello se realizará el análisis de las diferentes arquitecturas, SOAP y REST, para encontrar cuál de ellas es la mejor en el caso de rendimiento y escalabilidad, y así puedan utilizar las empresas o los desarrolladores de software, sea el caso de la necesidad que se tengan para elaborar los sistemas, además de optimizar procesos y recursos.

PALABRAS CLAVES:

Sistemas web, SOAP, REST, escalabilidad, rendimiento.

ABSTRACT

The present research intends to perform the analysis and evaluation of two architectures, REST and SOAP, for the web services development of the ADRISSMS module of the ERP system "AdrisERP".

The problem responds to the fact that today there are different architectures to perform the development of web services, for this reason makes the programmers do not know what it is the most optimal in certain points. It should also be taken into account that more people have Smart phones and tablets. This causes people to have access to the internet from different electronic devices such as PC, Smartphones, etc., so the proposal is to make a prototype in mobile version.

To do this, the analysis of the different architectures, SOAP and REST, will be carried out to know which is more optimal in the case of performance and scalability, so that companies or software developers can use them, as the case may be that they have to elaborate the systems, besides optimizing processes and resources.

KEYWORDS:

Web systems, SOAP, REST, scalability, performance.

INTRODUCCIÓN

En estos últimos tiempos el desarrollo para el uso de los servicios web tienen mucha acogida en los proveedores y clientes, es así que existe la demanda por nuevos servicios que ayuden en la labor y los tiempos de elaboración de las tareas cotidianas para que permitan que estos se reduzcan. Es por ello que las empresas que se dedican al desarrollo de estos servicios web han incrementado nuevos servicios que permitan dar soporte a esta demanda.

El desarrollo de servicios web tiene como objetivo utilizar las diferentes tecnologías para cubrir con los requerimientos solicitados por el cliente, y para ello existen diferentes arquitecturas que ayudaran a los programadores. La cantidad de arquitecturas como las formas de poder desarrollar un servicio web, cohibe la elección de éstas a los programadores, cada una de estas arquitecturas trabajan con peculiaridades distintas, permitiendo obtener algunas ventajas y desventajas en el proceso del desarrollo.

Para ello se hallara la arquitectura más apropiada, que nos permita tener un mejor desarrollo de las aplicaciones web.

En esta investigación nos encontramos con dos arquitecturas que nos permitirán realizar un mejor desarrollo de los servicios web, los cuales pasaran por un análisis y evaluación, al obtener los resultados se podrá elegir la mejor arquitectura, para así crear las aportaciones necesarias en el desarrollo de los servicios web en las diferentes empresas.



CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

1.1. Situación problemática

En el proceso de desarrollar un sitio web se presentan un conjunto de ventajas y desventajas de acuerdo a un número de factores como, quién desarrolla el sitio web, (Lajo, 2011), dónde se encuentra alojado el sitio web y quienes brindan el mantenimiento adecuado al sitio web. La decisión para cada factor, así como si el desarrollo lo realizara la empresa o si es que se encarga un proveedor, permite saber las ventajas y desventajas, como el realizar cambios significativos en su sitio web sea mucho más lento.

La elaboración de un sistema web con las configuraciones básicas, nos lleva a tener el grado de respuestas bajas, una latencia alta que puede visualizar el usuario y gran cantidad de datos redundantes transferidos por la red. (Dian Ayuba, 2013)

Hoy en día debido al avance tecnológico y al aumento de usuarios de dispositivos móviles con sistema operativo Android, la demanda de aplicaciones móviles sobre esta plataforma ha incrementado notablemente y como consecuencia de esto, también los desarrolladores para esta plataforma. El desarrollo se realiza sin el uso de buenas prácticas, por lo que se construyen aplicaciones simples, con muy poco carácter visual, niveles bajos de usabilidad, bajo performance y se construyen interfaces de usuario

pobres que dan como resultado una mala aplicación. De la misma manera, las aplicaciones muchas veces necesitan interactuar con otros sistemas de los cuales se debe consumir información a través de servicios web. Estos servicios deberían estar implementados siguiendo patrones de buenas prácticas para que la aplicación pueda tener un mejor performance. En otras palabras, los servicios deberían ser fáciles de entender y deducir para los desarrolladores. (Palacios & Aguilar, 2015)

Se tiene que las arquitecturas de modelado de web services más populares son SOAP y REST, es por ello que se entra en debate acerca de cuál sería más adecuada para un problema propuesto, es por ello que se deben contemplar las ventajas y desventajas de cada una de ellas. (Medina, 2012)

Según (Florián, 2015); Al incrementarse la demanda de disponibilidad y confiabilidad de los servicios web 24x7, por lo que representa un desafío para los administradores debido a la amplia escala de los servicios web y por la naturaleza distribuida y heterogénea de la infraestructura que los soporta. Esto se puede observar desde la facilidad de crear un medio de comunicación, como la creación de un blog, hasta la facilidad de transmitir información sin necesidad de que el receptor esté cerca al emisor, mediante e-mail o servicios de mensajería instantánea. Es por ello, que la interrupción o cortes de servicios web son perjudiciales tanto para los usuarios que brindan el servicio como para los usuarios que consumen el servicio.

1.2. Formulación del problema

Existen diversas preguntas para poder determinar cuál arquitectura es la correcta y cual escogeremos para esta tesis, es por eso que deberemos tener algunas respuestas bases que nos permitirán empezar a darle la solución a nuestro problema. Sin embargo empezaremos con una pregunta clave y nos vamos a orientar a algo más concreto así mismo nos permitirá tener una respuesta y conclusión más clara.

¿Qué arquitectura, SOAP o REST, utilizaremos para el desarrollo de servicios web aplicados al módulo de ADRISSMS del sistema ERP “AdrisERP”?

1.3. Delimitación de la investigación

Esta investigación nos permitirá mediante el análisis y la evaluación, la identificación de la mejor arquitectura SOAP o REST, la cual será recomendada y apoyada con los resultados obtenidos.

De forma que los desarrolladores de la empresa AD y L Consulting puedan realizar el desarrollo de los servicios web, para el módulo de ADRISMS del ERP ADRISERP.

1.4. Justificación e importancia de la Investigación

1.4.1. Justificación Científica

En la investigación “PLATAFORMAS DE INTEGRACIÓN. SERVICIOS WEB BASADOS EN REST Y SOAP”, dijo: Muchos diseñadores de Servicios Web están llegando a la conclusión que SOAP es demasiado complicado. Por tanto, están comenzando a utilizar Servicios Web basados en REST para mostrar cantidades de datos masivos (Carmona P.).

Por esta razón la presente investigación, de las arquitecturas SOAP y REST determinará cuál de ellas es la mejor en rendimiento y escalabilidad, lo cual servirá como apoyo a los desarrolladores en sus posteriores implementaciones.

1.4.2. Justificación económica

De acuerdo al análisis comparativo de ambas arquitecturas, SOAP y REST, podrá permitir a los diferentes desarrolladores de sistemas web, así como a las empresas, obtener un ahorro en los recursos y en los tiempos, permitiendo así obtener una mejor rentabilidad en los proyectos que se pretenden desarrollar.

1.4.3. Justificación Social

Existen diversas entidades e que no cuentan con los estándares ni patrones de desarrollo de servicios web, y aquellas instituciones que si cuentas con ello no tienen el conocimiento de cómo utilizarlas de forma adecuada. Y es así que aparecen las arquitecturas ya que la forma adecuada y correcta es mediante estas arquitecturas, las distintas instituciones o empresas manejan distintos marcos de trabajo, por tal motivo se desarrolla este proyecto que nos permitirá brindar la información adecuada sobre las arquitecturas y sus rendimientos y escalabilidad para conseguir conformidad del cliente y mejores resultados para las empresas encargadas del desarrollo.

1.5. Limitaciones de la Investigación

Dentro de las nuestras limitaciones y obstáculos que podemos presentar es la poca información, por la cual no se logró abarcar mayores conocimientos acerca del tema; así como también el corto periodo de tiempo para poder realizar la presente investigación.

1.6. Objetivos

1.6.1. Objetivo general

Analizar y evaluar las arquitecturas, REST y SOAP, para el desarrollo de servicios web del módulo ADRISSMS del sistema ERP “AdrisERP”.

1.6.2. Objetivos específicos

Los objetivos específicos se basan en:

- a) Analizar que arquitectura es la mejor para el desarrollo de servicios web.
- b) Determinar las herramientas que se utilizaran para las mediciones de las arquitecturas.
- c) Analizar los resultados de las mediciones de los indicadores utilizados.
- d) Realizar la evaluación económica de la propuesta.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de Estudios

2.1.1. Nivel Internacional

A) Según el autor Carmona P. (2014), en su investigación denominada: **“Plataformas De Integración: Servicios web basados en las arquitecturas SOAP y REST”**, cuyo objetivo principal el poder interrelacionar datos de distintas aplicaciones ya en uso. Se pudo determinar que algunos diseñadores de Servicios Web llegan a concluir que SOAP ocupa más esfuerzos. Por ello, empiezan a inclinarse por los Servicios Web utilizando la arquitectura REST para de esta manera mostrar gran cantidad de datos. Y así como esto, es el caso de algunas de las empresas más grandes como eBay y Google.

Asimismo, concluye, REST es una arquitectura que permite crear aplicaciones “cliente-servidor”, mientras que SOAP funciona como un protocolo que permite intercambiar todos los datos que se encuentren entre 2 extremos.

Finalmente, recomienda usar SOAP cuando se establece un documento formal que detalle las interacciones que tendrá la interfaz y REST si nos encontramos cuando el servicio web no necesita tener estado. (Carmona P. , 2014)

Relevancia.- Cito a este antecedente porque dio una explicación clara de las arquitecturas, así mismo porque rescato el concepto que el autor tiene al indicar que se puede trabajar con la arquitectura REST, siempre que se tenga un estilo para generar aplicaciones del lado cliente-servidor.

B) Según el autor Nicolás Cordero Fernández, en su investigación denominada: ***“Diseño de un Cliente para Dispositivos Móviles para el sistema de almacenamiento voluntario en la nube”***, el objetivo de la investigación fue el análisis, diseño e implementación de un sistema multiplataforma para el sistema de almacenamiento en la nube VoCS mediante el uso de tecnologías y estándares Web (Nicolás, 2014). Entre las conclusiones que llegaron fue se podrá utilizar la arquitectura SOAP cuando en el proyecto sea necesario procesar de forma asíncronamente, dado que REST es una arquitectura que no nos permite a sincronías. Es más adecuado utilizar la arquitectura REST siempre que el servicio Web no tenga estado, o cuando se necesite que el ancho de la banda sea limitado, REST permite diseñar de forma más fácil que la arquitectura SOAP. Por lo expresado anteriormente se determina que utilizar REST, es más práctico, ya que las capas adicionales y las cabeceras de la arquitectura SOAP crea una sobrecarga extra en dispositivos que cuentan con escasos recursos. Otra ventaja de la arquitectura REST

es su uso de caché, lo cual es un factor de mucha importancia.
(Fernández, “Diseño de un Cliente para Dispositivos Mviles para un Sistema de Almacenamiento Voluntario en la Nube”, 2014)

Relevancia.- Se citó la investigación como antecedente porque realiza una comparación entre las arquitecturas SOAP y REST, indicando las ventajas y desventajas de ambas al desarrollar un sistema web, lo cual sirve como apoyo para el presente proyecto.

2.1.2. Nivel Nacional

A. Según el autor Requena, C. (2013), en su investigación denominada: **“Servicios web basados en REST o SOAP”**, cuyo objetivo es sustentar mediante que contexto sería más apropiado utilizar los servicios web con REST o en su defecto la arquitectura SOAP. Obtuvo como resultados que los Servicios son APIS que pueden utilizarse en una red y se ejecutan en los sistemas que los aloja, Asimismo, concluye, SOAP es una arquitectura que permite el intercambio de datos basados en XML y se encuentra diseñado para que se ejecute del protocolo HTTP, mientras que REST es una arquitectura que funciona con los sistemas hipermedias. Finalmente recomienda, utilizar la arquitectura REST siempre que sea necesario. (Requena, 2013)

Relevancia.- Cito a este antecedente porque no le quita el protagonismo a la arquitectura SOAP, ya que defiende en qué casos usar esta arquitectura; sin embargo termina orientándose por la arquitectura REST, indicando que es mayormente utilizado para sistema hipermedias tales como Web, de esta manera fortalece mi investigación y mi inclinación por esta última.

B. Según los autores Cesar Ynga Palacios y Pedro Palacios Aguilar (2015), en su investigación denominada: **“Propuesta de implementación del marco de trabajo para el desarrollo de aplicaciones ANDROID”**, Cuyo objetivo es establecer un marco de trabajo que sirva como referencia para los desarrolladores de aplicaciones Android nativas; con temas referentes a lineamientos y buenas prácticas de desarrollo. Se obtuvo como resultado al utilizar la arquitectura REST en comparación con otras arquitecturas, esta es la mejor opción en lo que respecta a desarrollo de servicios para móviles, debido a que la arquitectura REST tiene la naturaleza de hacer que existe una transferencia de información mucho más rápida, y en las app móviles este pedido de calidad es de mucha importancia. Además el uso de comunicación mediante el formato de JSON, nos permite que la app móvil y el servicio web sea mucho más óptima y con una correcta solución. Esto se debe a que JSON se integra muy bien con las aplicaciones móviles y da facilidad para manejarse dentro de la aplicación (Parsear). (Palacios & Aguilar, 2015)



Relevancia.- Se citó el antecedente porque se tuvo la propuesta de la elaboración de un sistema web en Android, que mediante la búsqueda de documentación al elegir un servicio web con el cual trabajar, llegaron a la conclusión que la arquitectura REST es la mejor opción para este tipo de proyectos móviles, siendo esta información de referencia para la presente investigación.

C. Según el autor Pedro Jesús Carrión (2016), en su investigación denominada: **“Visualización de puntos de interés en un campus universitario usando realidad aumentada”**, se buscó desarrollar una app móvil que contenga información de eventos y lugares de un campus universitario, esta pretenden mostrar el lugar de ubicación de algunos puntos de interés mediante las tecnologías utilizadas.

Se realizó el análisis acerca de servicio web y dado la facilidad de desarrollo e implementación y el más bajo consumo de ancho de banda es el que se utilizó la arquitectura REST; con respecto al formato de archivo para la transferencia de información se utilizará JSON por su facilidad de uso.

Este servicio web podría ser reutilizado en un futuro por otras aplicaciones cliente, como por ejemplo un sitio web. Este servicio web, implementado con lenguaje de programación PHP, conectará el servidor Apache con la aplicación móvil desarrollada para el sistema operativo Android. (Castagnola, 2016)

Relevancia.- Se tomó en consideración el antecedente para este proyecto, porque se realizó una investigación acerca de los servicios web, en donde se llegó a la conclusión que REST es la arquitectura con la cual es más fácil de desarrollar e implementar para la elaboración de una aplicación móvil.

2.1.3. Nivel Regional

No se pudieron encontrar antecedentes de nivel regional enfocados a nuestro proyecto.

2.2. Estado del arte

Desarrollaremos este estado del arte en base a ampliar el conocimiento sobre el tema que venimos estudiando con el fin de aportar argumentos que contribuyan a justificar y definir el alcance y solución a la problemática de esta investigación.

Para ello mostraremos mediante una imagen el procedimiento que deben tener los servicios web y así llegamos a la incógnita de que estructura deberemos escoger para el mejor y más óptimo desarrollo.

Lo bueno de que existan alternativas es que se están tomando cosas buenas de uno y otro lado, cada uno resalta las deficiencias del otro, y se buscan soluciones para que éstas se minimicen. (Asúa, 2012)

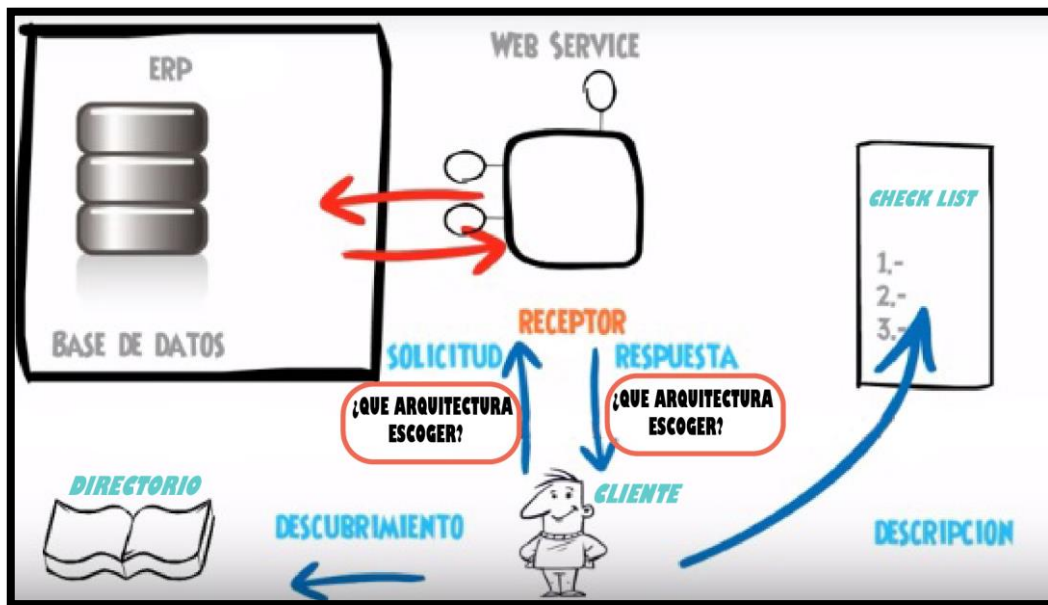


Ilustración 1. Web service "Descripción Grafica del funcionamiento de las arquitecturas"
FUENTE: <https://www.youtube.com/watch?v=O4bLR8KB9y8>

Descripción: Al igual que podemos solicitar un plato a la carta o un medicamento podemos solicitar información a nuestra base de datos en este caso al ERP, para ello deberá existir un receptor al cual le daremos nombre de web service, para ello deberemos enviar la solicitud de nuestro pedido y obtener la respuesta del mismo mediante el desarrollo que será en base a un estructura, para ello deberemos escoger que y cual estructura será la más adecuada para que esta nos pueda permitir un mejor desempeño en nuestra interacción.

2.2.1. Plataforma web

2.2.1.1. Web 2.0

En el paper titulado "What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software", nos dice que la Web 2.0 es



la red como interfaz, la cual engloba a los dispositivos conectados.
(O'Reilly, 2007)

Aplicaciones Web 2.0 son los que permiten realizar la mayor ventaja intrínsecas posible de esa plataforma: la adjudicación de software como servicio incesantemente renovados, que mejora la mayor cantidad de número de gente que lo utiliza, agotando y re mezclando información de diferentes fuentes agregando los usuarios individuales, entre tanto se proporciona su información y sus servicios de alguna manera que permita volver a mezclar por otros, el ingenio de los efectos de la red mediante una "arquitectura de participación", esto ya va mucho más allá del concepto de la metáfora de la Web 1.0 que le permite tener diferente experiencia al usuario. (O'Reilly, 2007)

2.2.1.2. Servicios Web

Según, Alberto Gonzales en su investigación titulada "Diseño e implementación de un servicio web que permita crear y consultar un vocabulario de la lengua de señas", nos indica que existe un grupo de estándares y protocolos que ayudan en el intercambio de información entre aplicaciones, adicional a ello se pueden utilizar otras aplicaciones desarrolladas en lenguajes de programación alternas, y ejecutadas sobre cualquier plataforma para intercambiar datos en redes como

Internet, en el cual solo haciendo una petición ya se puede obtener la respuesta. (Gonzales, 2010)

La posibilidad de operar con varios de los servicios web se adquiere a través de la afiliación de estándares abiertos, para poder hacer una mejor interoperabilidad se creó WS-I (web services), quien tiene a cargo el desarrollar de varios perfiles para determinar de manera más completa los estándares. (Gonzales, 2010)

2.2.1.3. *Arquitecturas de un sistema*

Hoy en día ya no son suficiente las aplicaciones web actuales. La presentación de negocio electrónico no permite la fusión de las aplicaciones de Internet con los demás software de los negocios. Las empresas necesitan adquirir mayor beneficio por lo que la solución es que los sitios web evolucionen. Este es el entorno en el cual brotan los servicios web.

Como sabemos los servicios web son factores de necesitan los software para permitir a los usuarios usar aplicaciones de negocio que compartan información con diferente programas modulares, vía Internet.

Son aplicaciones autosuficientes de la plataforma las cuales se pueden publicar con facilidad, invocadas y localizadas a través de protocolos web estándar, como UDDI, WSDL, XML, SOAP. El principal objetivo

final es realizar un directorio online de servicios web, el cual permita localizarse de forma fácil y pueda tener una alta fiabilidad.

Los protocolos empleados tienen la siguiente funcionalidad:

- a. XML: Un servicio web es una aplicación creada en XML.
- b. WSDL: Permite describir el servicio siempre y cuando este publicado. Es el lenguaje XML que los proveedores emplean para describir sus servicios web.
- c. Permite que un cliente y un servicio puedan establecer un acuerdo en lo que refiere a los detalles de contenido y transporte de mensajes, mediante un documento procesable por dispositivos.
- d. WSDL representa un documento formal entre quien lo solicita y el proveedor. WSDL especifica los mecanismos de intercambio de mensajes y sintaxis.

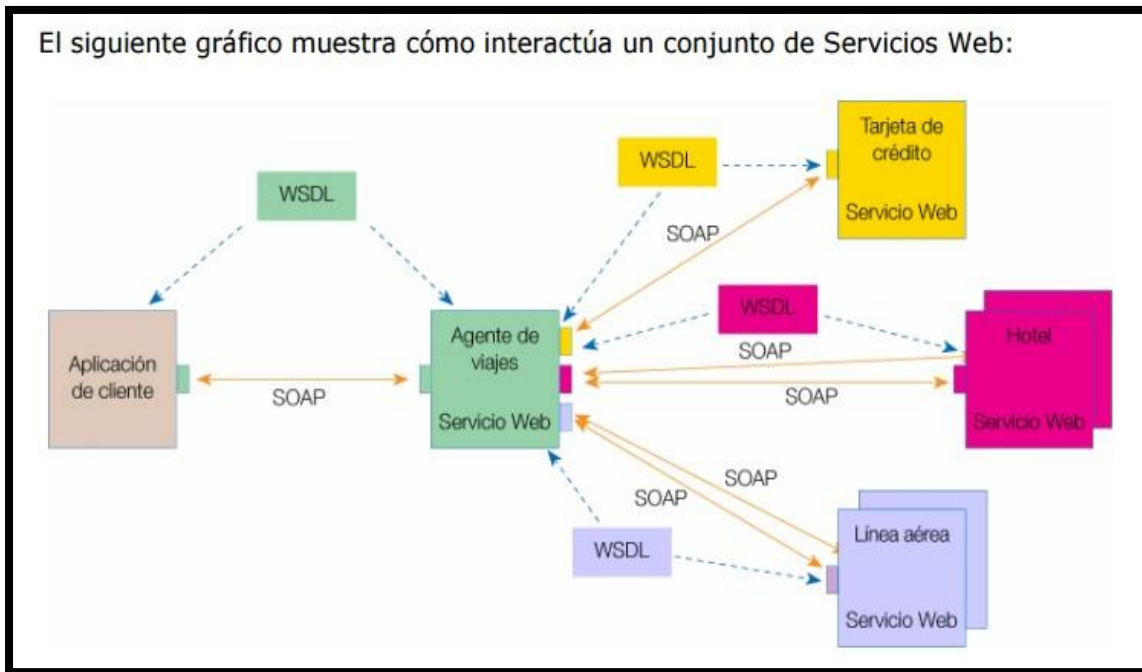


Ilustración 2. Interacción entre un conjunto de Servicios Web FUENTE: Arquitectura de servicios web (ws) / <http://www.mad.es/>

2.3. Bases teórico científicas

2.3.1. Aplicaciones web

Una aplicación web es lo que llamaríamos cliente/servidor, donde el servicio, el protocolo por el cual se comunican y el cliente, están normalizados y no son desarrollados el programador.

Se diferencian en 3 niveles: el nivel superior quien crea la interacción con el usuario, el nivel inferior quien se encarga de brindar la información a la base datos y el nivel intermedio quien trata los datos.



2.3.2. Web 2.0

En un inicio Las páginas web se utilizaban únicamente para mostrar información. Por lo cual se les llama web estáticas, dado que la información que muestra siempre de la misma forma. Una web estática está compuesta por archivos HTML, CSS y JavaScript, los cuales se encuentran ubicados en un servidor web, a quienes se les envía a un navegador web para que éste los muestre a la persona quien va a interactuar. A esta forma de trabajo se le llama Web 1.0 y se muestra en la ilustración 3. (Lajo, 2011)

Las páginas web dinámicas pusieron la base para la web 2.0. La cual tiene la particularidad a al visitante quien antes solo podía visualizar o leer el contenido, ahora pueda crear información para el sitio web, ejemplo una ventana de noticias.



Ilustración 3. Aplicaciones web 1.0 y 2.0

FUENTE: Tesis "Desarrollo De Un Sistema De Construcción De Webs 2.0 E Integración Con Un Sistema De Venta De Dominio"

Como se muestra en la ilustración 3, el concepto Web 2.0 también son aplicaciones que consumen la opción de una retroalimentación, como lo son los foros, redes sociales, blogs, entre otros.

2.3.3. Sistema Operativo Android

Sistema operativo que se utiliza en los dispositivos móviles de pantallas táctiles como Tablets y Smartphones, basado en el sistema operativo Linux, es desarrollado por Google. Actualmente, Se encuentra en su versión 5.0 (Lollipop). Este sistema es usado en una gran gama de



celulares de distintos fabricantes, con características diferentes como resolución, cámara, teclado,

etc. Esto ha hecho que tenga una gran aceptación ya que puede ser usado en distintos tipos de móviles y de distintos precios, lo que hace que pueda estar al alcance de muchos más usuarios obteniendo casi el 52% del mercado. Para la posible ejecución de este proyecto, el sistema operativo Android permitirá instalar y alojar la aplicación móvil desarrollada para este proyecto. La plataforma permite que la aplicación se pruebe directamente en el sistema operativo, haciendo mucho más fácil el desarrollo de la aplicación. Los controles que componen las interfaces de usuarios mantienen un estilo propio. La aplicación se puede instalar en Android 2.2 y versiones superiores.

2.3.4. Arquitectura SOAP y REST

2.3.4.1. SOAP

Debido a las investigaciones realizadas los desarrolladores tienen como conclusión que la arquitectura SOAP es más complicada y tediosa para su desarrollo, por esa razón se ha empezado a reemplazar esa arquitectura por la REST que permitirá visualizar gran cantidad de información masiva. (Mayta, 2012)

El problema radica de la idea inicial que se tenía para la arquitectura SOAP, ya que la tecnología fue pensada para una versión sobre Internet, de DCOM, así se muestra por su predecesor XML-RPC. Esto se debe a que estas tecnologías se basan en modelos RPC, los cuales son adecuados para campos aislados. La transformación de este tipo de sistemas es muy fácil, es cambiada para cada usuario de forma que cumpla con los nuevos pedidos. (Mayta, 2012)

Cuando la cantidad de usuarios es grande se necesita crear una estrategia diferente. Se necesita sugerir un artificio que permita explicar de manera explícita la interoperabilidad que se tiene en la API. Puede pensarse que SOAP es el más adecuado, ya que utiliza HTTP. Sin embargo la Web se maneja mejor cuando utilizamos estilos basados en REST.

La arquitectura SOAP permite a los programadores agrupar la dificultad del sistema, de forma que dé lugar a las interfaces creadas automáticamente, las cuales facilitan el diseño del sistema.

La ventaja de la arquitectura SOAP es que utiliza lenguajes de niveles altos que permiten implementar y llamar al servicio. (Flores, 2012)

Al parecer REST tendrá mayor aceptación en el futuro, pero eso va a depender de lo cambiante de la tecnología. Sin embargo actualmente se están presentando propuestas donde se pretende enlazar los principios REST con la nueva versión SOAP.



2.3.4.2. REST

REST y SOAP están afiliados a estándares para el desarrollo y diseño de RESTful o web services. Se suele dar de forma común el uso de integrar varios componentes y sistemas de una o varias plataformas. Con el uso de los estándares podremos obtener que la integración se transforme en interoperabilidad. La acción de ida y vuelta se obtiene a través del intercambio de mensajes.

Cada vez se vuelve más común ver cómo se utiliza tanto REST como SOAP para entender cómo funcionan los recursos de distintos aplicativos. Por los requisitos y la sencillez de los recursos que se tienen a través de un API REST, es que se exponen para poder interoperar con otros sistemas. Efectivamente esto se replica en una plataforma empresarial. (Arlandy, 2014)

Empezamos a conocer que no es necesario pasar por SOAP, por lo que se puede diseñar muy buenas servicios web basados en la arquitectura REST, y todo esto de una forma más sencilla. Además podríamos basarnos en aplicaciones, que cuentan con algunas desventajas pero a su vez con las ventajas para poder realizar la integración entre sí. SOAP es una forma de poder obtener algunos beneficios, sin embargo esto dependerá del tipo de negocio que se tenga, la estructura del servicio web que se necesite y la adecuada investigación.

2.3.5. Herramienta de Insomnia

Insomnia es una aplicación que se instala como complemento en el navegador Google Chrome, Insomnia te permite consumir los servicios de un Api REST de una manera sencilla y cómoda gracias a su interfaz gráfica. (Technology, 2016)

Insomnia le permite gestionar las peticiones de una manera sencilla, proporcionando las siguientes características básicas: generar parámetros de cadena de consultas, organizar las solicitudes en áreas de trabajo y grupos, fragmentos reutilizables, plantillas Swig, validación JSON, resaltado de sintaxis (JSON, JavaScript, XML, CSS...), ayudante de autenticación básica HTTP, conmutación rápida de solicitudes (estilo ctrl-p), codificación y decodificación URL, Codificación y decodificación base 64. (CHROME WEB STORE, 2016)

2.3.6. Herramienta de Valid Wsdl

WSDL se puede ejecutar manualmente contra una especificación .WSDL o recursos en el banco de trabajo. Esta opción le permite investigar y solucionar los problemas de cumplimiento WS-I; todos los problemas de validación se muestran como los errores de la lista de tareas y advertencias. También se ejecutar de forma automática, cuando se importa o se genera el WSDL. Puede importar

definiciones WSDL utilizando el asistente del modelo de mensajes. (IBM Integration Bus, 2016)

WSDL cumple con la necesidad de que un servicio Web se pueda escribir en cualquier idioma y residir en cualquier plataforma. Un documento WSDL proporciona los detalles necesarios para un solicitante de servicios contactar y consumir un servicio. (Yan, 2011)

2.4. Definición de términos básicos

A. BASE DE DATOS:

La base de datos es el conjunto de datos vinculados que permiten el monitoreo de la información de las compañías. La documentación puede ser vista como una colección de registros y cada registro está compuesto de una colección de campos. (Chávez, 2010)

B. DIRECTORIO UDDI:

UDDI es uno de los estándares de los servicios Web el cual tiene el objetivo de mostrar la documentación WSDL y poder ser accedido por los mensajes SOAP, de forma que se pueda tener respuesta de los servicios Web. (Ramírez, 2008)

C. ERP:

Un ERP es un sistema que permite la gestión de las empresas el cual está diseñado para poder realizar la automatización de los procesos del

área comercial, finanzas, producción, y otros, en una empresa. El principal objetivo es hacer las tareas del usuario de forma más fácil y sencilla. (Rey, 2010)

D. NAVEGADOR:

El navegador es el software que nos permite acceder a internet, así mismo nos permitirá pasear por las diferentes páginas web, podremos revisar nuestros correos, ingresar a contenidos multimedia, entre muchas otras actividades. Los buscadores más conocidos y comunes en el mercado son: Mozilla Firefox, Opera, Microsoft Internet Explorer, Apple Computer's Safari.

E. REST:

Es un conjunto de restricciones que, cuando son aplicadas al diseño de un sistema, crean un estilo arquitectónico de software. Navarro, R (2006).

F. SERVIDOR:

Un servidor web nos permite gestionar las aplicaciones del lado del servidor.

El código que es re seccionado por el lado del cliente es compilado y posterior a ello pasa a ser ejecutado. Es necesario utilizar protocolos para

por así realizar la transmisión de la información que se va a utilizar. Usualmente utilizamos para las comunicaciones HTTP, el cual pertenece a al modelo OSI. El cual es utilizado cuando queremos referirnos al ordenador en el que se va a ejecutar el programa.

G. SISTEMA INFORMÁTICO:

Es el conjunto de los elementos con los cuales se realiza el proceso de automatización de toda la información. La Organización Internacional de Normalización permite definir las funciones de procesos, entradas, salidas y controles, para poder así llevar la secuencia de las operaciones con datos. (Rodríguez, 2003)

H. SOAP:

La arquitectura Soap es un mecanismo que permite exponer la información la cual se presenta de acuerdo a modelos de empaquetado de datos modular y diversos de mecanismos de codificación de datos. Así nos permitirá que la arquitectura SOAP se pueda utilizar en rangos que trabajen los modelos de comunicación RPC (Remote Procedure Call).

I. TIEMPOS DE RESPUESTA:

Este tiempo de respuesta se puede determinar como aquel tiempo que transcurre desde que se envía la comunicación y nos permite recibir la respuesta. El tiempo de respuesta se debe a los siguientes tiempos:

- a. El tiempo que toma en esperar en el transmisor.
- b. Tiempo que se utiliza en realizar la transferencia del mensaje.
- c. Tiempo de la difusión, la cual dependerá de la arquitectura de la red, y la cantidad de saltos al satélite.
- d. Tiempo que tome en procesarse el mensaje, y tiempo adecuado para poder obtener la respuesta.
- e. Atraso producto del protocolo.

J. WEB SERVICE:

Un Servicio Web es el factor software que se permite registrar, mostrar a la luz por medio de protocolos estándares, que se aprueba plantear y poder tener servicios de los sistemas.

Los Servicios Web, independientes capaces de integrarse pueden formar sistemas distribuidos complejos, cada uno es el responsable de realizar las diferentes funciones en concreto. (Cueli, 2009)

CAPÍTULO III: MARCO METODOLÓGICO

3.1. Tipo y diseño de investigación

3.1.1. Tipo de investigación

Para esta investigación bajo el esquema cuantitativo, se utiliza este paradigma ya que nos permitirá la recolección de información sobre nuestras arquitecturas REST y SOAP, y el análisis de datos para así poder contestar las preguntas de investigación, de cual arquitectura es la mejor en rendimiento y escalabilidad, y poder probar nuestra hipótesis establecida; por tanto tenemos como tipo de investigación la descriptiva; porque nos permitirá detallar y ordenar las características de las diferentes arquitecturas para poder realizar una comparativa, así mismo podremos observar los resultados obtenidos de cada indicador, para poder desarrollar los servicios web.

3.1.2. Diseño de la investigación

Esta investigación es de diseño, Descriptiva comparativa: ya que identificara, describirá y analizara las características de las dos arquitecturas que presentamos, las cuales son REST y SOAP, así como también observaremos el comportamiento de las variables indicadas.

Decimos que es de tipo comparativa, porque se evaluará los resultados obtenidos en los indicadores, que medirán el rendimiento y la escalabilidad.

3.2. Población y muestra

Unidad de análisis.- Análisis y evaluación de las arquitecturas para el desarrollo de servicios web.

3.2.1. Población

Como población tenemos todas las metodologías con las cuales podremos desarrollar los servicios web

Hay muchos estándares que se relacionan a los servicios web, entre ellos:

- a) XML
- b) HTTP
- c) SOAP
- d) REST
- e) WSDL
- f) UDDI

3.2.2. Muestra

La muestra para este proyecto de investigación serán dos de las arquitecturas orientadas a servicios web, las que nos permitirán reconocer la información que requerimos para así encontrar la mejor en rendimiento y escalabilidad.

- a) Arquitectura SOAP
- b) Arquitectura REST

3.3. Hipótesis

Mediante el análisis y la evaluación de los indicadores de rendimiento y escalabilidad de las arquitecturas “Soap y Rest”, permitirán encontrar cuál es la arquitecturas es la mejor para el desarrollo de servicios web del módulo ADRISSMS del sistema ERP “AdrisERP”.

3.4. Operacionalización

(Ver anexo 97 - 98)

3.4.1. Variable independiente

El rendimiento y escalabilidad

- a) La escalabilidad, (Nivel de calidad del servicio web).
- b)

b) El rendimiento de las arquitecturas, (tiempo promedio de carga del servicio web).

VARIABLE INDEPENDIENTE	DEFINICION CONCEPTUAL	INDICADORES	INSTRUMENTO	FORMULA	UNIDADES DE MEDIDA
Escalabilidad y rendimiento.	<p>Escalabilidad. Se define como el problema que surge al crecer la aplicación y el tiempo que este demanda en recurso de personal en conocimiento y desarrollo.</p>	La escalabilidad, (Nivel de calidad del servicio web).	ISO/IEC 9126	$\frac{\sum \text{metricas}}{\text{cant metricas}}$	Decimal
	<p>Rendimiento, Está definido como aquel tiempo que transcurre al enviar una comunicación y el tiempo en que se toma en tener la respuesta.</p>	El rendimiento de las arquitecturas, (tiempo promedio de carga del servicio web).	ISO/IEC 9126	—	Milisegundos

Tabla 1. Operacionalización de variable independiente.
Fuente: Elaboración propia

Leyenda:

Métricas = Se encuentran determinadas mediante la Norma ISO/IEC 9126, la métrica tiene el valor de rangos de 0 a 10 que indicara de menor a mayor calidad de acuerdo a las características propuestas (rendimiento y escalabilidad).



3.4.2. Variable dependiente

La mejor arquitectura.

a) Curva de aprendizaje.

b) Tiempo de desarrollo por cada servicio web utilizando las arquitecturas REST y SOAP.

VARIABLE DEPENDIENTE	DEFINICION CONCEPTUAL	INDICADORES	INSTRUMENTO	FORMULA	UNIDADES DE MEDIDA
La mejor arquitectura SOAP o REST.	La arquitectura SOAP es soportada por los desarrolladores de software y los analistas.	Curva de aprendizaje.	Juicio de Expertos	$\frac{E1+E2+E3+E4}{4}$	Entero
	REST intenta imitar el protocolo HTTP y fijar la interfaz en un grupo de trabajos estándares.	Tiempo de desarrollo por cada servicio web utilizando las arquitecturas REST y SOAP	Juicio de Expertos	$\frac{E1+E2+E3+E4}{4}$	Entero

Tabla 2. Operacionalización, variable dependiente.
Fuente: Elaboración propia

Leyenda:

E = Experto en el tema.



3.5. Métodos, técnicas e instrumentos de recolección de datos

3.5.1. Métodos de recolección de datos

De Análisis: Se realizara este método de análisis ya que se tendrá que desordenar los elementos que se estudiaran en partes para así poder aprender las propiedades y sus riesgos. Para este proyecto tendremos que aprender la conducta de cada arquitectura, en sus tiempos de carga y respuesta.

Experimental: Debido a que se ejecutara a raíz de una escena real de algún problema, acercándose al desarrollo del servicio web para cada arquitectura, en ella se realizara la comprobación de la hipótesis y la fundamentación de la elaboración.

3.5.2. Técnicas

La técnica que se utilizara para la recolección de información es: el juicio de expertos, análisis documental, la observación.

Análisis Documental:

Para este análisis se recolectaran la información de fuentes secundarias. Revistas, libros, estas se utilizaran como fuentes para así poder obtener información sobre las variables de interés.

Observación:

Es el análisis que realizan los asesores, y el jurado encargado de la calificación de la presente tesis, con respecto a la investigación de las dos arquitecturas y sus tiempos de rendimiento y escalabilidad.

Juicio de expertos:

Este es un método que permite una validación con respecto a la fiabilidad de la útil para verificar la fiabilidad de una búsqueda realizada, basándose en el conocimiento y su trayectoria en el tema expuesto, además son reconocidos como expertos calificados pudiendo dar así valoraciones y juicios.

3.5.3. Instrumentos de recolección de datos

(Ver anexo 2)

Ficha de registro de datos

En las fichas se registrarán los elementos de información encontrada y la fuente consultada, algunos se agregan otros más: Ubicación, Autor, Título, y fecha.

Guía de observación

La guía se estructura a través de columnas que ayudan en la organización de la información recolectada.

Este documento permitirá encapsular las observaciones realizadas con respecto a diferentes fenómenos.

Planilla de Juicio de Expertos

Aquí los expertos escogidos con las capacidades y conocimientos relacionados a la investigación realizada darán un puntaje al indicador para luego obtener un promedio mediante las sumas de los puntajes.

3.6. Procedimiento para la recolección de datos

Escoger una arquitectura u otra es un poco difícil, por lo cual esta investigación ha sido elaborada basándose en varios documentos (tesis, paper, páginas web), y sobre todo enfocándonos en comparaciones, sin embargo este tema resulta un poco más extenso de lo que parece, de tal manera para concluir con la investigación se tendrá que hacer una comparativa habiendo recolectado el significado y procedimiento de cada arquitectura.

Para este procedimiento se analizarán las dos arquitecturas de acuerdo al estudio, y la documentación obtenida, para después de ello realizar la evaluación de ambas mediante las herramientas correspondientes para la selección de la mejor arquitectura.



3.7. Análisis estadístico e interpretación de datos

Utilizaremos la ya conocida herramienta Excel, para las mediciones de los datos estadísticos que se utilizarán en la media aritmética.

Tabulación de datos.- Se ordenarán por diferentes preguntas, hechas en el transcurso de la investigación, las cuales nos han ayudado a obtener respuestas claras.

Análisis de datos.- Cada análisis realizado será interpretado por indicadores.

3.7.1. Herramienta

Excel. Con la herramienta Excel podremos realizar las tablas y calcular los datos. Excel facilita la creación de tablas que calculan de un modo automático los valores numéricos que especifiquemos, así mismo nos permite crear gráficos.

Excel es parte de un grupo de productos con el nombre de “Office”, los cuales te permiten crear presentaciones o documentos.

3.7.2. Medida de datos estadísticos

Media aritmética. Sumatoria de los valores de cada variable, entre la cantidad total de elementos.

$$\bar{X} = \frac{x_1 + x_2 + x_3 + \dots + x_{n-1} + x_n}{n} = \frac{\sum_{i=1}^n x_i}{n}$$

En el caso de que el valor X_i de la variable X se repita N_i veces, aparecerá en la expresión de la media aritmética:

$$\bar{X} = \frac{\sum x_i n_i}{n}$$

, será la expresión que se considere definitiva

de la media aritmética.

Como $f_i = \frac{n_i}{N}$ otra expresión sería $\bar{X} = \sum_{i=1}^n x_i f_i$

3.8. Criterios éticos

Los criterios éticos que se muestran en esta tesis es el Código Deontológico del Colegio de Ingenieros del Perú, que se encuentra ubicado en el Capítulo III “*Faltas Contra la Ética Profesional y Sanciones*” y el sub Capítulo II “*De la Relación con El Público*” en el 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.

Con este proyecto de investigación se pretende en todo lo posible crear un aporte claro sobre las arquitecturas “SOAP y REST”, al poder estar disponible la presente tesis para todo aquel que la requiera.

Objetividad: El análisis se basara en hipótesis y conclusiones verdaderas con ideas claras y resultados óptimos.



Originalidad: Mostraremos las fuentes bibliográficas de donde se obtuvieron la información que se mostrara en dicha tesis.

Veracidad: La información mostrada en dicha investigación será verdadera y correcta para que puedan tomarla como referencia en las próximas investigaciones.

3.9. Criterios de rigor científico

Contrastación: Se realizara el desarrollo de la tesis procurando dar como finalizada la interrogante que se tenía, así mismo darle respuesta a la hipótesis plateada, con la finalidad de saber con qué arquitectura desarrollaremos la versión móvil del ARIS ERP.

CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

4.1. Resultados en tablas y gráficos.

4.1.1. Selección de arquitecturas a utilizar

Se realizó una investigación en tesis, paper y páginas web sobre las arquitecturas SOAP y REST, para así con esta información fundamentar porque fue la selección de dichas arquitecturas. El resultado de la búsqueda fue la siguiente:

ARQUITECTURA SOAP			
MEDIO DE INVESTIGACIÓN	Tesis: "Diseño de un Cliente para Dispositivos Móviles para un Sistema de Almacenamiento Voluntario en la Nube"	Página Web: http://informaciontecnologiachiny.blogspot.pe/	Paper: "Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON"
JUSTIFICACIÓN	Los Servicios Web SOAP, están soportados por la mayoría de tecnologías. Una de sus características fundamentales es que añade interoperabilidad entre plataformas. Además de ser más fácil de utilizar SOAP cuando la arquitectura necesite procesar asíncronamente	SOAP requiere menos código de plumbing code de servicios REST, (es decir, las transacciones, la seguridad, la coordinación, direccionamiento, la confianza, etc.) La mayoría de las aplicaciones en el mundo real no son simples y apoyar las operaciones complejas, que requieren para mantener el estado de conversación y la información contextual. Con el enfoque de SOAP, los desarrolladores no tienen que preocuparse acerca de cómo escribir el código de plomería en la capa de aplicación a sí mismos.	El primero hace referencia al protocolo usado para la comunicación entre cliente y servidor intercambiando mensajes basados en XML (WSDL).

Tabla 3. Selección de la arquitectura SOAP
Fuente: Elaboración propia



ARQUITECTURA REST			
MEDIO DE INVESTIGACIÓN	Tesis: Propuesta de implementación de un marco de trabajo para el desarrollo de aplicaciones Android	Página Web: http://restysoap.blogspot.pe/	Tesis: “Diseño de un Cliente para Dispositivos Móviles para un Sistema de Almacenamiento Voluntario en la Nube”
JUSTIFICACIÓN	REST define un conjunto de principios de arquitectura por los cuales se puede diseñar servicios web que se enfoquen en recursos. Los componentes de la comunicación (cliente-servidor) usan estos recursos, los cuales son direccionados y transferidos sobre http por un amplio rango de clientes escritos en diferentes lenguajes.	Rest es muy ligero, sus respuestas contienen exactamente la información que se necesita (datos planos), pues utiliza pocos recursos. REST es una arquitectura sin estado, significa que cada petición al servidor es tratada de manera totalmente independiente. La ventaja está en la simplicidad del componente servidor y lo fácil que resulta escalar y hacer crecer el sistema.	Es más útil utilizar REST cuando el servicio Web no tiene estado, cuando se utiliza una infraestructura de catching que mejora el rendimiento, cuando el ancho de banda necesita ser limitado. REST ofrece un bajo consumo de recursos y ofrece una gran escalabilidad. Una de sus características principales es que es muy fácil de construir y acoplar.

Tabla 4. Selección de la arquitectura REST
Fuente: Elaboración propia

4.1.2. Selección de herramientas: medición en rendimiento y escalabilidad

En el transcurso de selección de las herramientas que he utilizado para las pruebas, requerí realizar una investigación en algunas páginas web confiables, donde me indicaban los datos más relevantes sobre las herramientas enfocadas en las pruebas, en el cual obtuve el resultado siguiente:



HERRAMIENTAS	INSOMNIA		VALID WSDL	
MEDIO DE INVESTIGACIÓN	PÁGINA WEB: (Technology, 2016)	PÁGINA WEB (CHROME WEB STORE, 2016)	PÁGINA WEB (IBM Integration Bus, 2016)	TESIS: "Seguridad de Servicios Web Utilizando SOAP, WSDL y UDDI"
JUSTIFICACIÓN	Insomnia permite consumir los servicios de un Api REST de una manera sencilla y cómoda gracias a su interfaz gráfica.	Permite gestionar las peticiones de una manera sencilla, generando parámetros de cadena de consultas, fragmentos reutilizables, entre otras características.	Puede ejecutar manualmente contra una especificación .Wsdlo recursos en el banco de trabajo.	Cumple con la necesidad de que un servicio Web se pueda escribir en cualquier idioma y residir en cualquier plataforma.

Tabla 5. Selección de Herramientas para medir tiempos
Fuente: Elaboración Propia

4.1.3. Resultados de las mediciones en rendimiento y escalabilidad

4.1.3.1. Resultados de las mediciones de Rendimiento

Arquitecturas Soap y Rest						
NOMBRE DE INVESTIGACIÓN	"Análisis y evaluación de las arquitecturas REST y SOAP para el desarrollo de servicios web aplicados al ERP "AdrisERP" y su versión móvil en Android."					
Arquitecturas	SOAP			REST		
CRITERIOS	RENDIMIENTO			RENDIMIENTO		
TIEMPO EN MS	Listar	Insertar	Buscar	Listar	Insertar	Buscar
		402299000	1723572000	416322000	1944	778

Tabla 6. Mediciones en rendimiento de las arquitecturas SOAP Y REST
Fuente: Elaboración propia



A. PROMEDIO DE DATOS

- Promedio Listar: $402299000 + 1944 = \frac{402300944}{2} = 201150472$ ms
- Promedio Insertar: $1723572000 + 778 = \frac{1723572778}{2} = 861786389$ ms
- Promedio Buscar: $416322000 + 1825 = \frac{416323825}{2} = 208161912.5$ ms

B. INTERPRETACIÓN DE RESULTADOS

B.1. Rendimiento para la arquitectura REST

El rendimiento para la arquitectura REST según la herramienta INSOMNIA, es:

- **LISTAR:** Es de 1944 ms, habiendo sido el menor promedio calculado de las dos arquitecturas con un valor de: 201150472 ms.
- **INSERTAR:** Es de 778 ms, habiendo sido el menor promedio calculado de las dos arquitecturas con un valor de: 861786389 ms.
- **BUSCAR:** Es de 1825 ms, habiendo sido el menor promedio calculado de las dos arquitecturas con un valor de: 208161912.5 ms.

B.2. Rendimiento para la arquitectura SOAP

El rendimiento para la arquitectura SOAP mediante la herramienta VALID WSDL es el siguiente:

- **LISTAR:** Es de 402299000 ms, habiendo sido el mayor promedio calculado de las dos arquitecturas con un valor de: 201150472 ms.

- **INSERTAR:** Es de 1723572000 ms habiendo sido el mayor promedio calculado de las dos arquitecturas con un valor de: 861786389 ms.
- **BUSCAR:** Es de 416322000 ms, habiendo sido el mayor promedio calculado de las dos arquitecturas con un valor de: 208161912.5 ms.

C. PROMEDIO DE DATOS DE RENDIMIENTO DE LAS ACCIONES DE CADA

ARQUITECTURA:

- REST: $1944 + 778 + 1825 = \frac{4547}{3} = 1515.7$ ms
- SOAP: $402299000 + 1723572000 + 416322000 = \frac{2542193000}{3} = 847397666.7$

ms

D. RESULTADO DE DATOS DE LAS ARQUITECTURAS

Según los resultados obtenidos en tiempos de rendimiento mediante la herramienta INSOMNIA para REST y VALID WSDL para SOAP; el más óptimo es la arquitectura REST, con un tiempo de 1515.7 ms.

4.1.3.2. Resultados de las mediciones de Escalabilidad

4.1.3.2.1. Arquitecturas SOAP

(Ver anexo 101 – 104)

Métricas de evaluación:

Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:



CONFIABILIDAD:

Satisfactoria **10**
 Insatisfactoria **0**

PORTABILIDAD:

Si **10**
 No **0**

FUNCIONALIDAD/EFICIENCIA:

Cumple **10**
 Aceptable **7**
 Mínima **4**
 Inaceptable **0**

TABLA DE MEDICIÓN DE CALIDAD DEL SERVICIO WEB						
ARQUITECTURA		SOAP				
CARACTERISTICA	EXPERTO 1	EXPERTO 2	EXPERTO 3	EXPERTO 4	OPERACIÓN	PROMEDIO DE EXPERTOS POR CARACTERISTICAS
FUNCIONALIDAD	4	7	10	7	28/4	7
CONFIABILIDAD	10	10	0	10	30/4	7.5
EFICIENCIA	10	7	10	7	34/4	8.5
PORTABILIDAD	0	0	10	0	10/4	2.5

*Tabla 7 Medición de características de la arquitectura SOAP mediante la ISO/IEC 9126
 Fuente: Elaboración Propia*

Mediante la norma ISO/IEC 9126 se ha evaluado las características siguientes:

Funcionalidad:

- El experto 1, Juan Jiner Delgado Chávez, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 4 puntos.
- El experto 2, Miguel Ángel Vidaurre indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 7 puntos.
- El experto 3, Miguel Díaz Vidarte, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 10 puntos.



- El experto 4, Alex Muro Núñez, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 7 puntos.

Confiabilidad:

- El experto 1, Juan Jiner Delgado Chávez, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 10 puntos.
- El experto 2, Miguel Ángel Vidaurre, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 10 puntos.
- El experto 3, Miguel Díaz Vidarte, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 0 puntos.
- El experto 4, Alex Muro Núñez, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 10 puntos.

Eficiencia:

- El experto 1, Juan Jiner Delgado Chávez, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 10 puntos.
- El experto 2, Miguel Ángel Vidaurre, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 7 puntos.
- El experto 3, Miguel Díaz Vidarte, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 10 puntos.
- El experto 4, Alex Muro Núñez, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 7 puntos.



Portabilidad:

- El experto 1, Juan Jiner Delgado Chávez, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 0 puntos.
- El experto 2, Miguel Ángel Vidaurre, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 0 puntos.
- El experto 3, Miguel Díaz Vidarte, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 10 puntos.
- El experto 4, Alex Muro Núñez, indico el siguiente puntaje para la funcionalidad de la arquitectura SOAP: 0 puntos.

4.1.3.2.2. Arquitectura REST
(Ver anexo 101 – 104)

Métricas de evaluación:

Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:

CONFIABILIDAD:

Satisfactoria **10**

Insatisfactoria **0**

PORTABILIDAD:

Si **10**

No **0**

FUNCIONALIDAD/EFICIENCIA:

Cumple **10**

Aceptable **7**

Mínima **4**

Inaceptable **0**



TABLA DE MEDICIÓN DE CALIDAD DEL SERVICIO WEB						
ARQUITECTURA		REST				
CARACTERÍSTICA	EXPERTO 1	EXPERTO 2	EXPERTO 3	EXPERTO 4	OPERACIÓN	PROMEDIO DE EXPERTOS POR CARACTERÍSTICAS
FUNCIONALIDAD	7	10	10	4	31/4	7.75
CONFIABILIDAD	10	10	10	0	30/4	7.5
EFICIENCIA	10	10	7	10	39/4	9.25
PORTABILIDAD	10	10	10	10	40/4	10

Tabla 8 Medición de características de la arquitectura Rest mediante la ISO/IEC 9126
 Fuente: Elaboración Propia

Funcionalidad:

- El experto 1, Juan Jiner Delgado Chávez, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 7 puntos.
- El experto 2, Miguel Ángel Vidaurre, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 10 puntos.
- El experto 3, Miguel Díaz Vidarte, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 10 puntos.
- El experto 4, Alex Muro Núñez, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 4 puntos.

Confiabilidad:

- El experto 1, Juan Jiner Delgado Chávez, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 10 puntos.



- El experto 2, Miguel Ángel Vidaurre, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 10 puntos.
- El experto 3, Miguel Díaz Vidarte, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 10 puntos.
- El experto 4, Alex Muro Núñez, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 0 puntos.

Eficiencia:

- El experto 1, Juan Jiner Delgado Chávez, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 10 puntos.
- El experto 2, Miguel Ángel Vidaurre, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 10 puntos.
- El experto 3, Miguel Díaz Vidarte, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 7 puntos.
- El experto 4, Alex Muro Núñez, indico el siguiente puntaje para la funcionalidad de la arquitectura REST: 10 puntos.

Portabilidad:

- El experto 1, Juan Jiner Delgado Chávez, se obtuvo el siguiente puntaje de las funcionalidades de la arquitectura REST: 10 puntos.
- El experto 2, Miguel Ángel Vidaurre, se obtuvo el siguiente puntaje de las funcionalidades de la arquitectura REST: 10 puntos.

- El experto 3, Miguel Díaz Vidarte, se obtuvo el siguiente puntaje de las funcionalidades de la arquitectura REST: 10 puntos.
- El experto 4, Alex Muro Núñez, se obtuvo el siguiente puntaje de las funcionalidades de la arquitectura REST: 10 puntos.

4.1.3.2.3. Promedio de medición: características de calidad de las arquitecturas

PROMEDIO DE MEDICION DE LAS CARACTERISTICAS DE CALIDAD ARQUITECTURAS		
CARACTERISTICA	SOAP	REST
FUNCIONALIDAD	7	7.75
CONFIABILIDAD	7.5	7.5
EFICIENCIA	8.5	9.25
PORTABILIDAD	2.5	10

Tabla 9. Promedio de medición de características de calidad las arquitecturas
Fuente: Elaboración Propia

- a) El promedio obtenido por los 4 expertos para las características en la arquitectura SAOP:
- FUNCIONALIDAD: 7 puntos
 - CONFIABILIDAD: 7.5 puntos
 - EFICIENCIA: 8.5 puntos
 - PORTABILIDAD: 2.5 puntos
- b) El promedio obtenido por los 4 expertos para las características en la arquitectura REST:
- FUNCIONALIDAD: 7.75 puntos



- CONFIABILIDAD: 7.5 puntos
- EFICIENCIA: 9.25 puntos
- PORTABILIDAD: 10 puntos

A. PROMEDIO DE DATOS DE ESCALABILIDAD DE LAS CARACTERISTICAS DE CADA ARQUITECTURA:

- SOAP: $7 + 7.5 + 8.5 + 2.5 = \frac{25.5}{4} = 6.38$ p
- REST: $7.75 + 7.5 + 9.25 + 10 = \frac{34.5}{4} = 8.63$ p

B. RESULTADO DE DATOS DE LAS ARQUITECTURAS

Según la medición de la calidad, realizada mediante la evaluación de los expertos para encontrar el promedio de escalabilidad para REST y SOAP; el más óptimo es la arquitectura REST, con un promedio de 8.63.

4.1.4. Curva De Aprendizaje

(Ver anexo 105 – 108)

Métricas de evaluación:

Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:

EXCELENTE: 10

MUY BUENO: 7 - 9

BUENO: 4 - 6

MEJORABLE: 1 - 3

EXPERTO 1: Juan Jiner Delgado Chávez

CURVA DE APRENDIZAJE		
	Arquitecturas	
ATRIBUTOS	SOAP	REST
Fase de Análisis (tiempo comprensión)	10	10
Fase de Construcción (desarrollo)	10	9
PERACIÓN	20/2	19/2
PROMEDIO	10	9.5

*Tabla 10. Curva de Aprendizaje - Experto 1
Fuente: Elaboración Propia*

• **INTERPRETACION DATOS:**

De acuerdo al puntaje colocado por el experto Ing. Juan Jiner Delgado Chávez, se ha obtenido:

Arquitecturas:

- **SOAP:** 10 = 10 (Excelente)
- **REST:** 9.5 = 10(Excelente)



EXPERTO 2: Miguel Ángel Vidaurre

CURVA DE APRENDIZAJE		
	Arquitecturas	
ATRIBUTOS	SOAP	REST
Fase de Análisis (tiempo comprensión)	9	9
Fase de Construcción (desarrollo)	8	9
PERACIÓN	17/2	18/2
PROMEDIO	8.5	9

*Tabla 11. Curva de Aprendizaje - Experto 2
Fuente: Elaboración Propia*

• **INTERPRETACION DATOS:**

De acuerdo al puntaje colocado por el experto Ing. Miguel Ángel Vidaurre, se ha obtenido:

Arquitecturas:

- **SOAP:** 8.5 = 9 (Muy bueno)
- **REST:** 9 = 9 (Muy bueno)



EXPERTO 3: Miguel Díaz Vidarte

CURVA DE APRENDIZAJE		
	Arquitecturas	
ATRIBUTOS	SOAP	REST
Fase de Análisis (tiempo comprensión)	10	10
Fase de Construcción (desarrollo)	10	10
PERACIÓN	20/2	20/2
PROMEDIO	10	10

*Tabla 12. Curva de Aprendizaje - Experto 3
Fuente: Elaboración Propia*

• **INTERPRETACION DATOS:**

De acuerdo al puntaje colocado por el experto Ing. Miguel Díaz Vidarte, se ha obtenido:

Arquitecturas:

- **SOAP:** 10 = 10 (Excelente)
- **REST:** 10 = 10 (Excelente)



EXPERTO 4: Alex Muro Núñez

CURVA DE APRENDIZAJE		
	Arquitecturas	
ATRIBUTOS	SOAP	REST
Fase de Análisis (tiempo comprensión)	8	10
Fase de Construcción (desarrollo)	10	10
PERACIÓN	18/2	20/2
PROMEDIO	9	10

*Tabla 13. Curva de Aprendizaje - Experto 4
Fuente: Elaboración Propia*

• **INTERPRETACION DATOS:**

De acuerdo al puntaje colocado por el experto Ing. Alex Muro Núñez, se ha obtenido:

Arquitecturas:

- **SOAP:** 9= 9 (Muy Bueno)
- **REST:** 10 = 10 (Excelente)



4.1.4.1. Promedio para la curva de aprendizaje de acuerdo a los expertos:

PROMEDIO DE LA CURVA DE APRENDIZAJE		
EXPERTOS	Arquitecturas	
	SOAP	REST
Experto 1	10	9.5
Experto 2	8.5	9
Experto 3	10	10
Experto 4	9	10
OPERACIÓN	37.5/4	38.5/4
PROMEDIO	9.38	9.63

*Tabla 14. Promedio de la curva de aprendizaje de los expertos
Fuente: Elaboración Propia*

El promedio obtenido mediante el puntaje de todos los expertos para la arquitectura SOAP en lo que es la curva de aprendizaje, 9 puntos que nos indica una un valor MUY BUENO.

El promedio obtenido en los 4 expertos para la arquitectura REST, en lo que sería la curva de aprendizaje es de 10 puntos, lo que nos indica que el valor es EXCELENTE.



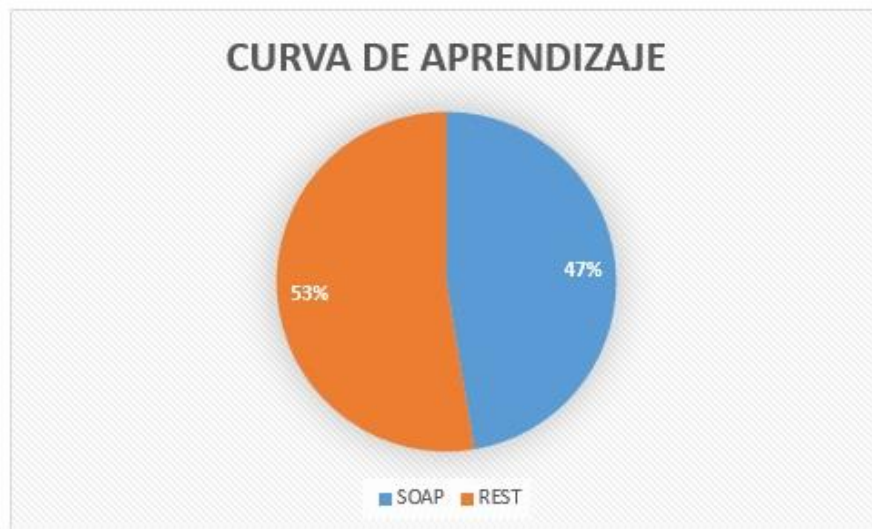


Ilustración 4. Gráfico del Puntaje Promedio de la curva de aprendizaje.
Fuente: Elaboración Propia

4.1.5. Tiempo del desarrollo

(Ver anexo 101 – 104)

Métricas de evaluación:

Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:

EXCELENTE: 10

MUY BUENO: 7 - 9

BUENO: 4 - 6

MEJORABLE: 1 - 3



EXPERTO 1: JUAN JINER DELGADO CHÁVEZ

TIEMPO EN EL DESARROLLO		
	Arquitecturas	
ATRIBUTOS	SOAP	REST
Fase de Planificación	8	9
Fase de Análisis (tiempo comprensión)	10	10
Fase de Construcción (desarrollo)	10	10
OPERACION	28/3	29/3
PROMEDIO	9.3	9.6

*Tabla 15. Tiempos de desarrollo – Experto 1.
Fuente: Elaboración Propia*

• **INTERPRETACION DATOS:**

El puntaje dado por el experto Ing. Juan Jiner Delgado Chávez, tiene el promedio para cada Arquitectura:

Arquitecturas:

- **SOAP:** 9 = 9 (Muy bueno)
- **REST:** 9.6 = 10 (Excelente)



EXPERTO 2: MIGUEL ANGEL VIDAURRE

TIEMPO DE DESARROLLO		
	Arquitecturas	
ATRIBUTOS	Soap	Rest
Fase de Planificación	9	8
Fase de Análisis	9	9
Fase de Construcción	9	9
PERACIÓN	27/3	26/3
PROMEDIO	9	8.6

*Tabla 16. Tiempos de desarrollo - Experto 2
Fuente: Elaboración Propia*

• **INTERPRETACION DATOS:**

El puntaje dado por el experto Ing. Miguel Ángel Vidaurre, tiene el promedio para cada Arquitectura:

Arquitecturas:

- **SOAP:** 9 = 9 (Muy bueno)
- **REST:** 8.6 = 9 (Muy bueno)



EXPERTO 3: MIGUEL DÍAZ VIDARTE

TIEMPO DE DESARROLLO		
	Arquitecturas	
ATRIBUTOS	Soap	Rest
Fase de Planificación	9	10
Fase de Análisis	10	10
Fase de Construcción	10	10
PERACIÓN	29/3	30/3
PROMEDIO	9.6	10

*Tabla 17. Tiempos de desarrollo - Experto 3
Fuente: Elaboración Propia*

• **INTERPRETACION DATOS:**

El puntaje dado por el experto Ing. Miguel Díaz Vidarte tiene el promedio para cada Arquitectura:

Arquitecturas:

- **SOAP:** 9.6 = 10 (Excelente)
- **REST:** 10 = 10 (Excelente)



EXPERTO 4: ALEX MURO NUÑEZ

TIEMPO DE DESARROLLO		
	Arquitecturas	
ATRIBUTOS	Soap	Rest
Fase de Planificación	9	9
Fase de Análisis	10	9
Fase de Construcción	8	8
PERACIÓN	27/3	26/3
PROMEDIO	9	8.6

*Tabla 18. Tiempos de desarrollo - Experto 4
Fuente: Elaboración Propia*

• **INTERPRETACION DATOS:**

El puntaje dado por el experto Ing. Alex Muro Núñez, tiene el promedio para cada Arquitectura:

Arquitecturas:

- **SOAP:** 8.6 = 9 (Muy Bueno)
- **REST:** 9.6 = 10 (Excelente)



4.1.5.1. Promedio para el tiempo de desarrollo de las arquitecturas por cada experto

PROMEDIO PARA EL TIEMPO DE DESARROLLO		
EXPERTOS	Arquitecturas	
	Soap	Rest
Experto 1	9.3	9.6
Experto 2	9	8.6
Experto 3	9.6	10
Experto 4	9	8.6
OPERACION	36.9/4	36.8/4
PROMEDIO	9.2	9.2

Tabla 19. Promedio para el tiempo de desarrollo
Fuente: Elaboración Propia

El promedio obtenido de todos los expertos para las arquitecturas SOAP y REST en el tiempo de desarrollo es de 9 puntos, lo cual nos indica que es un valor MUY BUENO.



Ilustración 5. Promedio del tiempo de desarrollo – Grafica
Fuente: Elaboración Propia



4.1.6. Evaluación económica de la propuesta.

Esta evaluación la realice en 3 pasos:

4.1.6.1. Costos

4.1.6.1.1. Costos de la inversión

RECURSO HUMANO			
Descripción	Pago por Hora	Días de trabajo	Total (S/.)
Programador	50	10	500
Estadístico	50	15	750
Analista	50	10	500
Total			1750

Tabla 20. Inversión de Recurso Humano
Fuente: Elaboración Propia

UTILES DE ESCRITORIO			
Descripción	Cantidad	Precio Unitario (S/.)	Total (S/.)
Papel Bond A-4 (1/2 millar)	1	13.00	13.00
Lapicero Pilot	6	2.00	12.00
Memorias USB (4Gb) SanDisk	1	35.00	35.00
Engrampadora Artesco	1	5.00	5.00
Perforador Artesco	1	7.00	7.00
Caja de grapas 50 unid	1	5.00	5.00
Plumones acrílicos	4	2.50	10.00
Pizarra acrílica	1	40.00	40.00
Cuaderno Stamford doble anillado 200 hojas	1	30.00	30.00
Corrector Artesco	1	2.00	2.00
Portatodo	1	30.00	30.00
TOTAL			S/.189.00

Tabla 21. Inversión de Útiles de escritorio
Fuente: Elaboración Propia



HARDWARE		
Nombre	Descripción	Precio (S/.)
Laptop Toshiba	Procesador : Intel 32 bits x2 Ram: 12 GB Core i5	2800.00
Laptop Toshiba	Procesador : Intel 32 bits x2 Ram: 3 GB Core i5	1500.00
Impresora HP		150.00
TOTAL		S/. 4450.00

Tabla 22. Inversión de Hardware
Fuente: Elaboración Propia

4.1.6.1.2. Costos de operación

RECURSOS HUMANOS			
	Cantidad	Sueldo	Total (S/.)
Operarios	4	50	200
Total			S/. 200.00

Tabla 23. Recursos Humanos
Fuente: Elaboración Propia

DEPRECIACION			
	Costo	Depreciación (%)	Total (S/.)
Hardware	3,000 .00	25	750
Total			S/. 750.00

Tabla 24. Inversión de la depreciación
Fuente: Elaboración Propia

MATERIALES DE ESCRITORIO			
	Costo	Depreciación (%)	Total (S/.)
Hardware	3000	25	750
Total			S/. 750.00

Tabla 25. Inversión de los materiales de escritorio
Fuente: Elaboración Propia



4.1.6.2. BENEFICIOS

4.1.6.2.1. Beneficios Tangibles

Los beneficios tangibles que fueron aportados por el sistema propuestos:

- S/. 940.00 - Reducción de costos en los materiales de oficina
- S/. 1000.00 - Ahorro en los suministros para los equipos
- S/. 1000.00 - Disminución de tiempo en horas/hombre para la búsqueda de información
- S/. 700.00 - Aumento en la rentabilidad.

4.1.6.2.2. Beneficios Intangibles

Los beneficios intangibles del sistema que fueron propuestos:

- Poder dentro del proceso de negocio optimizar las actividades
- Ayuda en el empleo de recursos de forma financiera y material.
- Flexibilidad al poder manejar un volumen y diversidad de información con oportunidad, precisión y rapidez, para facilitar labores.
- Mayor actualización de la información y reducción en la fuerza de trabajo y control de los recursos.
- Mayor y mejor aprovechamiento de los recursos tecnológicos instalados.



4.1.6.3. Valor Presente Neto

FÓRMULA

$$VPN = \sum_{L=0}^{L=n} \frac{Bi - Ci}{\left(1 + \frac{I}{100}\right)^n}$$

- Información:

I = La tasa de interés anual en el tiempo es: 10%

Ci= Inversión S/. 7,638.00, de operación S/.985

n = Vida útil en años (5 años)

Bi= Los beneficios S/. 3,640.00

- **OPERACIÓN**

$$VPN = \frac{0 - 7638}{1} + \frac{3640 - 985}{\left(1 + \frac{10}{100}\right)^1} + \frac{3640 - 985}{\left(1 + \frac{10}{100}\right)^2} + \frac{3640 - 985}{\left(1 + \frac{10}{100}\right)^3} + \frac{3640 - 985}{\left(1 + \frac{10}{100}\right)^4} + \frac{3640 - 985}{\left(1 + \frac{10}{100}\right)^5}$$

$$VPN = -7638 + 2413.64 + 2194.21 + 1996.24 + 1806.12 + 1649.07$$

$$2421.28 = VPN$$

- **ANÁLISIS:**

VPN > 0, es factible que el desarrollo del proyecto.

Tasa del retorno de la inversión

$$PR = \frac{7638}{3640 - 985} = 2.88 \text{ años}$$

$$PR = 0.88 (12) = 10.56 \text{ meses}$$

$$PR = 0.56 (30) = 16.8 \text{ días}$$

El retorno de Inversión tienes una tasa de 2 años, 10 meses y 17 días.



4.2. Contrastación de hipótesis

Se realizó las validaciones mediante las arquitecturas REST y SOAP, y el análisis del rendimiento y escalabilidad; siendo REST el más óptimo tanto en rendimiento con 1515.7 ms y en escalabilidad con un promedio de 8.63.

4.3. Discusión de resultados

4.3.1. Selección de arquitecturas

Para la presente investigación se obtuvo que la mejor Arquitectura es REST, contrastando con la investigación *“Propuesta de Arquitectura Orientada a Servicios para el Módulo de Inventario del ERP Cubano”*, donde señala que la mejor Arquitectura sería REST. (Puebla, 2009)

*Según la investigación realizada en el artículo “**Arquitectura y estrategia de API Un enfoque coordinado**”, nos dice: Esto resulta particularmente cierto en casos de API abiertas y, en especial, en aquellas centradas en dispositivos móviles. Como norma, a los desarrolladores de aplicaciones no les gusta SOAP como lenguaje de programación. Además, las herramientas disponibles para crear clientes de servicio web no suelen ser compatibles con dispositivos móviles. Aparte de las consideraciones de carácter práctico, existe un problema de percepción: el empleo del tipo de diseño de servicio web puede hacer que su organización parezca un “dinosaurio” torpe, y eso estaría*

relacionado con una reducción de la adopción entre desarrolladores de aplicaciones móviles; por lo que nos atrevemos a dar como la Arquitectura más óptima a REST.

4.3.2. Selección de herramientas

4.3.2.1. Insomnia

Según el informe de la conferencia titulada “API de servicio Web REST” se menciona que Insomnia es una aplicación multiplataforma para organizar, ejecutar y depurar peticiones HTTP, reutiliza valores comunes como claves de la API o los identificadores de sesión, obtiene todos los detalles en cada respuesta. Se visualiza el código de estado, el cuerpo y cabeceras. (API de servicio Web REST, 2016)

4.3.2.2. Valid WSDL

Según Lin Yan, en su tesis titulada “Seguridad de Servicios Web Utilizando SOAP, WSDL y UDDI” nos indica que WSDL cumple con la necesidad de que un servicio Web se pueda escribir en cualquier idioma y residir en cualquier plataforma. Un documento WSDL proporciona los detalles necesarios para un solicitante de servicios contactar y consumir un servicio. (Yan, 2011)

Concluyendo que tanto Insomnia y WSDL son herramientas fáciles y útiles al trabajar con los servicios Web, para este caso, la ejecución de pruebas de rendimiento al emplear las arquitecturas REST y SOAP.



4.3.3. Análisis de Rendimiento y Escalabilidad

4.3.3.1. Rendimiento

Según Alberto Los Santos Aransay, nos dice que SOAP en un inicio no fue creado con el objetivo de ser eficiente, ni para poder enviar un mensaje, ya que incluye demasiados datos adicionales. Hemos intentado revisar cómo se podría solucionar esta desventaja, o al menos mejorar este rendimiento, puesto que hay bastantes iniciativas entorno a este aspecto. Además detallar, que existe cada vez más actividad con la idea de dotar de características extras a los Servicios Web basados en REST, como podría ser una opción equiparable a WS-Security en SOAP. (Aransay, 2009)

La tesis titulada “*WS SOAP VERSUS REST EN SOA*”, nos dice los servicios REST son más desconocidos que los servicios de la arquitectura SOAP. Pero estos son más fáciles de usar, más ligeros en términos de rendimiento. (Hevia, 2011)

Afirmando así el resultado obtenido mediante las herramientas utilizadas para medir cuál de las dos arquitecturas es más eficaz en cuanto a rendimiento, el cual fue obtenido del promedio de sus tres criterios, siendo 1515.7 ms para la Arquitectura Rest.

4.3.3.2. Escalabilidad

Según Alberto Los Santos Aransay, nos dice: Se puede apreciar que los Servicios Web bajo la arquitectura REST son mucho mejor que los de la arquitectura SOAP en relación a la escalabilidad. (Aransay, 2009)

Por otro lado en la tesis titulada *“Diseño de un Cliente para Dispositivos Móviles para un Sistema de Almacenamiento Voluntario en la Nube”*, no indica que la arquitectura REST tiene un estilo subyacente a la Web y una de sus cualidades es la escalabilidad de la iteración con los componentes. La Web ha crecido exponencialmente y esto no ha perjudicado el rendimiento. Se puede ingresar desde varios tipos de clientes, desde varias estaciones de trabajo a dispositivos móviles. REST ofrece un bajo consumo de recursos y ofrece una gran escalabilidad. (Fernández, “Diseño de un Cliente para Dispositivos Moviles para un Sistema de Almacenamiento Voluntario en la Nube”, 2014)

Según *Palacios Aguilar y Ynga Palacios*, indican que la arquitectura propuesta, REST, cumple este atributo de calidad al poder interpretar con otros sistemas a través de servicios web, hasta incluso conectarse directamente a la base de datos, sin importar el motor al que se requiere conectar (MySQL, Oracle, PostgreSQL, SQL Server, entre otros). (Palacios & Aguilar, 2015)

Mediante el análisis realizado a las diferentes investigaciones, cabe destacar que la arquitectura REST es recomendada en cuanto a escalabilidad al poseer la característica de calidad, así mismo indicar que los resultados de medición de escalabilidad en el presente proyecto fue 8.63 de promedio que fue hallado por la fórmula de calidad que se realizó a los diferentes resultados de las evaluaciones de los expertos.

4.3.4. Evaluación económica de la propuesta.

En esta investigación se utilizaron 10 días con un costo de S/. 500.00 para realizar las actividades siguientes: el plan de actividades, los fundamentos teóricos y recursos, y al compararla con la tesis "**SISTEMA DE CONTROL DE INVENTARIO**" tuvo un tiempo de 11 días y un costo de S/.550.00 soles, haciendo así la diferencia en costos y tiempo. (Rojas, 2009)

Esta investigación tiene como finalidad poder acortar los tiempos de las búsquedas de información para todos los desarrolladores de sistemas web y poder acortar los tiempos en horas/hombre, para poder aumentar la rentabilidad de las empresas.

4.3.5. Curva De Aprendizaje

4.3.5.1. Arquitectura SOAP

Esta investigación se obtuvo con el valor promedio de acuerdo a los expertos y sus puntajes en la curva de aprendizaje para la Arquitectura SOAP obtuvo el puntaje 9 (muy bueno), lo que significa que la curva de aprendizaje de esta investigación es diferente a la información que nos brinda en el informe “*Simple Object Access Protocol*”, donde expresan que SOAP obtiene una curva de aprendizaje excelente, ya que manifiestan que se acopla de forma adecuada. (Ledo, 2007)

4.3.5.2. Arquitectura REST

Esta investigación tiene un valor promedio de los expertos en la curva de aprendizaje para la Arquitectura REST un puntaje de 10 (excelente), la curva de aprendizaje desarrollada en la presente investigación es conforme a la tesis “*Desarrollo e implementación de una API RESTful aplicada a un Sistema Académico Universitario*”, donde expresan que Rest obtiene una curva de aprendizaje excelente, ya que manifiestan que en cortos periodos de tiempo es posible alcanzar grandes resultados. (Arce, 2012)

4.3.6. Tiempo Desarrollo

Según las investigaciones realizadas durante la siguiente tesis, de acuerdo a promedio realizado a nuestros expertos nos indican que el tiempo de desarrollo entre una arquitectura y otra no difiere, obteniendo un valor 9p (Muy Bueno) para ambas arquitecturas, sin embargo el documento *“Arquitectura y estrategia de API Un enfoque coordinado”*, nos indica que: A los desarrolladores de aplicaciones no les gusta SOAP como lenguaje de programación. Además, las herramientas disponibles para crear clientes de servicio web no suelen ser compatibles con dispositivos móviles.

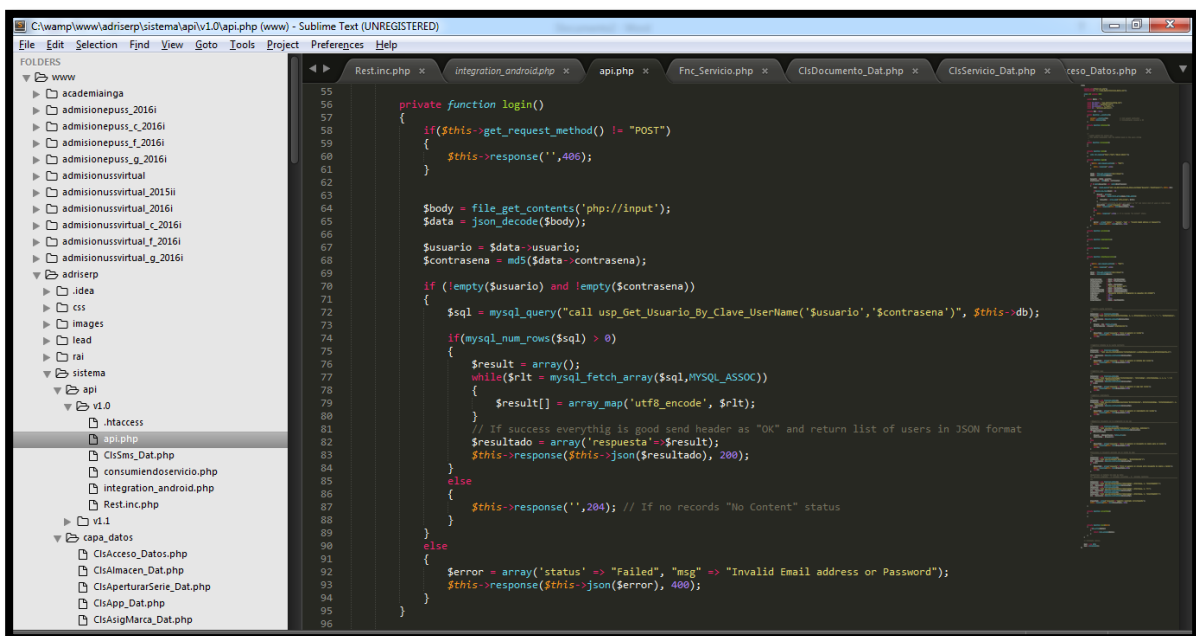
CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN

5.1. Documentación

5.1.1. Servicios de la Arquitectura REST

5.1.1.1. Código de los Servicios

5.1.1.1.1. Login



```

55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
private function login()
{
    if($this->get_request_method() != "POST")
    {
        $this->response("",406);
    }

    $body = file_get_contents('php://input');
    $data = json_decode($body);

    $usuario = $data->usuario;
    $contrasena = md5($data->contrasena);

    if (!empty($usuario) and !empty($contrasena))
    {
        $sql = mysql_query("call usp_Get_Usuario_By_Clave_UserName('$usuario','$contrasena')", $this->db);

        if(mysql_num_rows($sql) > 0)
        {
            $result = array();
            while($rit = mysql_fetch_array($sql,MYSQL_ASSOC))
            {
                $result[] = array_map('utf8_encode', $rit);
            }
            // If success everything is good send header as "OK" and return list of users in JSON format
            $resultado = array("respuesta"=>$result);
            $this->response($this->json($resultado), 200);
        }
        else
        {
            $this->response("",204); // If no records "No Content" status
        }
    }
    else
    {
        $error = array("status" => "Failed", "msg" => "Invalid Email address or Password");
        $this->response($this->json($error), 400);
    }
}
    
```



5.1.1.1.2. Listar Clientes

```

219         else
220         {
221             $sql = mysql_query("call
                usp_sms_get_clientes_consumos_sms", $this->db);
222
223             if(mysql_num_rows($sql) > 0)
224             {
225                 $result = array();
226                 while($r1t = mysql_fetch_array($sql,MYSQL_ASSOC))
227                 {
228                     $result[] = array_map('utf8_encode', $r1t);
229                 }
230                 // If success everythig is good send header as "OK"
                // and return list of users in JSON format
231                 $resultado = array('respuesta'=>$result);
232                 $this->response($this->json($resultado), 200);
233             }
234         }
235     }
236     else
237     {
238         $this->response('',204); // If no records "No
                Content" status
239     }
240 }
241
    
```

5.1.1.1.3. Buscar Clientes

```

189     private function clientes()
190     {
191         // Cross validation if the request method is GET else it will return "Not
                Acceptable" status
192         if($this->get_request_method() != "GET")
193         {
194             $this->response('',406);
195         }
196         $name=$this->_request['name'];
197
198         if (empty($name))
199         {
200             $sql = mysql_query("call usp_sms_get_clientes_sms_by_nombre('$name')", $this->
                db);
201
202             if(mysql_num_rows($sql) > 0)
203             {
204                 $result = array();
205                 while($r1t = mysql_fetch_array($sql,MYSQL_ASSOC))
206                 {
207                     $result[] = array_map('utf8_encode', $r1t);
208                 }
209                 // If success everythig is good send header as "OK" and return list of
                users in JSON format
210                 $resultado = array('respuesta'=>$result);
211                 $this->response($this->json($resultado), 200);
212             }
213         }
214         else
215         {
216             $this->response('',204); // If no records "No Content" status
217         }
218     }
    
```



5.1.1.1.4. Registrar Venta

```

243 private function clienteservicios()
244 {
245     if($this->get_request_method() != "POST")
246     {
247         $this->response("",406);
248     }
249     $json = file_get_contents('php://input');
250     $data = json_decode($json);
251
252     $nPerctacodigo = $data->nPerctacodigo;
253     $fctacteimporte = $data->fctacteimporte;
254     $cctacteicibo = "";
255     $nsercodigo = $data->nsercodigo;
256     $cctactecelosa = "servicios Extras Api";
257     $cpercodigo = $data->cpercodigo;
258     $nctactecomodigo = $data->nctactecomodigo;
259     $cctactecomumero = $data->cctactecomumero;
260     $docobserv = "Documento Interno // Asignacion de paquetes SMS APIREST";
261     $ndoctype = 1011;
262     $nestado = 1;
263     $nparclase = 6001;
264     $ncantidadsms = $data->ncantidadsms;
265
266     //registro cuenta corriente
267     $conexion = new ClAccesoDatos();
268     $vconsulta = "call usp_ins_ctacte($nPerctacodigo, 3, 1, $fctacteimporte, 1, 1, '', '', 'cctactecelosa', 0,1)";
269     $rs = $conexion->ejecutar_consulta($vconsulta);
270     if ($rs)
271     {
272         $rowsms = $rs->fetch_array();
273         $cctacteicibo = $rowsms['cctacteicibo'];
274     }
275     else
276     {
277         $resultado = array('respuesta'=>'Error al generar el recibo');
278         $this->response($this->json($resultado),406);
279         exit();
280     }
281 }
282
283
284
285
286
287
288
289

```

```

243 private function clienteservicios()
244 {
245     if($this->get_request_method() != "POST")
246     {
247         $this->response("",406);
248     }
249     $json = file_get_contents('php://input');
250     $data = json_decode($json);
251
252     $nPerctacodigo = $data->nPerctacodigo;
253     $fctacteimporte = $data->fctacteimporte;
254     $cctacteicibo = "";
255     $nsercodigo = $data->nsercodigo;
256     $cctactecelosa = "servicios Extras Api";
257     $cpercodigo = $data->cpercodigo;
258     $nctactecomodigo = $data->nctactecomodigo;
259     $cctactecomumero = $data->cctactecomumero;
260     $docobserv = "Documento Interno // Asignacion de paquetes SMS APIREST";
261     $ndoctype = 1011;
262     $nestado = 1;
263     $nparclase = 6001;
264     $ncantidadsms = $data->ncantidadsms;
265
266     //registro cuenta corriente
267     $conexion = new ClAccesoDatos();
268     $vconsulta = "call usp_ins_ctacte($nPerctacodigo, 3, 1, $fctacteimporte, 1, 1, '', '', 'cctactecelosa', 0,1)";
269     $rs = $conexion->ejecutar_consulta($vconsulta);
270     if ($rs)
271     {
272         $rowsms = $rs->fetch_array();
273         $cctacteicibo = $rowsms['cctacteicibo'];
274     }
275     else
276     {
277         $resultado = array('respuesta'=>'Error al generar el recibo');
278         $this->response($this->json($resultado),406);
279         exit();
280     }
281 }
282
283
284
285
286
287
288
289

```



```

328 //Registrar Documento con la asignación de los sms
329 .....
330 $soConexion = new CLSAcceso_Datos();
331 $vConsulta = "call usp_Reg_Documento('$docCodigo', $DocTipo, $Estado)";
332 $soCasiPlansms = $soConexion->Ejecutar_Instruccion($vConsulta);
333 if ($soCasiPlansms)
334 {
335     $rowsms = $soCasiPlansms->fetch_array();
336     $docCodigo = $rowsms[0];
337 }
338 else
339 {
340     $resultado = array('respuesta'=>'Error al generar el documento de compra para el recibo');
341     $this->response($this->json($resultado), 400);
342     exit();
343 }
344 .....
345 #vinculamos el documento generado con el recibo de pago
346 .....
347 $soConexion = new CLSAcceso_Datos();
348 $vConsulta = "call usp_Ins_DocRef('$docCodigo', '$ctactaRecibo')";
349 $rs = $soConexion->Ejecutar_Instruccion($vConsulta);
350 if (!$rs)
351 {
352     $resultado = array('respuesta'=>'Error al generar el vinculo entre documento de compra y recibo');
353     $this->response($this->json($resultado), 400);
354     exit();
355 }
356 .....
357 #registramos el detalle del plan de datos
358 #1 mensajes asignados - 2 mensajes consumidos - 3 mensajes restantes
359 .....
360 $soConexion = new CLSAcceso_Datos();
361 $vConsulta = "call usp_Set_DocParametro('$docCodigo', $nParClase, 1, '$cantidadSMS')";
362 $rs = $soConexion->Ejecutar_Instruccion($vConsulta);
363 $soConexion = new CLSAcceso_Datos();
364 $vConsulta = "call usp_Set_DocParametro('$docCodigo', $nParClase, 2, '0')";
365 $rs = $soConexion->Ejecutar_Instruccion($vConsulta);
366 $soConexion = new CLSAcceso_Datos();
367 $vConsulta = "call usp_Set_DocParametro('$docCodigo', $nParClase, 3, '$cantidadSMS')";
368 $rs = $soConexion->Ejecutar_Instruccion($vConsulta);
369 .....
370 $resultado = array('respuesta'=>'registro realizado correctamente');
371 $this->response($this->json($resultado), 200);
372 }
373 .....
374 .....
375 .....
376 .....
377 .....
378 .....
379 .....
380 .....
381 .....
382 .....
383 .....
384 .....
385 .....
386 .....
387 .....
388 .....
389 .....
390 .....
391 .....
392 .....
393 .....
394 .....
395 .....
396 .....
397 .....
398 .....
399 .....
400 .....
401 .....
402 .....
403 .....
404 .....
405 .....
406 .....
407 .....
408 .....
409 .....
410 .....
411 .....
412 .....
413 .....
414 .....
415 .....
416 .....
417 .....
418 .....
419 .....
420 .....
421 .....
422 .....
423 .....
424 .....
425 .....
426 .....
427 .....
428 .....
429 .....
430 .....
431 .....
432 .....
433 .....
434 .....
435 .....
436 .....
437 .....
438 .....
439 .....
440 .....
441 .....
442 .....
443 .....
444 .....
445 .....
446 .....
447 .....
448 .....
449 .....
450 .....
451 .....
452 .....
453 .....
454 .....
455 .....
456 .....
457 .....
458 .....
459 .....
460 .....
461 .....
462 .....
463 .....
464 .....
465 .....
466 .....
467 .....
468 .....
469 .....
470 .....
471 .....
472 .....
473 .....
474 .....
475 .....
476 .....
477 .....
478 .....
479 .....
480 .....
481 .....
482 .....
483 .....
484 .....
485 .....
486 .....
487 .....
488 .....
489 .....
490 .....
491 .....
492 .....
493 .....
494 .....
495 .....
496 .....
497 .....
498 .....
499 .....
500 .....
501 .....
502 .....
503 .....
504 .....
505 .....
506 .....
507 .....
508 .....
509 .....
510 .....
511 .....
512 .....
513 .....
514 .....
515 .....
516 .....
517 .....
518 .....
519 .....
520 .....
521 .....
522 .....
523 .....
524 .....
525 .....
526 .....
527 .....
528 .....
529 .....
530 .....
531 .....
532 .....
533 .....
534 .....
535 .....
536 .....
537 .....
538 .....
539 .....
540 .....
541 .....
542 .....
543 .....
544 .....
545 .....
546 .....
547 .....
548 .....
549 .....
550 .....
551 .....
552 .....
553 .....
554 .....
555 .....
556 .....
557 .....
558 .....
559 .....
560 .....
561 .....
562 .....
563 .....
564 .....
565 .....
566 .....
567 .....
568 .....
569 .....
570 .....
571 .....
572 .....
573 .....
574 .....
575 .....
576 .....
577 .....
578 .....
579 .....
580 .....
581 .....
582 .....
583 .....
584 .....
585 .....
586 .....
587 .....
588 .....
589 .....
590 .....
591 .....
592 .....
593 .....
594 .....
595 .....
596 .....
597 .....
598 .....
599 .....
600 .....
601 .....
602 .....
603 .....
604 .....
605 .....
606 .....
607 .....
608 .....
609 .....
610 .....
611 .....
612 .....
613 .....
614 .....
615 .....
616 .....
617 .....
618 .....
619 .....
620 .....
621 .....
622 .....
623 .....
624 .....
625 .....
626 .....
627 .....
628 .....
629 .....
630 .....
631 .....
632 .....
633 .....
634 .....
635 .....
636 .....
637 .....
638 .....
639 .....
640 .....
641 .....
642 .....
643 .....
644 .....
645 .....
646 .....
647 .....
648 .....
649 .....
650 .....
651 .....
652 .....
653 .....
654 .....
655 .....
656 .....
657 .....
658 .....
659 .....
660 .....
661 .....
662 .....
663 .....
664 .....
665 .....
666 .....
667 .....
668 .....
669 .....
670 .....
671 .....
672 .....
673 .....
674 .....
675 .....
676 .....
677 .....
678 .....
679 .....
680 .....
681 .....
682 .....
683 .....
684 .....
685 .....
686 .....
687 .....
688 .....
689 .....
690 .....
691 .....
692 .....
693 .....
694 .....
695 .....
696 .....
697 .....
698 .....
699 .....
700 .....
701 .....
702 .....
703 .....
704 .....
705 .....
706 .....
707 .....
708 .....
709 .....
710 .....
711 .....
712 .....
713 .....
714 .....
715 .....
716 .....
717 .....
718 .....
719 .....
720 .....
721 .....
722 .....
723 .....
724 .....
725 .....
726 .....
727 .....
728 .....
729 .....
730 .....
731 .....
732 .....
733 .....
734 .....
735 .....
736 .....
737 .....
738 .....
739 .....
740 .....
741 .....
742 .....
743 .....
744 .....
745 .....
746 .....
747 .....
748 .....
749 .....
750 .....
751 .....
752 .....
753 .....
754 .....
755 .....
756 .....
757 .....
758 .....
759 .....
760 .....
761 .....
762 .....
763 .....
764 .....
765 .....
766 .....
767 .....
768 .....
769 .....
770 .....
771 .....
772 .....
773 .....
774 .....
775 .....
776 .....
777 .....
778 .....
779 .....
780 .....
781 .....
782 .....
783 .....
784 .....
785 .....
786 .....
787 .....
788 .....
789 .....
790 .....
791 .....
792 .....
793 .....
794 .....
795 .....
796 .....
797 .....
798 .....
799 .....
800 .....
801 .....
802 .....
803 .....
804 .....
805 .....
806 .....
807 .....
808 .....
809 .....
810 .....
811 .....
812 .....
813 .....
814 .....
815 .....
816 .....
817 .....
818 .....
819 .....
820 .....
821 .....
822 .....
823 .....
824 .....
825 .....
826 .....
827 .....
828 .....
829 .....
830 .....
831 .....
832 .....
833 .....
834 .....
835 .....
836 .....
837 .....
838 .....
839 .....
840 .....
841 .....
842 .....
843 .....
844 .....
845 .....
846 .....
847 .....
848 .....
849 .....
850 .....
851 .....
852 .....
853 .....
854 .....
855 .....
856 .....
857 .....
858 .....
859 .....
860 .....
861 .....
862 .....
863 .....
864 .....
865 .....
866 .....
867 .....
868 .....
869 .....
870 .....
871 .....
872 .....
873 .....
874 .....
875 .....
876 .....
877 .....
878 .....
879 .....
880 .....
881 .....
882 .....
883 .....
884 .....
885 .....
886 .....
887 .....
888 .....
889 .....
890 .....
891 .....
892 .....
893 .....
894 .....
895 .....
896 .....
897 .....
898 .....
899 .....
900 .....
901 .....
902 .....
903 .....
904 .....
905 .....
906 .....
907 .....
908 .....
909 .....
910 .....
911 .....
912 .....
913 .....
914 .....
915 .....
916 .....
917 .....
918 .....
919 .....
920 .....
921 .....
922 .....
923 .....
924 .....
925 .....
926 .....
927 .....
928 .....
929 .....
930 .....
931 .....
932 .....
933 .....
934 .....
935 .....
936 .....
937 .....
938 .....
939 .....
940 .....
941 .....
942 .....
943 .....
944 .....
945 .....
946 .....
947 .....
948 .....
949 .....
950 .....
951 .....
952 .....
953 .....
954 .....
955 .....
956 .....
957 .....
958 .....
959 .....
960 .....
961 .....
962 .....
963 .....
964 .....
965 .....
966 .....
967 .....
968 .....
969 .....
970 .....
971 .....
972 .....
973 .....
974 .....
975 .....
976 .....
977 .....
978 .....
979 .....
980 .....
981 .....
982 .....
983 .....
984 .....
985 .....
986 .....
987 .....
988 .....
989 .....
990 .....
991 .....
992 .....
993 .....
994 .....
995 .....
996 .....
997 .....
998 .....
999 .....
1000 .....

```

5.1.1. Servicios de la Arquitectura SOAP

5.1.1.1. Código de los Servicios

5.1.2.1.1. Login

```

1 <?php
2 require_once('nusoap/lib/nusoap.php');
3 require_once('.../capa_datos/clsacceso_datos.php');
4
5 function getLogin($usuario, $clave)
6 {
7     $usuario = $usuario;
8     $contrasena = md5($clave);
9
10    if (strlen(trim($usuario)) > 0 and strlen(trim($contrasena))>0)
11    {
12        $soConexion = new CLSAcceso_Datos();
13        $vConsulta = "call usp_Get_Usuario_By_Clave_UserName('$usuario', '$contrasena')";
14
15        $rs = $soConexion->Ejecutar_Consulta($vConsulta);
16        if ($rs)
17        {
18            $texto = "Ingreso autorizado al sistema . . .";
19            return utf8_encode($texto);
20        }
21        else
22        {
23            return utf8_decode(implode(", ", array("Usuario o clave incorrectos")));
24        }
25    }
26    else
27    {
28        return utf8_decode(implode(", ", array("El usuario y clave son datos obligatorios")));
29    }
30 }
31
32
33 $server = new soap_server();
34 $server->configureWSDL("login", "urn:login");
35
36 $server->register("getLogin",
37     array("usuario" => "xsd:string", "clave" => "xsd:string"),
38     array("return" => "xsd:string"),
39     "urn:login",
40     "urn:login#getLogin",
41     "rpc",
42     "encoded",
43     "Login al sistema");
44
45 $server->service($HTTP_RAW_POST_DATA);
46

```



5.1.2.1.2. Listar Clientes

```

1 <?php
2 require_once('nusoap/lib/nusoap.php');
3 require_once('../capa_datos/ClasAcceso_Datos.php');
4
5 function getServicios($listar)
6 {
7     $nTipoCodigo = 1;
8     $nServicioCodigo = 0;
9     $nSerCodigo = 0;
10    $cSerNombre = '-';
11
12    $oConexion = new ClasAcceso_Datos();
13    $vConsulta = "call usp_erp_get_paquetes_sms($nTipoCodigo, $nServicioCodigo, $nSerCodigo, '$cSerNombre')";
14
15    $rs = $oConexion->Ejecutar_consulta($vConsulta);
16    if ($rs)
17    {
18        $texto = "[";
19        while ($reg=$rs->fetch_array())
20        {
21            $texto .= $reg[0]."," - ".$reg[1]."," - ".$reg[2]."," - ".$reg[5]."," - ".$reg[6]."," - ".$reg[7]."," - ".$reg[8].",";
22        }
23        $texto = substr($texto, 0, (strlen($texto)-1));
24        $texto = $texto."]";
25        return utf8_encode($texto);
26    }
27    else
28    {
29        return utf8_decode(join(", ", array("No se encontro información de Servicios")));
30    }
31 }
32
33 $server = new soap_server();
34 $server->configureWSDL("servicio", "urn:servicio");
35
36 $server->register("getServicios",
37     array("listar" => "xsd:string"),
38     array("return" => "xsd:string"),
39     "urn:servicio",
40     "urn:servicio#getServicios",
41     "rpc",
42     "encoded",
43     "Listado de Servicios");
44
45 $server->service($HTTP_RAW_POST_DATA);
46 >>

```



5.1.2.1.3. Buscar Clientes

```

1 <?php
2 require_once('nusoap/lib/nusoap.php');
3 require_once('../capa_datos/clsAcceso_Datos.php');
4
5 function getClientes($name)
6 {
7     $oConexion = new clsAcceso_Datos();
8     if (strlen($name)==0)
9     {
10         $vConsulta="call usp_sms_get_clientes_consumos_sms";
11         $rs = $oConexion->Ejecutar_Consulta($vConsulta);
12         if ($rs)
13         {
14             $texto = "[";
15             while ($reg=$rs->fetch_array())
16             {
17                 $texto .= $reg[0]. " - " . $reg[1]. " - " . $reg[2]. " - " . $reg[3]. " - " . $reg[4]. " - " . $reg[5]. " - " . $reg[6]. ",";
18             }
19
20             $texto = substr($texto, 0, (strlen($texto)-1));
21             $texto = $texto."]";
22             return utf8_encode($texto);
23         }
24     }
25     {
26         return utf8_decode(join(", ", array("No se encontro información de clientes")));
27     }
28 }
29
30 elseif(strlen($name)>0)
31 {
32     $vConsulta="call usp_sms_get_clientes_sms_by_nombre('$name')";
33     $rs = $oConexion->Ejecutar_Consulta($vConsulta);
34     if ($rs)
35     {
36         $texto = "[";
37         while ($reg=$rs->fetch_array())
38         {
39             $texto .= $reg[0]. " - " . $reg[1]. " - " . $reg[2]. " - " . $reg[3]. " - " . $reg[4]. " - " . $reg[5]. " - " . $reg[6]. ",";
40         }
41
42         $texto = substr($texto, 0, (strlen($texto)-1));
43         $texto = $texto."]";
44         return utf8_encode($texto);
45     }
46     {
47         return utf8_decode(join(", ", array("No hay registro de clientes con estos datos: ".$name.".")));
48     }
49 }

```

```

40
41     $texto = substr($texto, 0, (strlen($texto)-1));
42     $texto = $texto."]";
43     return utf8_encode($texto);
44 }
45 }
46 else
47 {
48     return utf8_decode(join(", ", array("No hay registro de clientes con estos datos: ".$name.".")));
49 }
50 }
51 else
52 {
53     return "No hay registro de clientes con estos datos: ".$name.".";
54 }
55 }
56
57 $server = new soap_server();
58 $server->configureWSDL("cliente", "urn:cliente");
59
60 $server->register("getClientes",
61     array("name" => "xsd:string"),
62     array("return" => "xsd:string"),
63     "urn:cliente",
64     "urn:cliente#getClientes",
65     "rpc",
66     "encoded",
67     "Nos da una lista de clientes por nombre");
68
69 $server->service($HTTP_RAW_POST_DATA);
70 }>

```



CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

6.1.1. Primera conclusión

En la presente investigación se ha obtenido que después de realizar la interpretación y correcta discusión de todos los resultados del rendimiento de los servicios web, la mejor resultó ser REST.

6.1.2. Segunda conclusión

Como resultado de la presente investigación, se obtuvo que las herramientas más confiables para la validación del rendimiento era Valid WSDL para SOAP e Insomnia para REST.

6.1.3. Tercera conclusión

La presente investigación obtuvo mediante el análisis de los indicadores, que la mejor arquitectura para desarrollar servicios web es con REST.

6.1.4. Cuarta conclusión

El resultado para la presente investigación, nos indica que el valor económico obtenido de la propuesta es S/. 2,421.28 soles, cuyo monto deberá tener un retorno de inversión en 2 años, 10 meses y 17 días.



6.2. Recomendaciones

6.2.1. Primera recomendación

Recomiendo verificar algún código en caso existiera en algún navegador, también utilizar diferentes servicios a los presentados en esta investigación y así poder realizar la verificación de los resultados.

6.2.2. Segunda recomendación

Recomiendo realizar una prueba de la app con los distintos servicios para que pueda validar el análisis de los resultados expresados en esta investigación.

6.2.3. Tercera recomendación

Se recomienda considerar otros instrumentos de medición para realizar las pruebas de escalabilidad y rendimiento.

BIBLIOGRAFÍA

- IBM *Integration Bus*. (20 de 12 de 2016). Obtenido de https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.mft.doc/ac34750_.htm
- 5 herramientas gratuitas para medir la velocidad de carga de tu sitio web*. (s.f.). Obtenido de hipertextual.com
- API de servicio Web REST. (2016). *Servicio Web REST*. UNIVERSIDAD DE JYVÄSKYLÄ.
- Aransay, A. L. (2009). *Revisión de los Servicios Web SOAP/REST: Características y Rendimiento*.
- Arce, S. (2012). *Desarrollo e implementación de una API RESTful aplicada a un Sistema Académico Universitario*. Mendoza.
- Arlandy, M. (2014). Obtenido de <http://www.adictosaltrabajo.com/tutoriales/soavs-soap-rest/>
- Asúa, E. D. (17 de ABRIL de 2012). <http://movilforum.com>. Obtenido de <http://movilforum.com/apis-seguimos-con-el-rest-vs-soap/>
- Barbero., P. C. (2012). *SERVICIOS WEB BASADOS EN SOAP Y REST*. Lima.
- Brogden, W. (s.f.). "REST versus SOAP – the REST story". Obtenido de http://searchwebservices.techtarget.com/tip/0,289483,sid26_gci1227190,00.html
- Cabrera, Y. V. (2013). Transferencia de estado representacional (REST):. En Y. V. Cabrera.
- Carmona, P. (2014). "Plataformas De Integración: Servicios Web Basados En REST Y SOAP".
- Carmona, P. (s.f.). *PLATAFORMAS DE INTEGRACION SERVICIOS WEB BASADOS EN REST Y SOAP*.
- Castagnola, P. J. (2016). "Visualización de puntos de interés en un campus universitario usando realidad aumentada". LIMA.
- Chávez, M. A. (2010). Conceptos básicos de bases de datos.
- CHROME WEB STORE. (20 de 08 de 2016). Obtenido de <https://chrome.google.com/webstore/detail/insomnia-rest-client/gmodihnfibbjdecbanmpmbmeffnmloel?hl=es-419>
- Costello, R. L. (s.f.). "Building Web Services the REST Way" - xFront website. Obtenido de <http://www.xfront.com/REST-Web-Services.html>
- CRUZ, A. D. (2013). *UNA APROXIMACIÓN MDA PARA LA CONVERSIÓN*. Madrid.
- cruz, S. d. (2013). *UNA APROXIMACIÓN MDA PARA LA CONVERSIÓN*. Madrid.
- Cueli, J. V. (2009). *Implementacion e integracion de elementos software con tecnologias basadas en componentes*.



- Desarrollo de Software.* (2004). Obtenido de http://www.capacinet.gob.mx/Cursos/Tecnologia%20amiga/desarrolladoresoftware/ServiciosWeb_SE.pdf
- Dian Ayuba, A. I. (2013). "Evaluation of Page Response Time between Partial and Full Rendering in a Web-based Catalog System". *Procedia Technology* 11 (2013) 807 – 814.
- Fernández, N. C. (2012). VoCSDroid.
- Fernández, N. C. (2014). "Diseño de un Cliente para Dispositivos Moviles para un Sistema de Almacenamiento Voluntario en la Nube". ESPAÑA.
- Flores, C. A. (2012). Obtenido de <http://carlosmayta.blogspot.pe/>
- Florián, C. A. (2015). "IMPLEMENTACIÓN DE UNA APLICACIÓN MOVIL PARA EL MONITOREO DE CONTENIDO Y DISPONIBILIDAD DE SERVICIOS WEB, SERVICIO FTP Y PÁGINAS WEB". Lima.
- G., G. E., Campo, W. Y., Amaya, J. P., & Arciniegas, J. L. (2011). "Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON".
- Gonzales, A. (2010). *Diseño E Implementación De Un Servicio Web Que Permita Crear Y Consultar Un Vocabulario De La Lengua De Señas Y Visa gráfica Partiendo De Una Palabra Del Español.*
- González, M. A. (2010). *Diseño e implementacion de un servicio web que permita crear y consultar un vocabulario de la lengua de señas y visa grafica partiendo de una palabra del español.*
- Hevia, A. (18 de 02 de 2011). *WS SOAP versus REST en SOA.* Obtenido de <https://pensandoensoa.com/2011/02/18/ws-soap-versus-rest-en-soa/>
- Ladan, Z. (2015). *Comparing performance between plain JavaScript and popular JavaScript frameworks.*
- Lajo, J. O. (2011). *DESARROLLO DE UN SISTEMA DE CONSTRUCCIÓN DE WEBS 2.0.* Lima.
- Ledo, Y. (2007). *Simple Object Access Protocol.*
- Lima, J. C. (2012). *WEB SERVICES (SOAP X REST).* São Paulo.
- Marset, R. N. (2007). *REST VS WEB SERVICES.*
- Marset, R. N. (s.f.). *REST VS WEB SERVICES.*
- Mayta, c. (8 de diciembre de 2012). *SOAP VS REST.* Obtenido de <http://carlosmayta.blogspot.pe/>
- Medina, J. N. (2012). "DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE NOTIFICACIONES PARA EL CONTROL DEL ESTÁTUS ACADÉMICO ESTUDIANTIL".
- Miguel, A. (10 de enero de 2014). *adictosaltrabajo.* Obtenido de <https://www.adictosaltrabajo.com/tutoriales/soavs-soap-rest/>
- Nicolás, C. F. (2014). *Diseño de un cliente para dispositivos moviles para un sistema de almacenamiento voluntario en la nube.*



O'Reilly, T. (2007). *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*.

Palacios, C. Y., & Aguilar, P. P. (2015). *“Propuesta de implementación de un marco de trabajo para el desarrollo de aplicaciones ANDROID”*. LIMA.

Prescod, P. (s.f.). “Roots of the REST/SOAP Debate”.

programacion.jias.es. (2012).

Puebla, Y. A. (2009). *Propuesta de Arquitectura Orientada a Servicios para el Módulo de Inventario del ERP Cubano* Módulo de Inventario del ERP Cubano. Varadero.

Ramírez, E. (2008).

Requena, C. (2013). “Servicios web basados en REST o SOAP”.

Rey, C. S. (2010). *SISTEMAS INTEGRADOS DE GESTIÓN (ERP)*.

Rodríguez, J. (2003). *Sistema Informatico*.

Rojas, M. A. (2009). *Sistema de control de inventarios del almacén de productos terminados en una empresa metal mecánica*. LIMA.

Technology, J. (2016). Obtenido de <http://www.jc-mouse.net/proyectos/crea-un-servicio-web-rest-con-php-y-mysql>

Tyagi, S. (s.f.). “RESTful Web Services”.

www.ca.com/content/dam/ca/es/files/ebook/api-strategy-and-architecture-a-coordinated-approach.pdf. (s.f.).

Yan, L. (2011). *“Seguridad de Servicios Web Utilizando SOAP, WSDL y UDDI”*.

Zlatko, L. (2015). *Comparing performance between plain JavaScript and popular JavaScript frameworks*.

Facultad de tecnología de la información - Universidad de Tecnología, Sidney Australia
<https://navabautista.wikispaces.com/file/view/APLICACION+DE+LA+NORMA+ISO+9126.pdf>

ANEXOS

Anexo 01.- Matriz de consistencia

MATRIZ DE CONSISTENCIA

ANÁLISIS Y EVALUACIÓN DE LAS ARQUITECTURAS REST Y SOAP PARA EL DESARROLLO DE SERVICIOS WEB APLICADOS AL ERP "ADRISERP" Y SU VERSIÓN MÓVIL EN ANDROID.

Problema	Objetivos	Hipotesis	Variables	Indicadores
¿Qué arquitectura SOAP o REST, utilizaremos para el desarrollo de servicios web aplicados al módulo de ADRISMS del sistema ERP "AdrisERP"?	<p>1.- Objetivo general</p> <p>Analizar y evaluar las arquitecturas, REST y SOAP, para el desarrollo de servicios web del módulo ADRISMS del sistema ERP "AdrisERP".</p> <p>2.- Objetivos específicos</p> <p>A. Analizar cuál es la arquitectura más óptima para el desarrollo de servicios web.</p> <p>B. Evaluar las herramientas que se utilizaran para las mediciones de las arquitecturas.</p> <p>C. Analizar los resultados de las mediciones de los indicadores</p> <p>D. Realizar la evaluación económica de la propuesta.</p>	<p>Mediante el análisis y la evaluación de los indicadores de rendimiento y escalabilidad de las arquitecturas "Soap o Rest", permitirán encontrar cuál es la más óptima para el desarrollo de servicios web del módulo ADRISMS del sistema ERP "AdrisERP".</p>	<p>1.- Independiente</p> <p>a) La escalabilidad, (Nivel de calidad del servicio web).</p> <p>b) El rendimiento de las arquitecturas, (tiempo promedio de carga del servicio web).</p> <p>2.- Dependiente</p> <p>a) Curva de aprendizaje.</p> <p>b) Tiempo de desarrollo por cada servicio web utilizando las arquitecturas REST y SOAP.</p>	<p>El rendimiento</p> <p>La escalabilidad</p> <p>Curva de aprendizaje</p> <p>Tiempo de desarrollo</p>

Tabla 26. Matriz de consistencia



Anexo 02.- Instrumento de recolección de datos

Instrumento de recolección de datos

TÉCNICA	USO	INSTRUMENTO
Análisis Documental	Usaremos documentos de investigaciones realizadas por otras personas de forma que nos permita saber un poco mas a fondo de las diferentes arquitecturas "SOAT o REST" para asi saber cual de ella deberemos escoger para empezar con el desarrollo de la version movil de ADRISERP	Ficha de registro de datos
Observación	Haremos las observaciones correspondientes mediante las fichas de observacion para asi poder determinar si el sistemas es: FIABLE, ROBUSTO Y SEGURO	Ficha de observación
Juicio de expertos	Haremos las observaciones correspondientes a la version de ADRISERP que ya se tiene para que con ello podamos ir descubriendo como es que podemos desarrollar su version movil	Planilla de Juicio de Expertos

Tabla 27. Instrumento de recolección de datos



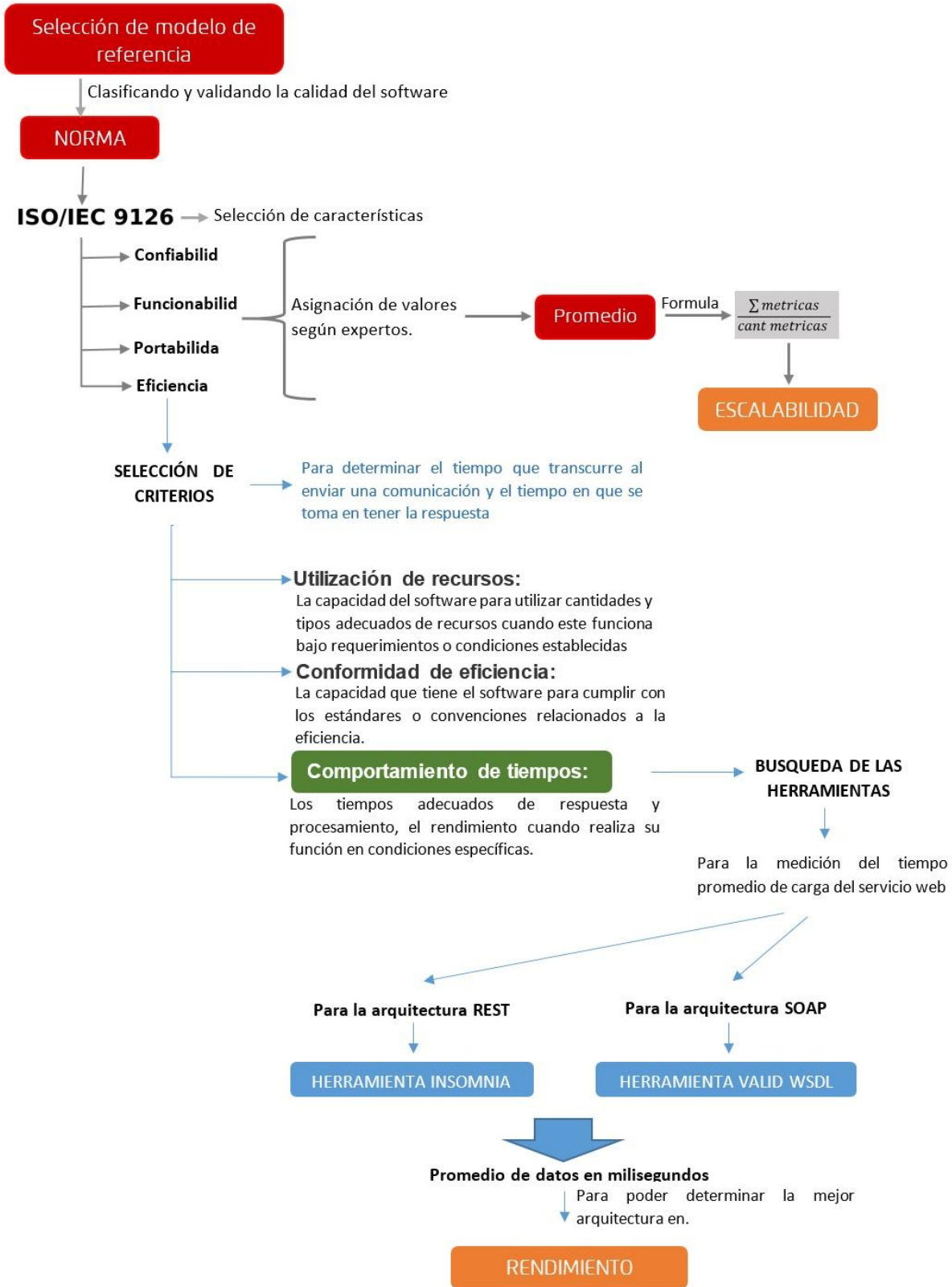
Anexo 03: Estructura de la Propuesta de Solución



Tabla 28. Datos Estructura de la Propuesta de Solución.



Anexo 04: Sistematización de rendimiento y escalabilidad



Anexo 05: Matriz de consistencia para la selección de características

TESIS	ANÁLISIS Y EVALUACIÓN DE LAS ARQUITECTURAS REST Y SOAP PARA EL DESARROLLO DE SERVICIOS WEB APLICADOS AL ERP "ADRISERP" Y SU VERSIÓN MÓVIL EN ANDROID				UNIDAD DE MEDIDA	METRICAS DE EVALUACION								
VARIABLE INDEPENDIENTE	ESCALABILIDAD	DEFINICION CONCEPTUAL Se define como el problema que surge al crecer la aplicación y el tiempo que este demanda en recurso de personal en conocimiento y desarrollo.	INDICADOR Nivel de calidad del servicio web	INSTRUMENTO ISO/IEC 9126	FORMULA $\frac{\sum \text{metricas}}{\text{cant metricas}}$	<p>Basado en: (Facultad de tecnología de la información - Universidad de Tecnología, Sidney Australia)</p> <table border="1"> <tr> <td>CONFIABILIDAD</td> <td>PORTABILIDAD</td> <td>FUNCIONABILIDAD</td> <td>EFICIENCIA</td> </tr> <tr> <td>Satisfactoria 10 Insatisfactoria 0</td> <td>Si 10 No 0</td> <td>Cumple 10 Aceptable 7 Minima 4 Inaceptable 0</td> <td></td> </tr> </table>	CONFIABILIDAD	PORTABILIDAD	FUNCIONABILIDAD	EFICIENCIA	Satisfactoria 10 Insatisfactoria 0	Si 10 No 0	Cumple 10 Aceptable 7 Minima 4 Inaceptable 0	
	CONFIABILIDAD	PORTABILIDAD	FUNCIONABILIDAD	EFICIENCIA										
Satisfactoria 10 Insatisfactoria 0	Si 10 No 0	Cumple 10 Aceptable 7 Minima 4 Inaceptable 0												
RENDIMIENTO	Está definido como aquel tiempo que transcurre al enviar una comunicación y el tiempo en que se toma en tener la respuesta.	Tiempo promedio de carga del servicio web	ISO/IEC 9126	<table border="1"> <tr> <td>LISTAR</td> <td>INSERTAR</td> <td>BURCAR</td> </tr> <tr> <td colspan="3">Herramienta Insomnia Herramienta Valid WSDL</td> </tr> </table>	LISTAR	INSERTAR	BURCAR	Herramienta Insomnia Herramienta Valid WSDL					
LISTAR	INSERTAR	BURCAR												
Herramienta Insomnia Herramienta Valid WSDL														
VARIABLE DEPENDIENTE	LA MEJOR ARQUITECTURA SOAP O REST.	La arquitectura SOAP es soportada por los desarrolladores de software y los analistas.	Curva de aprendizaje.	Juicio de Expertos	$\frac{E1+E2+E3+E4}{4}$	<table border="1"> <tr> <td>Fase de Análisis (tiempo comprensión)</td> <td>Fase de Construcción (desarrollo)</td> </tr> <tr> <td>Excelente 10 Muy Bueno 7 - 9 Bueno 4 - 6 Mejorable 1 - 3</td> <td></td> </tr> </table>	Fase de Análisis (tiempo comprensión)	Fase de Construcción (desarrollo)	Excelente 10 Muy Bueno 7 - 9 Bueno 4 - 6 Mejorable 1 - 3					
		Fase de Análisis (tiempo comprensión)	Fase de Construcción (desarrollo)											
Excelente 10 Muy Bueno 7 - 9 Bueno 4 - 6 Mejorable 1 - 3														
REST intenta imitar el protocolo HTTP y fijar la interfaz en un grupo de trabajos estándares.	Tiempo de desarrollo por cada servicio web utilizando las arquitecturas RESTY SOAP		<table border="1"> <tr> <td>Fase de Planificación</td> <td>Fase de Análisis (tiempo comprensión)</td> <td>Fase de Construcción (desarrollo)</td> </tr> <tr> <td>Excelente 10 Muy Bueno 7 - 9 Bueno 4 - 6 Mejorable 1 - 3</td> <td></td> <td></td> </tr> </table>	Fase de Planificación	Fase de Análisis (tiempo comprensión)	Fase de Construcción (desarrollo)	Excelente 10 Muy Bueno 7 - 9 Bueno 4 - 6 Mejorable 1 - 3							
Fase de Planificación	Fase de Análisis (tiempo comprensión)	Fase de Construcción (desarrollo)												
Excelente 10 Muy Bueno 7 - 9 Bueno 4 - 6 Mejorable 1 - 3														

(Facultad de tecnología de la información - Universidad de Tecnología, Sidney Australia)



Anexo 06: Carta de compromiso de los expertos

EXPERTO DE LA CARRERA DE INGENIERIA DE SISTEMAS

NOMBRE : ING. JUAN JINER DELGADO CHAVEZ

CARGO : GERENTE GENERAL AD Y L CONSULTING

Me comprometo a evaluar con total honestidad el trabajo desarrollado en esta presente investigación "Análisis y evaluación de las arquitecturas Rest y Soap para el desarrollo de servicios web aplicados al Erp "Adriserp" y su versión móvil en Android"; donde apelare a mi experiencia en la elaboración de sistemas de información, para emitir con criterio adecuado el puntaje necesario para la respectiva evaluación.

AD y L Consulting S.A.C.

Ing. Juan Jiner Delgado Chávez
GERENTE GENERAL

FIRMA

EXPERTO DE LA CARRERA DE INGENIERIA DE SISTEMAS

NOMBRE : ING. MIGUEL DÍAZ VIDARTE

CARGO : JEFE DEL AREA DE TECNOLOGIA DE INFORMACIÓN
EN INTIHOTEL

Me comprometo a evaluar con total honestidad el trabajo desarrollado en esta presente investigación "Análisis y evaluación de las arquitecturas Rest y Soap para el desarrollo de servicios web aplicados al Erp "Adriserp" y su versión móvil en Android"; donde apelare a mi experiencia en la elaboración de sistemas de información, para emitir con criterio adecuado el puntaje necesario para la respectiva evaluación.

FIRMA

EXPERTO DE LA CARRERA DE INGENIERIA DE SISTEMAS

NOMBRE : ING. MIGUEL ANGEL VIDAURRE

CARGO : DOCENTE EN LA UNIVERSIDAD SEÑOR DE SIPAN

Me comprometo a evaluar con total honestidad el trabajo desarrollado en esta presente investigación "Análisis y evaluación de las arquitecturas Rest y Soap para el desarrollo de servicios web aplicados al Erp "Adriserp" y su versión móvil en Android"; donde apelare a mi experiencia en la elaboración de sistemas de información, para emitir con criterio adecuado el puntaje necesario para la respectiva evaluación.



FIRMA

EXPERTO DE LA CARRERA DE INGENIERIA DE SISTEMAS

NOMBRE : ING. ALEX MURO NUÑEZ

CARGO : DIRECTOR DE TECNOLOGIAS EN UNIVERSIDAD
SEÑOR DE SIPAN

Me comprometo a evaluar con total honestidad el trabajo desarrollado en esta presente investigación "Análisis y evaluación de las arquitecturas Rest y Soap para el desarrollo de servicios web aplicados al Erp "Adriserp" y su versión móvil en Android"; donde apelare a mi experiencia en la elaboración de sistemas de información, para emitir con criterio adecuado el puntaje necesario para la respectiva evaluación.

FIRMA

Anexo 07: Tabla de medición por experto – Variable independiente

TABLA DE MEDICIÓN DE CALIDAD DEL SERVICIO WEB			
EXPERTO	Juan Jiner Delgado Chávez		
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición</p> <p>CONFIABILIDAD: Satisfactoria 10 Insatisfactoria 0</p> <p>PORTABILIDAD: Si 10 No 0</p> <p>FUNCIONALIDAD/EFICIENCIA: Cumple 10 Aceptable 7 Mínima 4 Inaceptable 0</p>			
CARACTERISTICA	METRICA	PUNTAJE ARQ. SOAP	PUNTAJE ARQ. REST
FUNCIONALIDAD	(Cumple,Aceptable,Minima,Inaceptable)	4	7
CONFIABILIDAD	(Satisfactoria, insatisfactoria)	10	10
EFICIENCIA	(Cumple,Aceptable,Minima,Inaceptable)	10	10
PORTABILIDAD	(Si, No)	0	10

Tabla 29. Tabla del Experto 01



TABLA DE MEDICIÓN DE CALIDAD DEL SERVICIO WEB			
EXPERTO		Miguel Angel Vidaurre	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición</p> <p>CONFIABILIDAD: Satisfactoria 10 Insatisfactoria 0</p> <p>PORTABILIDAD: Si 10 No 0</p> <p>FUNCIONALIDAD/EFICIENCIA: Cumple 10 Aceptable 7 Minima 4 Inaceptable 0</p>			
CARACTERISTICA	METRICA	PUNTAJE ARQ. SOAP	PUNTAJE ARQ. REST
FUNCIONALIDAD	(Cumple,Aceptable,Minima,Inaceptable)	7	10
CONFIABILIDAD	(Satisfactoria, insatisfactoria)	10	10
EFICIENCIA	(Cumple,Aceptable,Minima,Inaceptable)	7	10
PORTABILIDAD	(Si, No)	0	10

Tabla 30. Tabla del Experto 02



TABLA DE MEDICIÓN DE CALIDAD DEL SERVICIO WEB			
EXPERTO		Miguel Díaz Vidarte	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición</p> <p>CONFIABILIDAD: Satisfactoria 10 Insatisfactoria 0</p> <p>PORTABILIDAD: Si 10 No 0</p> <p>FUNCIONALIDAD/EFICIENCIA: Cumple 10 Aceptable 7 Minima 4 Inaceptable 0</p>			
CARACTERISTICA	METRICA	PUNTAJE ARQ. SOAP	PUNTAJE ARQ. REST
FUNCIONALIDAD	(Cumple,Aceptable,Minima,Inaceptable)	10	10
CONFIABILIDAD	(Satisfactoria, insatisfactoria)	0	10
EFICIENCIA	(Cumple,Aceptable,Minima,Inaceptable)	10	7
PORTABILIDAD	(Si, No)	10	10

Tabla 31. Tabla del Experto 03



TABLA DE MEDICIÓN DE CALIDAD DEL SERVICIO WEB			
EXPERTO		Alex Muro Nuñez	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición</p> <p>CONFIABILIDAD: Satisfactoria 10 Insatisfactoria 0</p> <p>PORTABILIDAD: Si 10 No 0</p> <p>FUNCIONALIDAD/EFICIENCIA: Cumple 10 Aceptable 7 Mínima 4 Inaceptable 0</p>			
CARACTERISTICA	METRICA	PUNTAJE ARQ. SOAP	PUNTAJE ARQ. REST
FUNCIONALIDAD	(Cumple, Aceptable, Mínima, Inaceptable)	7	4
CONFIABILIDAD	(Satisfactoria, insatisfactoria)	10	0
EFICIENCIA	(Cumple, Aceptable, Mínima, Inaceptable)	7	10
PORTABILIDAD	(Si, No)	0	10

Tabla 32. Tabla del Experto 04

Anexo 08: Juicio de experto - Variable dependiente

PLANILLA DE JUICIO DE EXPERTOS			
NOMBRE DEL EXPERTO		Juan Jiner Delgado Chávez	
ARQUITECTURA		SOAP	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:</p> <p>EXCELENTE: 10 MUY BUENO: 7 - 9 BUENO: 4 - 6 MEJORABLE: 1 - 3</p>			
	ATRIBUTOS DEL MODELO		
I	CURVA DE APRENDIZAJE	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Análisis (tiempo comprensión)	4	10
	2. Fase de Construcción (desarrollo)	11	10
	PROMEDIO TOTAL POR EXPERTO:		10
II	TIEMPO DE DESARROLLO	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Planificación	5	8
	2. Fase de Análisis	3	10
	3. Fase de Construcción	12	10
	PROMEDIO TOTAL POR EXPERTO:		9.3

Tabla 33. Planilla de la arquitectura SOAP del Experto 01



PLANILLA DE JUICIO DE EXPERTOS			
NOMBRE DEL EXPERTO		Miguel Angel Vidaurre	
ARQUITECTURA		SOAP	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:</p> <p>EXCELENTE: 10 MUY BUENO: 7 - 9 BUENO: 4 - 6 MEJORABLE: 1 - 3</p>			
ATRIBUTOS DEL MODELO			
I	CURVA DE APRENDIZAJE	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Análisis (tiempo comprensión)	4	9
	2. Fase de Construcción (desarrollo)	11	8
PROMEDIO TOTAL POR EXPERTO:			8.5
II	TIEMPO DE DESARROLLO	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Planificación	5	9
	2. Fase de Análisis	3	9
	3. Fase de Construcción	12	9
PROMEDIO TOTAL POR EXPERTO:			9

Tabla 344. Planilla de la arquitectura SOAP del Experto 02



PLANILLA DE JUICIO DE EXPERTOS			
NOMBRE DEL EXPERTO		Miguel Angel Vidaurre	
ARQUITECTURA		REST	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:</p> <p>EXCELENTE: 10 MUY BUENO: 7 - 9 BUENO: 4 - 6 MEJORABLE: 1 - 3</p>			
ATRIBUTOS DEL MODELO			
I	CURVA DE APRENDIZAJE	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Análisis (tiempo comprensión)	3	9
	2. Fase de Construcción (desarrollo)	10	9
PROMEDIO TOTAL POR EXPERTO:			9
II	TIEMPO DE DESARROLLO	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Planificación	5	8
	2. Fase de Análisis	2	9
	3. Fase de Construcción	10	9
PROMEDIO TOTAL POR EXPERTO:			8.6

Tabla 35. Planilla de la arquitectura SOAP del Experto 03



PLANILLA DE JUICIO DE EXPERTOS			
NOMBRE DEL EXPERTO		Miguel Díaz Vidarte	
ARQUITECTURA		SOAP	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:</p> <p>EXCELENTE: 10 MUY BUENO: 7 - 9 BUENO: 4 - 6 MEJORABLE: 1 - 3</p>			
ATRIBUTOS DEL MODELO			
I	CURVA DE APRENDIZAJE	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Análisis (tiempo comprensión)	4	10
	2. Fase de Construcción (desarrollo)	11	10
PROMEDIO TOTAL POR EXPERTO:			10
II	TIEMPO DE DESARROLLO	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Planificación	5	9
	2. Fase de Análisis	3	10
	3. Fase de Construcción	12	10
PROMEDIO TOTAL POR EXPERTO:			9.6

Tabla 36. Planilla de la arquitectura SOAP del Experto 03



PLANILLA DE JUICIO DE EXPERTOS			
NOMBRE DEL EXPERTO		Miguel Díaz Vidarte	
ARQUITECTURA		REST	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:</p> <p>EXCELENTE: 10 MUY BUENO: 7 - 9 BUENO: 4 - 6 MEJORABLE: 1 - 3</p>			
ATRIBUTOS DEL MODELO			
I	CURVA DE APRENDIZAJE	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Análisis (tiempo comprensión)	3	10
	2. Fase de Construcción (desarrollo)	10	10
PROMEDIO TOTAL POR EXPERTO:			10
II	TIEMPO DE DESARROLLO	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Planificación	5	10
	2. Fase de Análisis	2	10
	3. Fase de Construcción	10	10
PROMEDIO TOTAL POR EXPERTO:			10

Tabla 37. Planilla de la arquitectura REST del Experto 03



PLANILLA DE JUICIO DE EXPERTOS			
NOMBRE DEL EXPERTO		Alex Muro Nuñez	
ARQUITECTURA		SOAP	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:</p> <p>EXCELENTE: 10 MUY BUENO: 7 - 9 BUENO: 4 - 6 MEJORABLE: 1 - 3</p>			
ATRIBUTOS DEL MODELO			
I	CURVA DE APRENDIZAJE	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Análisis (tiempo comprensión)	4	8
	2. Fase de Construcción (desarrollo)	11	10
PROMEDIO TOTAL POR EXPERTO:			9
II	TIEMPO DE DESARROLLO	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Planificación	5	9
	2. Fase de Análisis	3	10
	3. Fase de Construcción	12	8
PROMEDIO TOTAL POR EXPERTO:			9

Tabla 38. Planilla de la arquitectura SOAP del Experto 04

PLANILLA DE JUICIO DE EXPERTOS			
NOMBRE DEL EXPERTO		Alex Muro Nuñez	
ARQUITECTURA		REST	
<p>Las siguientes métricas serán consideradas dentro de la evaluación, las cuales tienen asignado un valor de medición:</p> <p>EXCELENTE: 10 MUY BUENO: 7 - 9 BUENO: 4 - 6 MEJORABLE: 1 - 3</p>			
	ATRIBUTOS DEL MODELO		
I	CURVA DE APRENDIZAJE	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Análisis (tiempo comprensión)	3	10
	2. Fase de Construcción (desarrollo)	10	10
	PROMEDIO TOTAL POR EXPERTO:		10
II	TIEMPO DE DESARROLLO	TIEMPO/ DIAS	PUNTAJE
	1. Fase de Planificación	5	9
	2. Fase de Análisis	2	9
	3. Fase de Construcción	10	9
	PROMEDIO TOTAL POR EXPERTO:		8.6

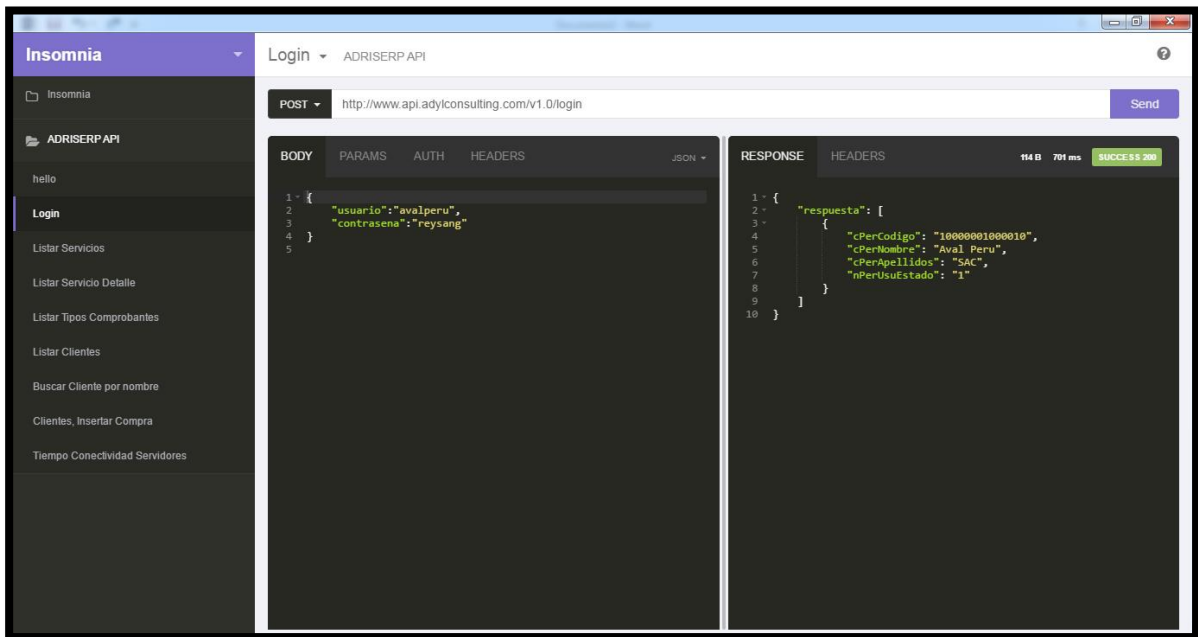
Tabla 39. Planilla de la arquitectura REST del Experto 04



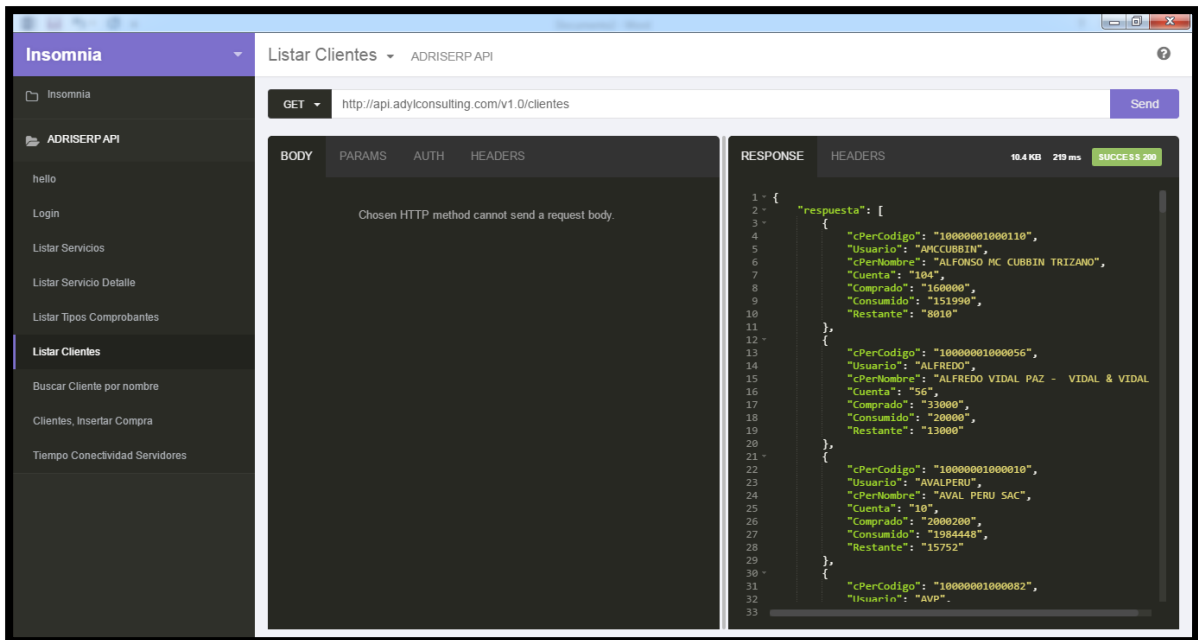
Anexo 09: Resultados de pruebas de rendimiento de arquitecturas REST y SOAP.

A) Servicios de la Arquitectura REST – HERRAMIENTA INSOMNIA

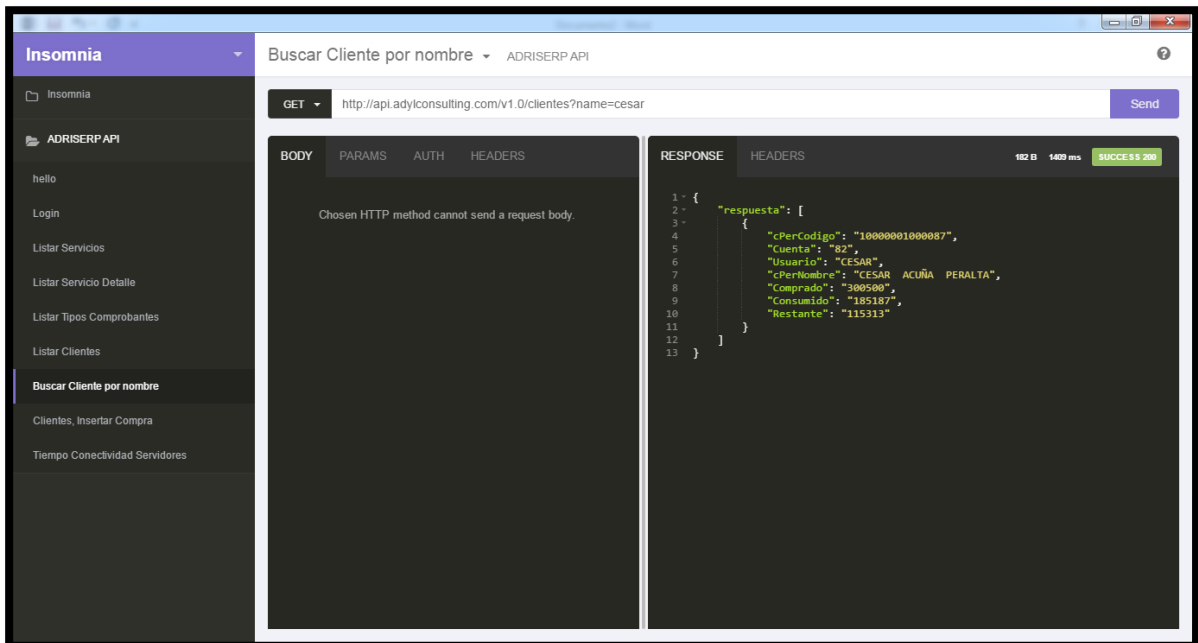
- *Login*



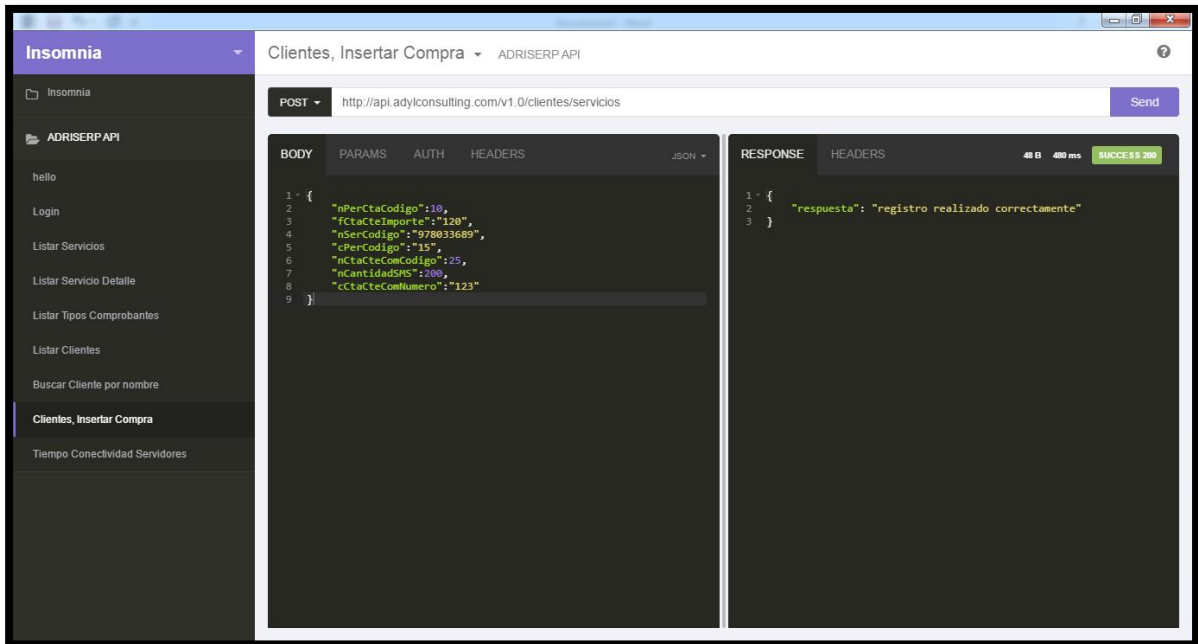
- Listar Clientes



- Buscar Clientes

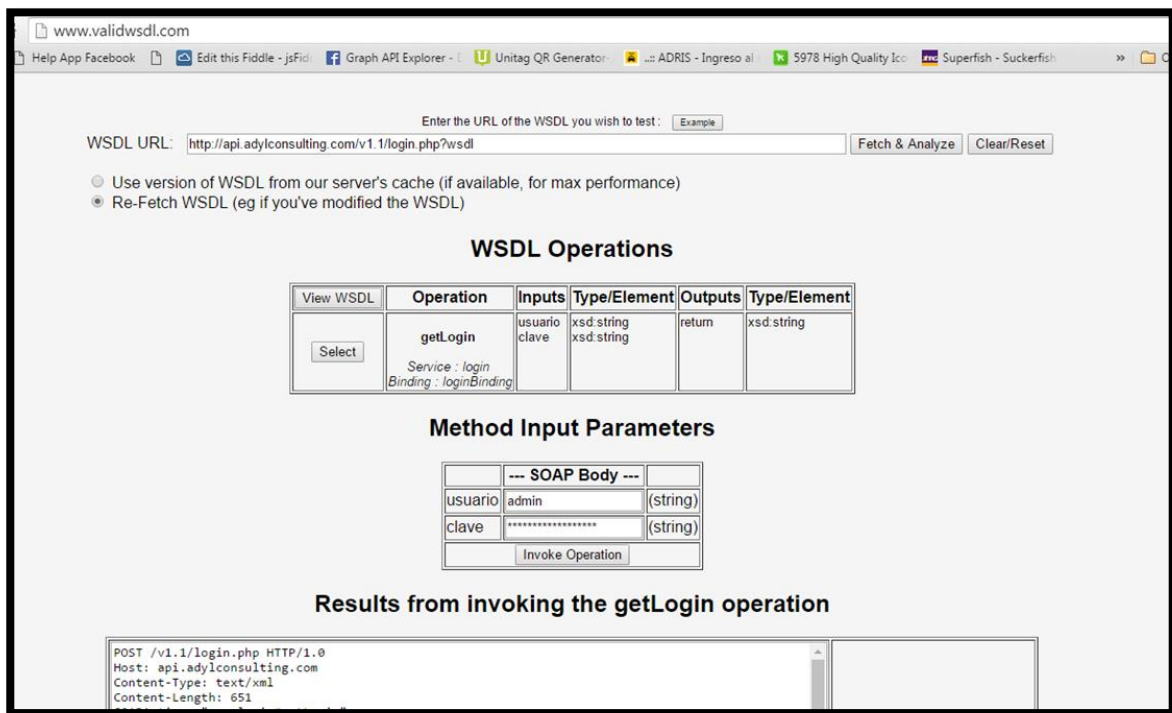


- Registrar Venta



B) Servicios de la Arquitectura SOAP – HERRAMIENTAVALID WSDL

- Login



- Listar Clientes

Enter the URL of the WSDL you wish to test:

WSDL URL:

Use version of WSDL from our server's cache (if available, for max performance)
 Re-Fetch WSDL (eg if you've modified the WSDL)

WSDL Operations

	Operation	Inputs	Type/Element	Outputs	Type/Element
<input type="button" value="Select"/>	getClientes <small>Service : cliente Binding : clienteBinding</small>	name	xsd:string	return	xsd:string

Method Input Parameters

	--- SOAP Body ---	
name	<input type="text"/>	(string)
<input type="button" value="Invoke Operation"/>		

Results from invoking the getClientes operation

<pre>POST /v1.1/cliente.php HTTP/1.0 Host: api.adylconsulting.com Content-Type: text/xml Content-Length: 578 SOAPAction: "urn:cliente#getClientes" <?xml version='1.0'?> <SOAP-ENV:Envelope SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/' xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/' xmlns:tns='urn:cliente' xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'> <SOAP-ENV:Body> <ns1:getClientes SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' xmlns:ns1='urn:cliente'> <name xsi:type='xsd:string' /> </ns1:getClientes> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>	Outgoing SOAP Request
<pre><?xml version='1.0' encoding='ISO-8859-1'?><SOAP-ENV:Envelope SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'><SOAP-ENV:Body><ns1:getClientesResponse xmlns:ns1='urn:cliente'><return xsi:type='xsd:string'[10000001000110 - AMCCUBBIN - ALFONSO MC CUBBIN TRIZANO - 104 - 209500 - 207479 - 2021,10000001000056 - ALFREDO - ALFREDO VIDAL PAZ - VIDAL & VIDAL E.I.R.L.T - 56 - 48000 - 47961 - 39,10000001000010 - AVALPERU - AVAL PERU SAC - 10 - 2000800 - 1565911 - 434889,10000001000082 - AVP - AVP LUDWIG VIGO VIGO - 77 - 245000 - 215612 - 29388,10000001000101 - BBP - BBP BBP BBP - 95 - 210000 - 189819 - 20181,10000001000100 - BFA - BFA BFA BFA - 94 - 465000 - 444394 - 20606,10000001000137 - BFN - BFN BFN BFN - 131 - 60000 - 34356 - 25644,10000001000051 - CCPERU - C&C CONSULTORES Y GESTORES EMPRESARIALES S.A.C - 51 - 360000 - 324387 - 35613,10000001000136 - CAMPANAPRUEBA - CAMPANA PRUEBA SMS - 130 - 100 - 75 - 25,10000001000057 - CARLOS - CARLOS PANTA NICO - VIDAL & VIDAL E.I.R.L.T - 57 - 34000 - 33908 - 92,10000001000087 - CESAR - CESAR ACUÑA PERALTA - 82 - 300500 - 185187 - 115313,100000505993 - FILIBERTOCUEVA@GMAIL.COM - CHIQUITIN SALAZAR GARCIA - 209996673 - 2000 - 1857 - 143,10000001000048 - CHRISTIAN - CHRISTIAN VIDAL PAZ - VIDAL & VIDAL E.I.R.L.T - 48 - 290000 - 249296 - 40704,10000001000076 - CLAUDIA.AGREDA@AVON.COM - CLAUDIA AGREDA AVON - COBRANZAS - 71 - 50000 - 45783 - 4217,10000001000105 - CLC - CLC CLC - 99 - 332000 - 311629 - 20371,10000001000080 - CLP - CLP LUDWIG VIGO VIGO - 75 - 1110122 - 1009050 - 101072,10000001000045 - DENISSE - DENISSE PAOLA SALDANA GASTULO - 45 - 45000 - 39189 - 5811,10000001000096 - DIEGO - DIEGO ROMERO R - 91 - 10000 - 9999 - 1,100000511260 -</pre>	Incoming SOAP Response response time .402299 seconds



- *Buscar Clientes*

Enter the URL of the WSDL you wish to test:

WSDL URL:

Use version of WSDL from our server's cache (if available, for max performance)
 Re-Fetch WSDL (eg if you've modified the WSDL)

WSDL Operations

View WSDL	Operation	Inputs	Type/Element	Outputs	Type/Element
<input type="button" value="Select"/>	getClientes Service : cliente Binding : clienteBinding	name	xsd:string	return	xsd:string

Method Input Parameters

--- SOAP Body ---	
name	Aval (string)
<input type="button" value="Invoke Operation"/>	

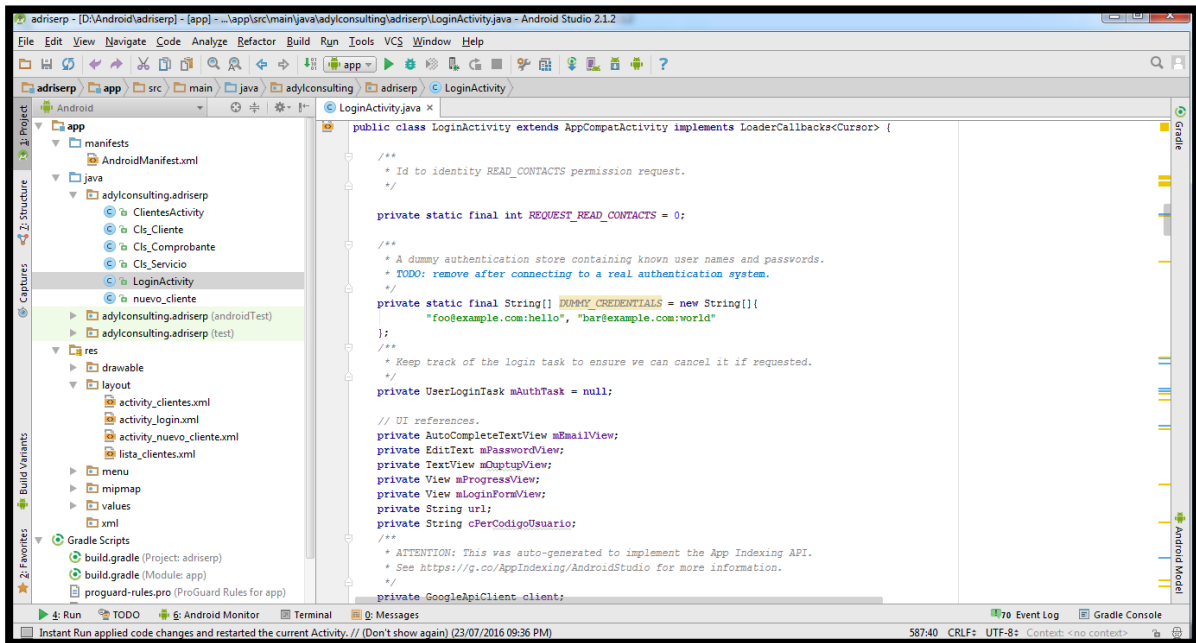
Results from invoking the getClientes operation

<pre>POST /v1.1/cliente.php HTTP/1.0 Host: api.adylconsulting.com Content-Type: text/xml Content-Length: 613 SOAPAction: "urn:cliente#getClientes" <?xml version='1.0'?> <SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/' xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/' xmlns:tns='urn:cliente' xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'> <SOAP-ENV:Body> <ns1:getClientes SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' xmlns:ns1='urn:cliente'> <name xsi:type='xsd:string'> Aval </name> </ns1:getClientes> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>	<p>Outgoing SOAP Request</p>
<pre><?xml version='1.0' encoding='ISO-8859-1'?> <SOAP-ENV:Envelope SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/' xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'> <SOAP-ENV:Body> <ns1:getClientesResponse xmlns:ns1='urn:cliente'> <return xsi:type='xsd:string'> [100000010000010 - 10 - AVALPERU - AVAL PERU SAC - 2000800 - 1565911 - 434889] </return> </ns1:getClientesResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>	<p>Incoming SOAP Response</p> <p>response time 1.782953 seconds</p>



Anexo 10: Consumo del servicio desde app en Android con la arquitectura más óptima “REST”

- Login



```

public class LoginActivity extends AppCompatActivity implements LoaderCallbacks<Cursor> {

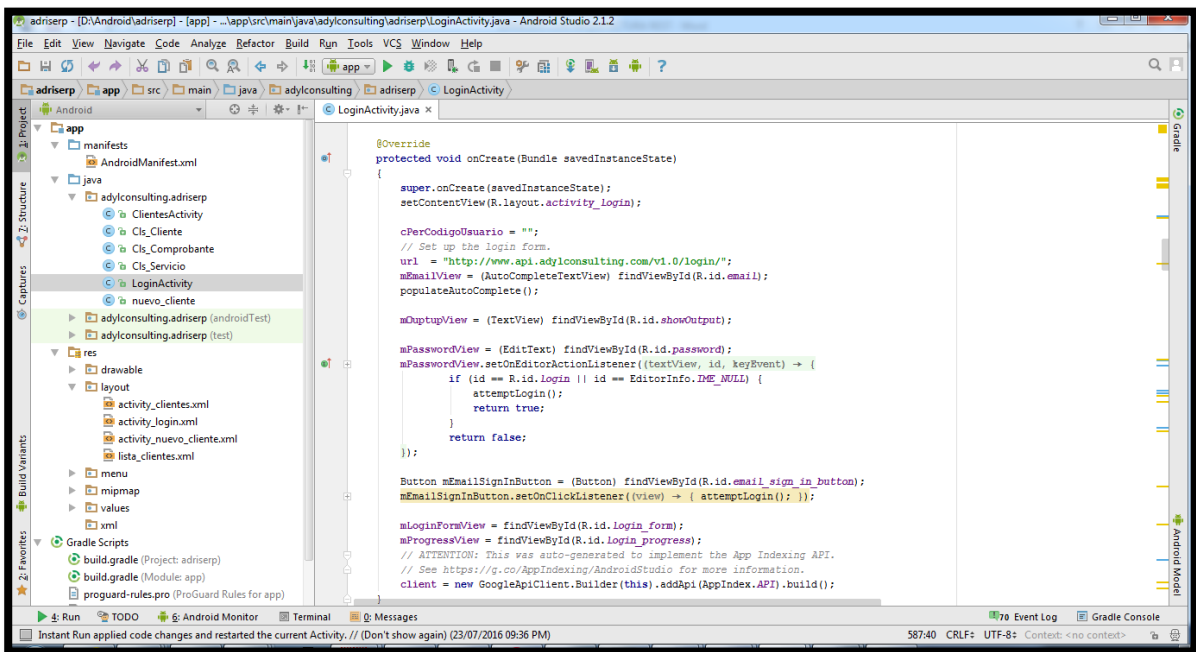
    /**
     * Id to identity READ_CONTACTS permission request.
     */
    private static final int REQUEST_READ_CONTACTS = 0;

    /**
     * A dummy authentication store containing known user names and passwords.
     * TODO: remove after connecting to a real authentication system.
     */
    private static final String[] DUMMY_CREDENTIALS = new String[]{
        "foo@example.com:hello", "bar@example.com:world"
    };

    /**
     * Keep track of the login task to ensure we can cancel it if requested.
     */
    private UserLoginTask mAuthTask = null;

    // UI references.
    private autoCompleteTextView mEmailView;
    private EditText mPasswordView;
    private TextView mOutputView;
    private View mProgressBar;
    private View mLoginFormView;
    private String url;
    private String cPerCodigoUsuario;

    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;
    
```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    cPerCodigoUsuario = "";
    // Set up the login form.
    url = "http://www.api.adylconsulting.com/v1.0/login/";
    mEmailView = (AutoCompleteTextView) findViewById(R.id.email);
    populateAutoComplete();

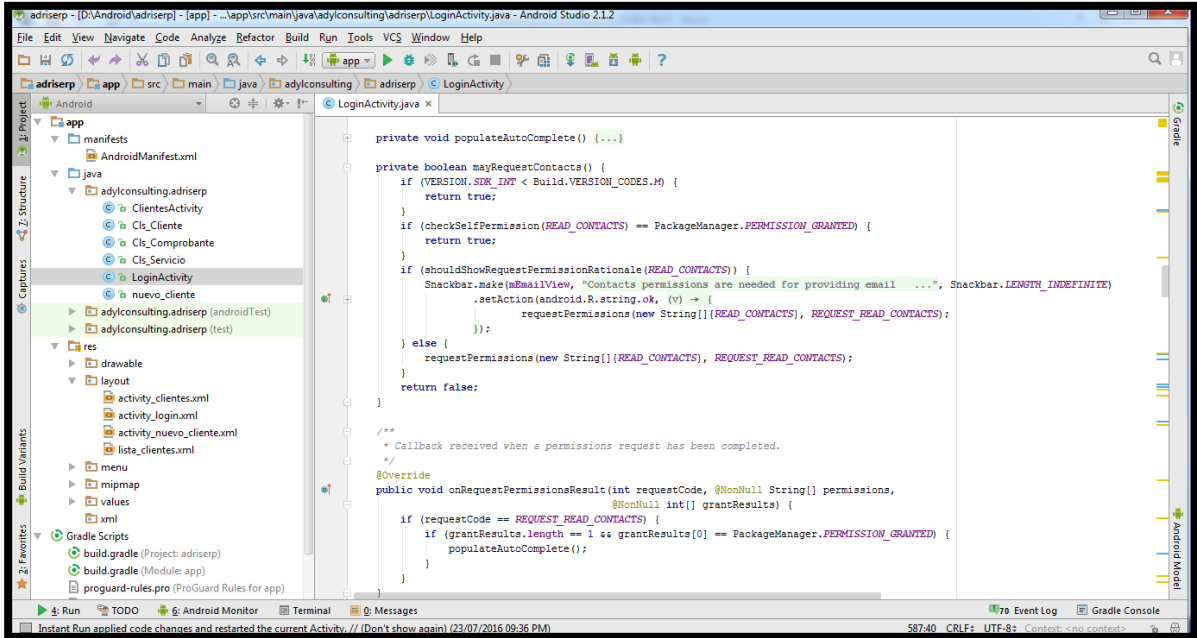
    mOutputView = (TextView) findViewById(R.id.showOutput);

    mPasswordView = (EditText) findViewById(R.id.password);
    mPasswordView.setOnEditorActionListener((textView, id, keyEvent) -> {
        if (id == R.id.login || id == EditorInfo.IME_NULL) {
            attemptLogin();
            return true;
        }
        return false;
    });

    Button mEmailSignInButton = (Button) findViewById(R.id.email_sign_in_button);
    mEmailSignInButton.setOnClickListener((view) -> { attemptLogin(); });

    mLoginFormView = findViewById(R.id.login_form);
    mProgressBar = findViewById(R.id.login_progress);
    // ATTENTION: This was auto-generated to implement the App Indexing API.
    // See https://g.co/AppIndexing/AndroidStudio for more information.
    client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
    
```



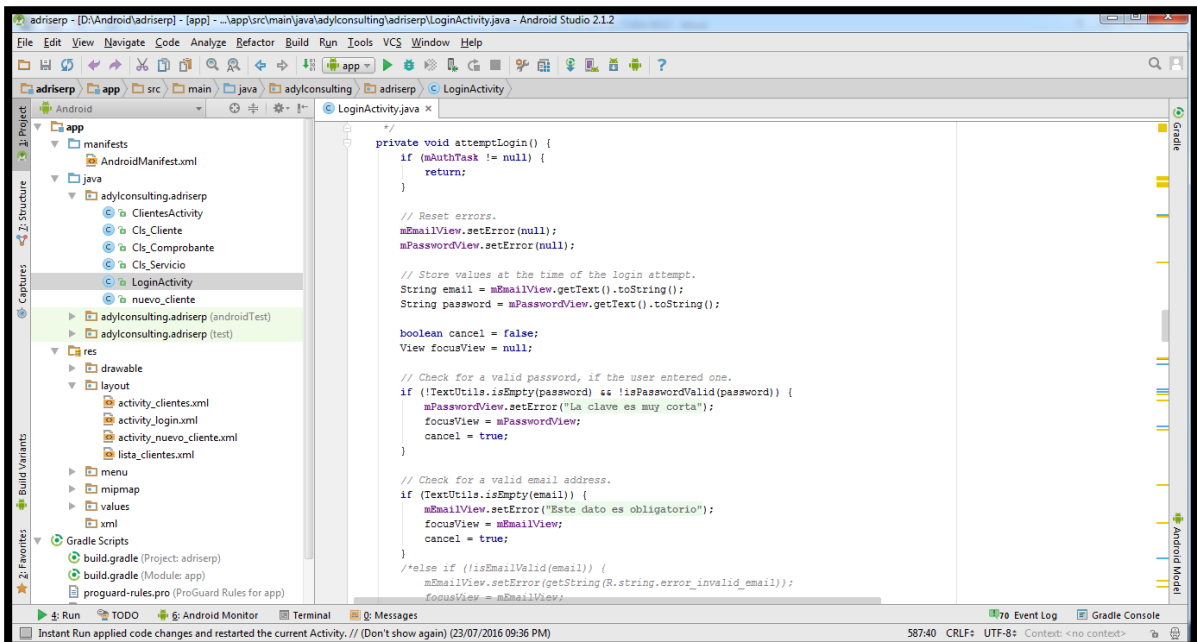


```

private void populateAutoComplete() {...}

private boolean mayRequestContacts() {
    if (VERSION.SDK_INT < Build.VERSION_CODES.M) {
        return true;
    }
    if (checkSelfPermission(READ_CONTACTS) == PackageManager.PERMISSION_GRANTED) {
        return true;
    }
    if (shouldShowRequestPermissionRationale(READ_CONTACTS)) {
        Snackbar.make(mEmailView, "Contacts permissions are needed for providing email ...", Snackbar.LENGTH_INDEFINITE)
            .setAction(android.R.string.ok, (v) -> {
                requestPermissions(new String[]{READ_CONTACTS}, REQUEST_READ_CONTACTS);
            });
    } else {
        requestPermissions(new String[]{READ_CONTACTS}, REQUEST_READ_CONTACTS);
    }
    return false;
}

/**
 * Callback received when a permissions request has been completed.
 */
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
    @NonNull int[] grantResults) {
    if (requestCode == REQUEST_READ_CONTACTS) {
        if (grantResults.length == 1 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            populateAutoComplete();
        }
    }
}
    
```



```

private void attemptLogin() {
    if (mAuthTask != null) {
        return;
    }

    // Reset errors.
    mEmailView.setError(null);
    mPasswordView.setError(null);

    // Store values at the time of the login attempt.
    String email = mEmailView.getText().toString();
    String password = mPasswordView.getText().toString();

    boolean cancel = false;
    View focusView = null;

    // Check for a valid password, if the user entered one.
    if (!TextUtils.isEmpty(password) && !isPasswordValid(password)) {
        mPasswordView.setError("La clave es muy corta");
        focusView = mPasswordView;
        cancel = true;
    }

    // Check for a valid email address.
    if (TextUtils.isEmpty(email)) {
        mEmailView.setError("Este dato es obligatorio");
        focusView = mEmailView;
        cancel = true;
    }
    } else if (!isEmailValid(email)) {
        mEmailView.setError(getString(R.string.error_invalid_email));
        focusView = mEmailView;
    }
}
    
```



```

if (cancel) {
    // There was an error; don't attempt login and focus the first
    // form field with an error.
    focusView.requestFocus();
} else {
    // Show a progress spinner, and kick off a background task to
    // perform the user login attempt.
    showProgress(true);
    mAuthTask = new UserLoginTask(email, password, url);
    mAuthTask.execute((Void) null);
}

private boolean isValidEmail(String email) {
    //TODO: Replace this with your own logic
    return email.contains("@");
}

private boolean isValidPassword(String password) {
    //TODO: Replace this with your own logic
    return password.length() > 4;
}

/**
 * Shows the progress UI and hides the login form.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB_MR2)
private void showProgress(final boolean show) {...}

@Override
public Loader<Cursor> onCreateLoader(int i, Bundle bundle)
{...}
    
```

```

@Override
public void onFinished(Loader<Cursor> cursorLoader, Cursor cursor) {...}

@Override
public void onLoaderReset(Loader<Cursor> cursorLoader) {
}

private void addEmailsToAutoComplete(List<String> emailAddressCollection) {...}

@Override
public void onStart() {...}

@Override
public void onStop() {...}

private interface ProfileQuery {...}

/**
 * Use an AsyncTask to fetch the user's email addresses on a background thread, and update
 * the email text field with results on the main UI thread.
 */
class SetupEmailAutoCompleteTask extends AsyncTask<Void, Void, List<String>> {...}

/**
 * Represents an asynchronous login/registration task used to authenticate
 * the user.
 */
public class UserLoginTask extends AsyncTask<Void, Void, Boolean>
    
```



```

public class UserLoginTask extends AsyncTask<Void, Void, Boolean>
{
    private final String mEmail;
    private final String mPassword;
    private final String urlapiREST;

    UserLoginTask(String email, String password, String url) {
        Log.i("ADRISERP", "Llego a la tarea e logueo");
        mEmail = email;
        mPassword = password;
        urlapiREST = url;
        Log.i("ADRISERP", email);
        Log.i("ADRISERP", password);
    }

    @Override
    protected Boolean doInBackground(Void... params)
    {
        // TODO: attempt authentication against a network service.
        Log.i("ADRISERP", "Llego a doInBackground");

        //final TextView outputView = (TextView) findViewById(R.id.showOutput);
        final StringBuilder output = new StringBuilder("");

        // Validamos conexion
        ConnectivityManager connMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
        if (networkInfo != null && networkInfo.isConnected())
        {
            Log.i("ADRISERP", "Paso conexion");
        }
    }
}
    
```

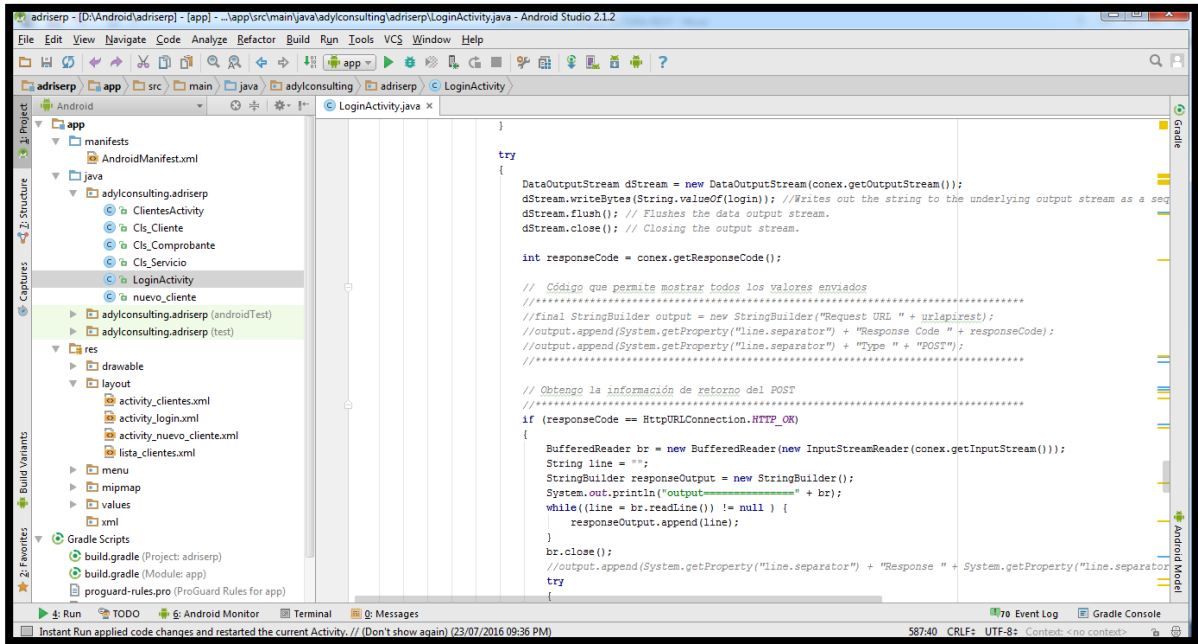
```

// Validamos conexion
ConnectivityManager connMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
if (networkInfo != null && networkInfo.isConnected())
{
    Log.i("ADRISERP", "Paso conexion");
    try
    {
        try
        {
            URL url = new URL(urlapiREST);
            HttpURLConnection conex = (HttpURLConnection)url.openConnection();

            conex.setDoOutput(true);
            conex.setDoInput(true);
            conex.setRequestProperty("Content-Type", "application/json");
            conex.setRequestProperty("Accept", "application/json");
            conex.setRequestMethod("POST");

            // Agregamos los valores a enviar como parametros
            JSONObject login = new JSONObject();
            try
            {
                login.put("usuario", mEmail);
                login.put("contrasena", mPassword);
            }
            catch (JSONException e)
            {
                e.printStackTrace();
            }
        }
    }
}
    
```





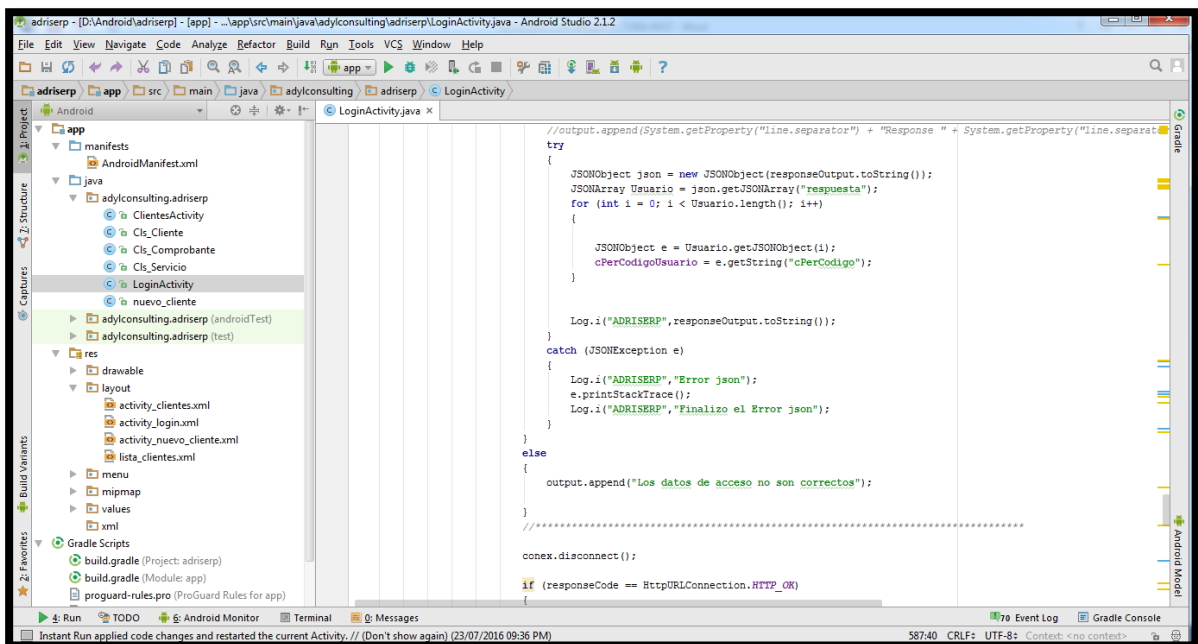
```

try
{
    DataOutputStream dStream = new DataOutputStream(conex.getOutputStream());
    dStream.writeBytes(String.valueOf(login)); //Writes out the string to the underlying output stream as a seq
    dStream.flush(); // Flushes the data output stream.
    dStream.close(); // Closing the output stream.

    int responseCode = conex.getResponseCode();

    // Código que permite mostrar todos los valores enviados
    //*****
    //final StringBuilder output = new StringBuilder("Request URL " + urlapiRest);
    //output.append(System.getProperty("line.separator") + "Response Code " + responseCode);
    //output.append(System.getProperty("line.separator") + "Type " + "POST");
    //*****

    // Obtengo la información de retorno del POST
    //*****
    if (responseCode == HttpURLConnection.HTTP_OK)
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(conex.getInputStream()));
        String line = "";
        StringBuilder responseOutput = new StringBuilder();
        System.out.println("output-----" + br);
        while (line = br.readLine() != null) {
            responseOutput.append(line);
        }
        br.close();
        //output.append(System.getProperty("line.separator") + "Response " + System.getProperty("line.separator")
        try
    }
}
    
```



```

//output.append(System.getProperty("line.separator") + "Response " + System.getProperty("line.separator")
try
{
    JSONObject json = new JSONObject(responseOutput.toString());
    JSONArray Usuario = json.getJSONArray("respuesta");
    for (int i = 0; i < Usuario.length(); i++)
    {
        JSONObject e = Usuario.getJSONObject(i);
        cPerCodigoUsuario = e.getString("cPerCodigo");
    }

    Log.i("ADRI SERP", responseOutput.toString());
}
catch (JSONException e)
{
    Log.i("ADRI SERP", "Error json");
    e.printStackTrace();
    Log.i("ADRI SERP", "Finalizo el Error json");
}
else
{
    output.append("Los datos de acceso no son correctos");
}
//*****
conex.disconnect();

if (responseCode == HttpURLConnection.HTTP_OK)
{
    
```



```

    if (responseCode == HttpURLConnection.HTTP_OK)
    {
        return true;
    }
    else
    {
        return false;
    }
}
finally
{
    conex.disconnect();
}
catch (MalformedURLException e)
{
    System.out.println("The URL is not valid.");
    System.out.println(e.getMessage());
}
}
catch (IOException e)
{
    System.out.println("IOException");
    System.out.println(e.getMessage());
}
}
else
{
    Log.i("ADRISERP", "No paso la conexion");
}
}

```

```

else
{
    Log.i("ADRISERP", "No paso la conexion");
    return false;
}

// TODO: register the new account here.
return true;
}

@Override
protected void onPostExecute(final Boolean success) {
    mAuthTask = null;
    showProgress(false);

    if (success)
    {
        mPasswordView.setText("");
        mEmailView.setText("");
        //startActivity(new Intent(LoginActivity.this, ClientesActivity.class));

        Intent intent = new Intent(LoginActivity.this, ClientesActivity.class);
        intent.putExtra("cPerCodigoUsuario", cPerCodigoUsuario);
        startActivity(intent);

        SharedPreferences preferencesUser = getSharedPreferences("PreferencesUser", Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = preferencesUser.edit();
        editor.putString("cPerCodigoUsuario", cPerCodigoUsuario);
        editor.commit();
    }
}

```



```

//startActivity(new Intent(LoginActivity.this, ClientesActivity.class));
Intent intent = new Intent(LoginActivity.this, ClientesActivity.class);
intent.putExtra("cPerCodigoUsuario", cPerCodigoUsuario);
startActivity(intent);

SharedPreferences preferencesUser = getSharedPreferences("PreferencesUser", Context.MODE_PRIVATE);
SharedPreferences.Editor editor = preferencesUser.edit();
editor.putString("cPerCodigoUsuario", cPerCodigoUsuario);
editor.commit();

} else
{
    mOuputView.setText("El usuario o clave ingresados son incorrectos");
}

@Override
protected void onCancelled() {
    mAuthTask = null;
    showProgress(false);
}
}
    
```

- Listar Clientes

```

class LoadClientesAsyn extends AsyncTask<Void, Void, Boolean>
{
    ProgressDialog pDialog = new ProgressDialog(ClientesActivity.this);
    final TextView outputView = (TextView) findViewById(R.id.showOutput);
    private final String urlapiREST;
    private final String jsonData = "";
    private JSONObject jsonObj = null;

    LoadClientesAsyn(String url) { urlapiREST = url; }

    protected void onPreExecute()
    {
        Log.i("ADRISERP", "Llegue al LoadClientesAsyn onPreExecute");
        pDialog.setMessage("Consultando datos");
        pDialog.setIndeterminate(true);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    @Override
    protected Boolean doInBackground(Void... params)
    {
        ConnectivityManager connMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
        if (networkInfo != null && networkInfo.isConnected())
        {
            try
            {
                try
            }
        }
    }
}
    
```



```

try
{
    URL url = new URL(urlapiREST);
    HttpURLConnection conex = (HttpURLConnection)url.openConnection();

    //conex.setDoOutput(true);
    //conex.setDoInput(true);
    conex.setRequestProperty("Content-Type", "application/json");
    conex.setRequestProperty("Accept", "application/json");
    conex.setRequestMethod("GET");

    try
    {
        //DataOutputStream dStream = new DataOutputStream(conex.getOutputStream());
        //dStream.writeBytes(String.valueOf(login)); //Writes out the string to the underlying output stream as
        //dStream.flush(); // Flushes the data output stream.
        //dStream.close(); // Closing the output stream.

        int responseCode = conex.getResponseCode();

        // Código que permite mostrar todos los valores enviados
        //final StringBuilder output = new StringBuilder("Request URL " + urlapiREST);
        //output.append(System.getProperty("line.separator") + "Response Code " + responseCode);
        //output.append(System.getProperty("line.separator") + "Type " + "POST");

        // Obtengo la información de retorno del POST
        if (responseCode == HttpURLConnection.HTTP_OK)
    
```

```

// Obtengo la información de retorno del POST
//*****
if (responseCode == HttpURLConnection.HTTP_OK)
{
    BufferedReader br = new BufferedReader(new InputStreamReader(conex.getInputStream()));
    String line = "";
    StringBuilder responseOutput = new StringBuilder();
    System.out.println("output-----" + br);
    while ((line = br.readLine()) != null)
    {
        responseOutput.append(line);
    }
    br.close();
    output.append(responseOutput.toString());

    Log.i("ADRI SERP", "Antes de");
}
else
{
    Log.i("ADRI SERP", "Los datos de acceso no son correctos");
}
//*****

conex.disconnect();

ClientesActivity.this.runOnUiThread() -> {
    try
    {
        JSONObject json = new JSONObject(output.toString());
        JSONArray jsonArray = json.getJSONArray("respuesta");
        ArrayList<Cls_Cliente> clienteLista = new ArrayList<>();
    
```




```

try
{
    JSONObject json = new JSONObject(output.toString());
    JSONArray jsonArray = json.getJSONArray("respuesta");
    ArrayList<Cls_Cliente> Cliente_Lista = new ArrayList<>();

    //ArrayList<cls_sms> smsList = new ArrayList<cls_sms>();
    for (int i = 0; i < jsonArray.length(); i++)
    {
        JSONObject e = jsonArray.getJSONObject(i);
        int u = (i+1); //e.getString("cPerCodigo");
        String n = e.getString("cPerNombre");
        String s = e.getString("Restante");
        String usuario = e.getString("Usuario");
        String cPerCodigo = e.getString("cPerCodigo");
        String Cuenta = e.getString("Cuenta");

        //Log.i("HEL Cliente es:",s + n + m);
        Cliente_Lista.add(new Cls_Cliente(cPerCodigo, String.valueOf(u), usuario,n,s,Cuenta));
    }
    Log.i("ADRISERP", String.valueOf(jsonArray.length()) + " Clientes");

    ListView lvClientes = (ListView) findViewById(R.id.listViewClientes);

    SimpleClientesAdapter adapter = new SimpleClientesAdapter(CientesActivity.this, Cliente_Lista);
    lvClientes.setAdapter(adapter);
    lvClientes.setOnItemClickListener((adapter, view, position, id) -> {
        Cls_Cliente cliente = (Cls_Cliente) (adapter.getItemAtPosition(position));
        //Toast.makeText(this, "De:" + cliente.getUsuario() + "\n" + cliente.getNombre(),
        Intent intent = new Intent(CientesActivity.this, nuevo_cliente.class);
        intent.putExtra("cPerCodigo", cliente.getCodigo());
    });
}
    
```

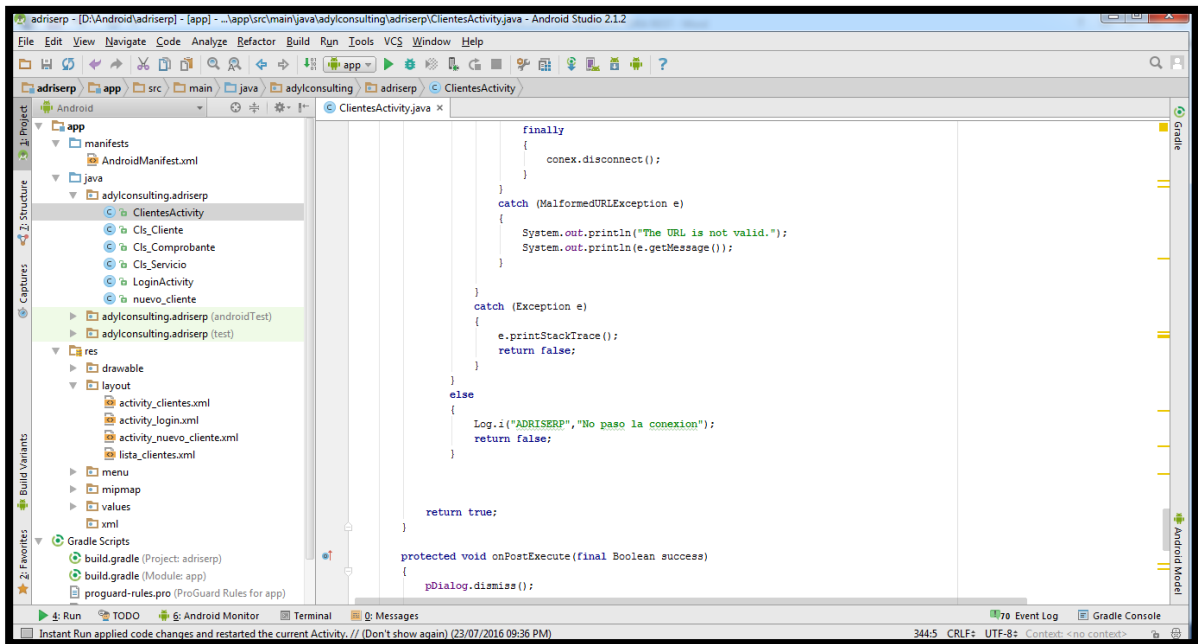
```

        ListView lvClientes = (ListView) findViewById(R.id.listViewClientes);

        SimpleClientesAdapter adapter = new SimpleClientesAdapter(CientesActivity.this, Cliente_Lista);
        lvClientes.setAdapter(adapter);
        lvClientes.setOnItemClickListener((adapter, view, position, id) -> {
            Cls_Cliente cliente = (Cls_Cliente) (adapter.getItemAtPosition(position));
            //Toast.makeText(this, "De:" + cliente.getUsuario() + "\n" + cliente.getNombre(),
            Intent intent = new Intent(CientesActivity.this, nuevo_cliente.class);
            intent.putExtra("cPerCodigo", cliente.getCodigo());
            intent.putExtra("cPerNombre", cliente.getNombre());
            intent.putExtra("cPerCodigoUsuario", cliente.getCodigoUsuario());
            intent.putExtra("nPerCtaCodigo", cliente.getCuenta());
            startActivity(intent);
            Log.i("ADRISERP", "recibi click");
        });

        //outputView.setText(output);
        //mPasswordView.setText("");
        //mEmailView.setText("");
    }
    catch (JSONException e)
    {
        Log.i("ADRISERP", "Error json");
        e.printStackTrace();
        Log.i("ADRISERP", "Finalizo el Error json");
    }
}
finally
{
    conex.disconnect();
}
    
```



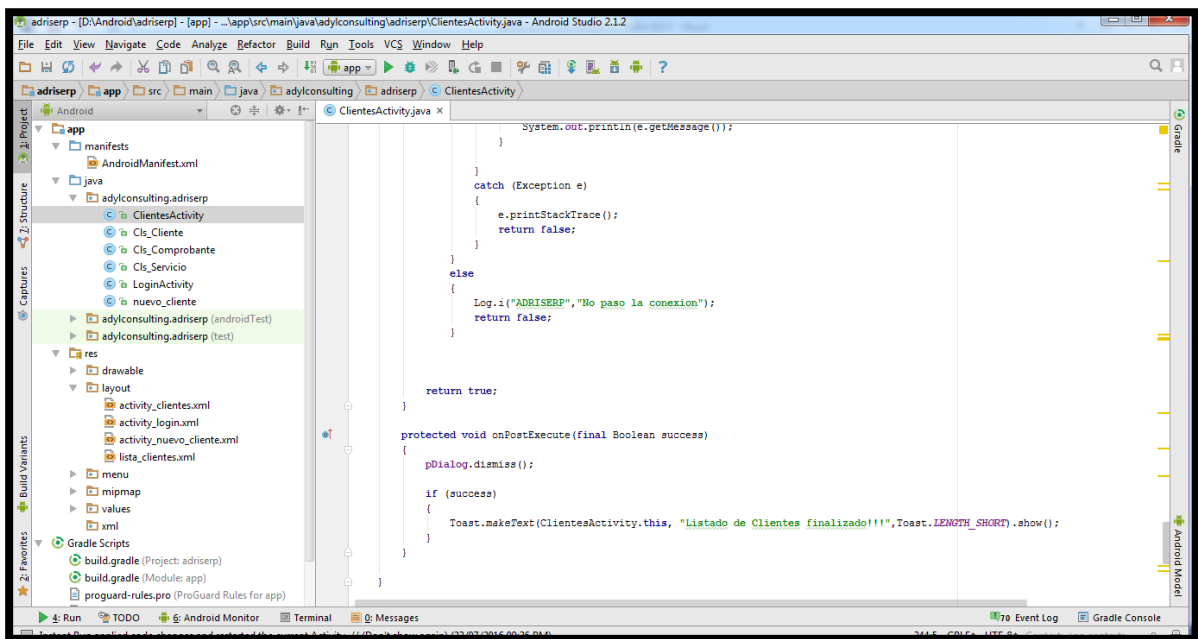


```

finally
    {
        conex.disconnect();
    }
}
catch (MalformedURLException e)
{
    System.out.println("The URL is not valid.");
    System.out.println(e.getMessage());
}
}
catch (Exception e)
{
    e.printStackTrace();
    return false;
}
}
else
{
    Log.i("ADRISERP", "No paso la conexion");
    return false;
}

return true;
}

protected void onPostExecute(final Boolean success)
{
    progressDialog.dismiss();
}
    
```



```

        System.out.println(e.getMessage());
    }
}
catch (Exception e)
{
    e.printStackTrace();
    return false;
}
}
else
{
    Log.i("ADRISERP", "No paso la conexion");
    return false;
}

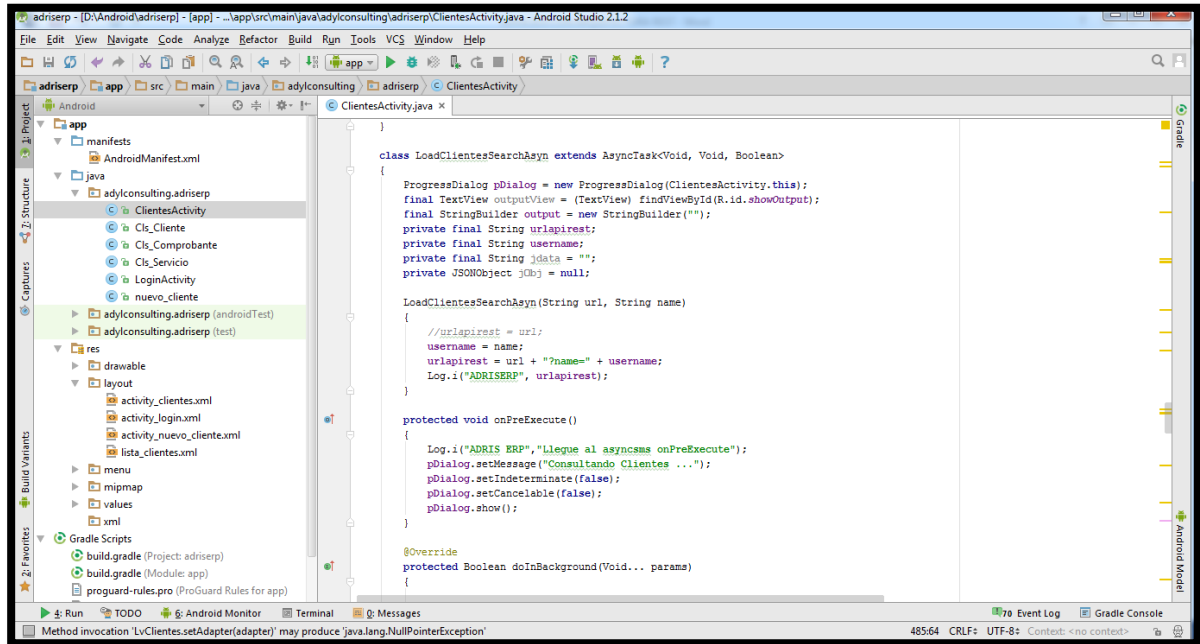
return true;
}

protected void onPostExecute(final Boolean success)
{
    progressDialog.dismiss();

    if (success)
    {
        Toast.makeText(CientesActivity.this, "Listado de Clientes finalizado!!!", Toast.LENGTH_SHORT).show();
    }
}
    
```



- **Buscar Clientes**



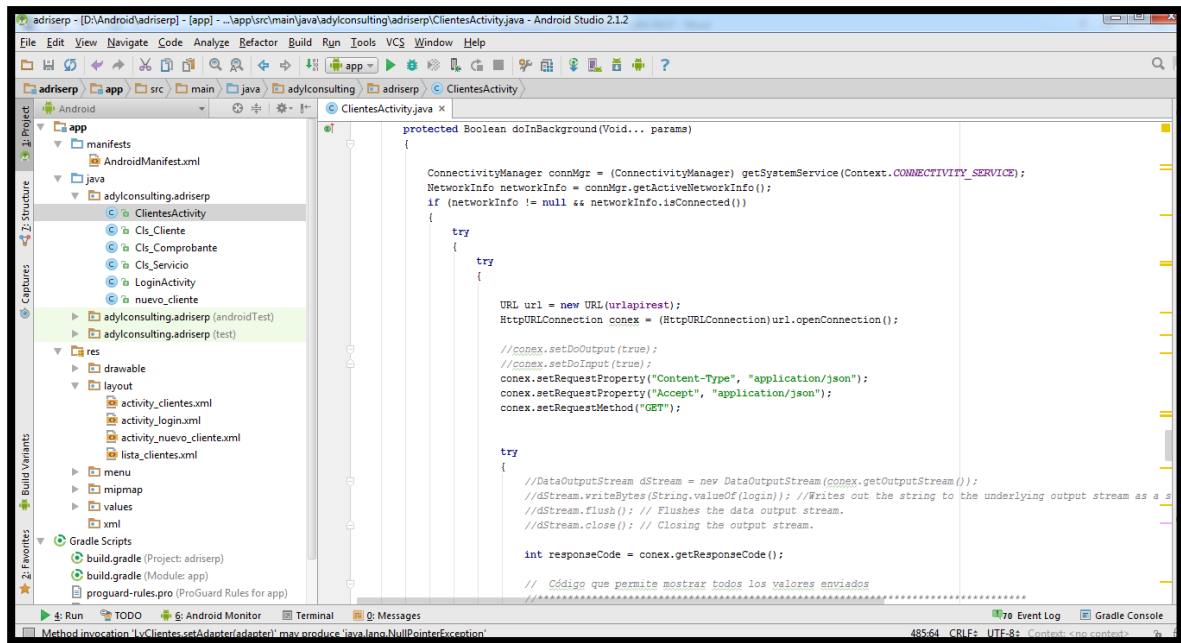
```

class LoadClientesSearchAsync extends AsyncTask<Void, Void, Boolean>
{
    ProgressDialog pDialog = new ProgressDialog(CientesActivity.this);
    final TextView outputView = (TextView) findViewById(R.id.showOutput);
    final StringBuilder output = new StringBuilder("");
    private final String urlapiRest;
    private final String username;
    private final String jsonData = "";
    private JSONObject jsonObj = null;

    LoadClientesSearchAsync(String url, String name)
    {
        //urlapiRest = url;
        username = name;
        urlapiRest = url + "?name=" + username;
        Log.i("ADRIERP", urlapiRest);
    }

    protected void onPreExecute()
    {
        Log.i("ADRI ERP", "Lleegue al asynccms onPreExecute");
        pDialog.setMessage("Consultando Clientes ...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    @Override
    protected Boolean doInBackground(Void... params)
    {
    }
}
    
```



```

protected Boolean doInBackground(Void... params)
{
    ConnectivityManager connMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected())
    {
        try
        {
            try
            {
                URL url = new URL(urlapiRest);
                HttpURLConnection conex = (HttpURLConnection)url.openConnection();

                //conex.setDoOutput(true);
                //conex.setDoInput(true);
                conex.setRequestProperty("Content-Type", "application/json");
                conex.setRequestProperty("Accept", "application/json");
                conex.setRequestMethod("GET");

            }

            try
            {
                //DataOutputStream dStream = new DataOutputStream(conex.getOutputStream());
                //dStream.writeBytes(String.valueOf(login)); //Writes out the string to the underlying output stream as a s
                //dStream.flush(); // Flushes the data output stream.
                //dStream.close(); // Closing the output stream.

                int responseCode = conex.getResponseCode();

                // Código que permite mostrar todos los valores enviados
                //.....
            }
        }
    }
}
    
```



```

adriserp - [D:\Android\adriserp] - [app] - ...lapp\src\main\java\adylconsulting\adriserp\CientesActivity.java - Android Studio 21.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
adriserp [app] [src] [main] [java] [adylconsulting] [adriserp] [CientesActivity]
//OutputStream.close(); // Closing the output stream.
int responseCode = conex.getResponseCode();
// Código que permite mostrar todos los valores enviados
//final StringBuilder output = new StringBuilder("Request URL " + urlapiRest);
//output.append(System.getProperty("line.separator") + "Response Code " + responseCode);
//output.append(System.getProperty("line.separator") + "Type " + "POST");
//.....
// Obtengo la información de retorno del POST
//.....
if (responseCode == HttpURLConnection.HTTP_OK)
{
    BufferedReader br = new BufferedReader(new InputStreamReader(conex.getInputStream()));
    String line = "";
    StringBuilder responseOutput = new StringBuilder();
    System.out.println("output =====> " + br);
    while ((line = br.readLine()) != null)
    {
        responseOutput.append(line);
    }
    br.close();
    output.append(responseOutput.toString());
    Log.i("ADRI SERP", "Antes de");
    //JSONObject myObj = new JSONObject(responseOutput.toString());
    //JSONObject responseJsonObj = new JSONObject( URLDecoder.decode( responseOutput.toString(), "UTF-8"
    //JSONParser parser_obj = new JSONParser();
    //JSONArray array_obj = (JSONArray) parser_obj.parse("String from web service");
    // Log.i("ADRI SERP", "Despues de");
}
Method invocation 'LvCientes.setAdapter(adapter)' may produce 'java.lang.NullPointerException'
485:64 CRLF: UTF-8 Context: <no context>

```

```

adriserp - [D:\Android\adriserp] - [app] - ...lapp\src\main\java\adylconsulting\adriserp\CientesActivity.java - Android Studio 21.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
adriserp [app] [src] [main] [java] [adylconsulting] [adriserp] [CientesActivity]
}
else
{
    Log.i("ADRI SERP", "Los datos de acceso no son correctos");
}
//.....
conex.disconnect();
CientesActivity.this.runOnUiThread() -> {
    try
    {
        /*JSONArray jArray = new JSONArray (output.toString()); //convert string to JSONArray
        for (int i = 0; i < jArray .length(); i++)
        {
            Log.i("ADRI SERP", "Los datos de acceso no son correctos");
        }
        */
        JSONObject json = new JSONObject (output.toString());
        JSONArray jArray = json.getJSONArray("respuesta");
        ArrayList<Cls_Cliente> Cliente_Lista = new ArrayList<>();
        //ArrayList<cls_sms> smsList = new ArrayList<cls_sms>();
        for (int i = 0; i < jArray.length(); i++)
        {
            JSONObject e = jArray.getJSONObject(i);
            int u = (i+1); //e.getString("%PerCodigo");
            String n = e.getString("cPerNombre");
            String a = e.getString("Restante");
            String usuario = e.getString("Gauario");

```



```

//ArrayList<cls_sms> smsList = new ArrayList<cls_sms>();
for (int i = 0; i < jsonArray.length(); i++)
{
    JSONObject e = jsonArray.getJSONObject(i);
    int u = (i+1); //e.getString("cPerCodigo");
    String n = e.getString("cPerNombre");
    String s = e.getString("Restante");
    String usuario = e.getString("Usuario");
    String cPerCodigo = e.getString("cPerCodigo");
    String Cuenta = e.getString("Cuenta");

    //Log.i("HEL Cliente ms:",s + n + m);
    Cliente_Lista.add(new Cls_Cliente(cPerCodigo, String.valueOf(u), usuario,n,s, Cuenta));
}
Log.i("ADRI SERP", String.valueOf(jsonArray.length()) + " Clientes");

ListView lvClientes = (ListView) findViewById(R.id.listViewClientes);

SimpleClientesAdapter adapter = new SimpleClientesAdapter(CientesActivity.this, Cliente_Lista);
lvClientes.setAdapter(adapter);
lvClientes.setOnItemClickListener((adapter, view, position, id) -> {
    Cls_Cliente cliente = (Cls_Cliente) (adapter.getItemAtPosition(position));
    //Toast.makeText(this, "De:" + cliente.getUsuario() + " a:" + cliente.getNombre(), Toast.LENGTH_SHORT).show();
    Intent intent = new Intent(CientesActivity.this, nuevo_cliente.class);
    intent.putExtra("cPerCodigo", cliente.getCodigo());
    intent.putExtra("cPerNombre", cliente.getNombre());
    intent.putExtra("cPerCodigoUsuario", cPerCodigoUsuario);
    intent.putExtra("nPerCtaCodigo", cliente.getCuenta());
    startActivity(intent);
    Log.i("ADRI SERP", "recibi click");
});
    
```

```

        Log.i("ADRI SERP", "recibi click");
    });

    //outputView.setText(output);
    //mPasswordView.setText("");
    //mEmailView.setText("");

    catch (JSONException e)
    {
        Log.i("ADRI SERP", "Error json");
        e.printStackTrace();
        Log.i("ADRI SERP", "Finalizo el Error json");
    }

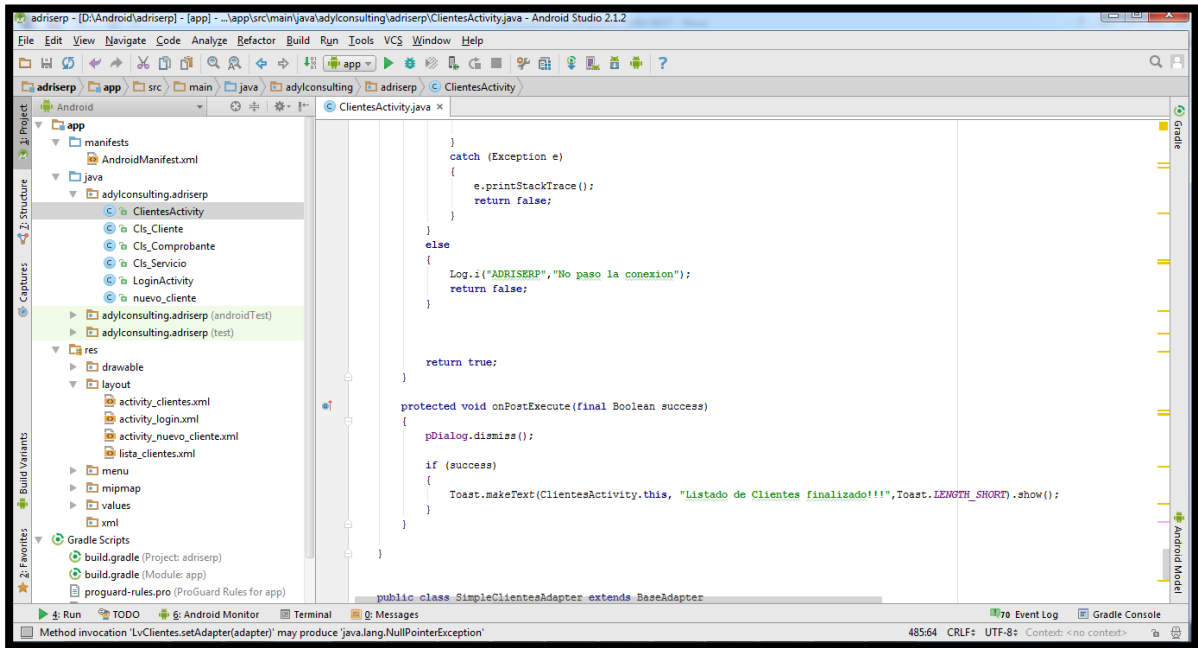
    });

    finally
    {
        conex.disconnect();
    }

    catch (MalformedURLException e)
    {
        System.out.println("The URL is not valid.");
        System.out.println(e.getMessage());
    }

    catch (Exception e)
    {
        e.printStackTrace();
        return false;
    }
    
```





```

    }
    catch (Exception e)
    {
        e.printStackTrace();
        return false;
    }
    else
    {
        Log.i("ADRI SERP", "No paso la conexon");
        return false;
    }
}

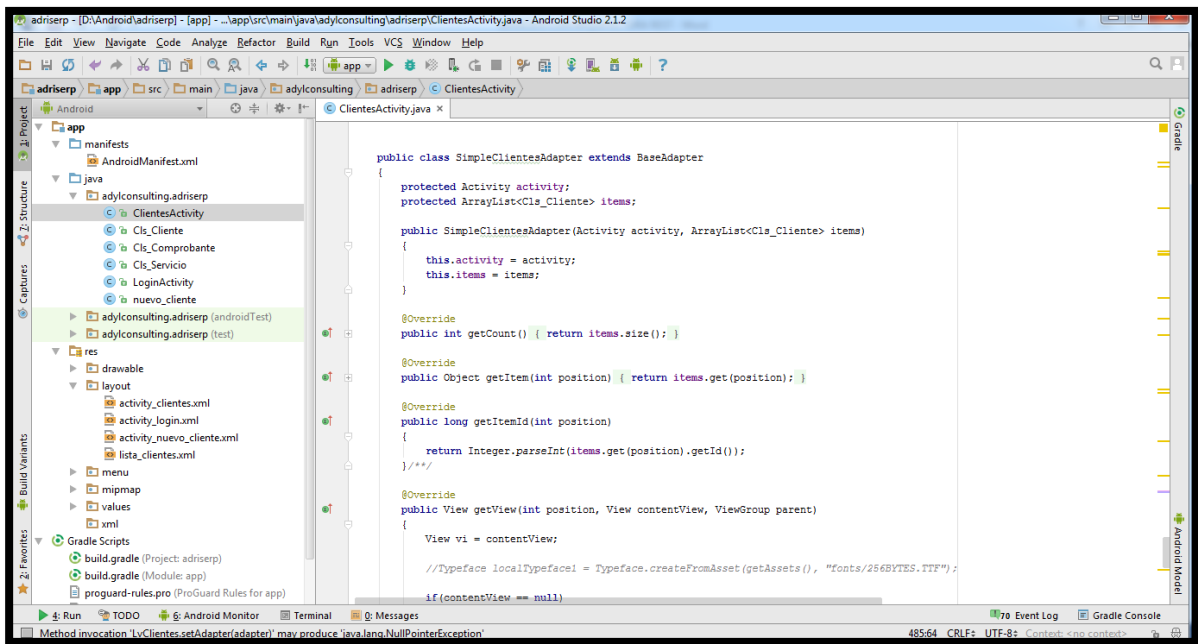
return true;

protected void onPostExecute(final Boolean success)
{
    ProgressDialog.dismiss();

    if (success)
    {
        Toast.makeText(CientesActivity.this, "Listado de Clientes finalizado!!!", Toast.LENGTH_SHORT).show();
    }
}
}

public class SimpleClientesAdapter extends BaseAdapter

```



```

public class SimpleClientesAdapter extends BaseAdapter
{
    protected Activity activity;
    protected ArrayList<Cls_Cliente> items;

    public SimpleClientesAdapter(Activity activity, ArrayList<Cls_Cliente> items)
    {
        this.activity = activity;
        this.items = items;
    }

    @Override
    public int getCount() { return items.size(); }

    @Override
    public Object getItem(int position) { return items.get(position); }

    @Override
    public long getItemId(int position)
    {
        return Integer.parseInt(items.get(position).getId());
    }

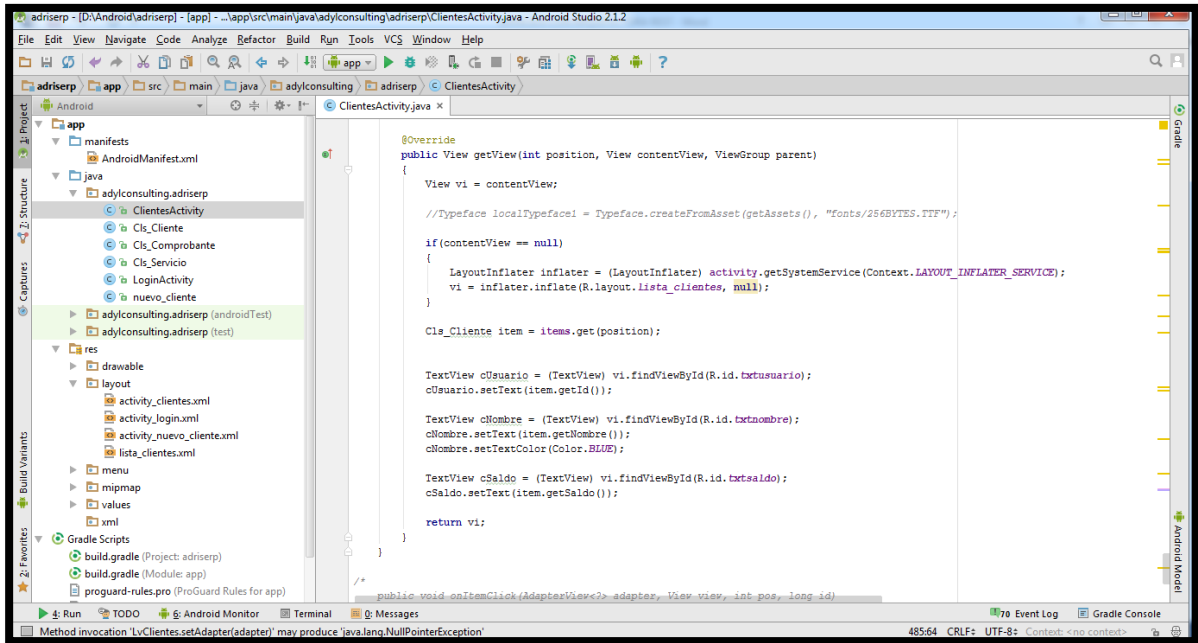
    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {
        View vi = convertView;

        //Typeface localTypeface1 = Typeface.createFromAsset(getAssets(), "fonts/256BYTES.TTF");

        if (convertView == null)

```





```

@Override
public View getView(int position, View convertView, ViewGroup parent)
{
    View vi = convertView;

    //Typeface localTypeface1 = Typeface.createFromAsset(getAssets(), "fonts/256BYTES.TTF");

    if (convertView == null)
    {
        LayoutInflater inflater = (LayoutInflater) activity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        vi = inflater.inflate(R.layout.lista_clientes, null);
    }

    Cla_Cliente item = items.get(position);

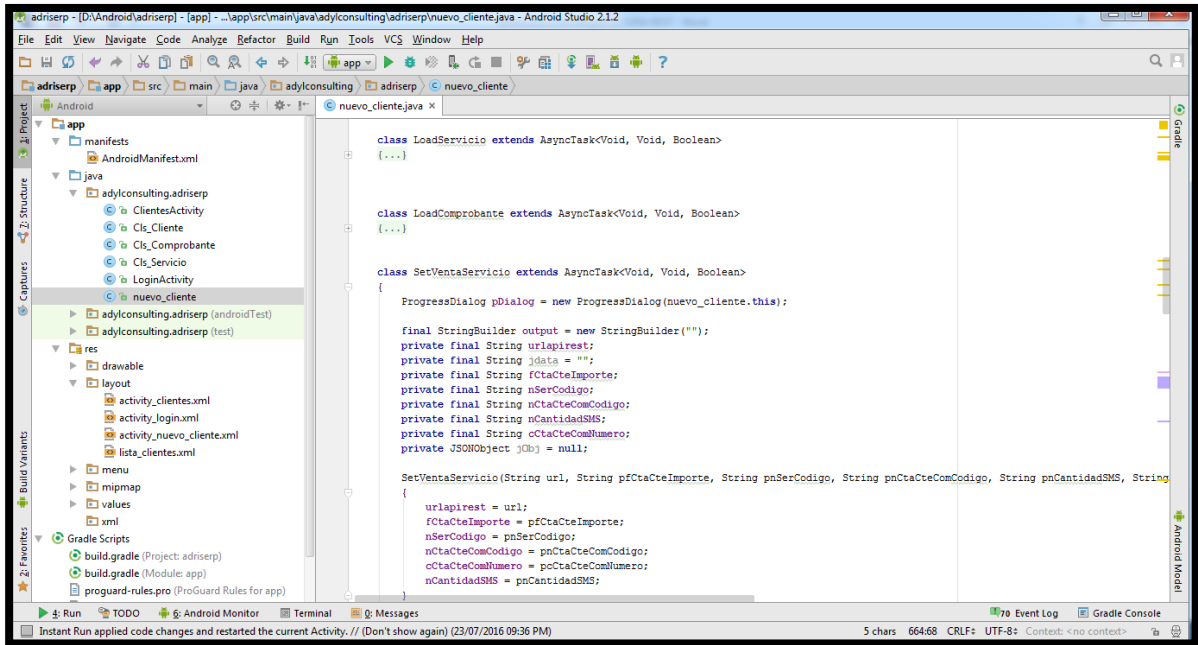
    TextView cUsuario = (TextView) vi.findViewById(R.id.txtusuario);
    cUsuario.setText(item.getId());

    TextView cNombre = (TextView) vi.findViewById(R.id.txtnombre);
    cNombre.setText(item.getNombre());
    cNombre.setTextColor(Color.BLUE);

    TextView cSaldo = (TextView) vi.findViewById(R.id.txtsaldo);
    cSaldo.setText(item.getSaldo());

    return vi;
}
    
```

- Registrar venta de servicio



```

class LoadServicio extends AsyncTask<Void, Void, Boolean>
{
    [...]
}

class LoadComprobante extends AsyncTask<Void, Void, Boolean>
{
    [...]
}

class SetVentaServicio extends AsyncTask<Void, Void, Boolean>
{
    ProgressDialog pDialog = new ProgressDialog(nuevo_cliente.this);

    final StringBuilder output = new StringBuilder("");
    private final String urlapires;
    private final String jdata = "";
    private final String fCtaCteImporte;
    private final String nSerCodigo;
    private final String nCtaCteComCodigo;
    private final String mCantidadSMS;
    private final String cCtaCteComNumero;
    private JSONObject jObj = null;

    SetVentaServicio(String url, String pfCtaCteImporte, String pnSerCodigo, String pnCtaCteComCodigo, String pnCantidadSMS, String
    {
        urlapires = url;
        fCtaCteImporte = pfCtaCteImporte;
        nSerCodigo = pnSerCodigo;
        nCtaCteComCodigo = pnCtaCteComCodigo;
        cCtaCteComNumero = pnCtaCteComNumero;
        mCantidadSMS = pnCantidadSMS;
    }
}
    
```



```

SetVentaServicio(String url, String pfCtaCteImporte, String pnSerCodigo, String pnCtaCteComCodigo, String pnCantidadSMS, Stri
{
    urlaprest = url;
    fCtaCteImporte = pfCtaCteImporte;
    nSerCodigo = pnSerCodigo;
    nCtaCteComCodigo = pnCtaCteComCodigo;
    cCtaCteComNumero = pcCtaCteComNumero;
    nCantidadSMS = pnCantidadSMS;
}

protected void onPreExecute()
{
    Log.i("ADRISS ERP", "Llegue al asyncsms onPreExecute");
    progressDialog.setMessage("Registrando Servicio ...");
    progressDialog.setIndeterminate(false);
    progressDialog.setCancelable(false);
    progressDialog.show();
}

@Override
protected Boolean doInBackground(Void... params)
{
    ConnectivityManager connMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected())
    {
        try
        {
            try
            {
                URL url = new URL(urlaprest);
            }
        }
    }
}
    
```

```

try
{
    try
    {
        URL url = new URL(urlaprest);
        HttpURLConnection conex = (HttpURLConnection)url.openConnection();

        conex.setDoOutput(true);
        conex.setDoInput(true);
        conex.setRequestProperty("Content-Type", "application/json");
        conex.setRequestProperty("Accept", "application/json");
        conex.setRequestMethod("POST");

        // Agregamos los valores a enviar como parametros
        JSONObject Venta = new JSONObject();
        try
        {
            Venta.put("nPerCtaCodigo", nPerCtaCodigo);
            Venta.put("fCtaCteImporte", fCtaCteImporte);
            Venta.put("nSerCodigo", nSerCodigo);
            Venta.put("cPerCodigo", cPerCodigoUsuario);
            Venta.put("nCtaCteComCodigo", nCtaCteComCodigo);
            Venta.put("cCtaCteComNumero", cCtaCteComNumero);
            Venta.put("nCantidadSMS", nCantidadSMS);

            Log.i("ADRISS ERP - NEWSERVICE 1", nPerCtaCodigo);
            Log.i("ADRISS ERP - NEWSERVICE 2", fCtaCteImporte);
            Log.i("ADRISS ERP - NEWSERVICE 3", nSerCodigo);
            Log.i("ADRISS ERP - NEWSERVICE 4", cPerCodigoUsuario);
            Log.i("ADRISS ERP - NEWSERVICE 5", nCtaCteComCodigo);
            Log.i("ADRISS ERP - NEWSERVICE 6", cCtaCteComNumero);
            Log.i("ADRISS ERP - NEWSERVICE 7", nCantidadSMS);
        }
    }
}
    
```




```

Ventas.put("nCantidadSMS", nCantidadSMS);

Log.i("ADRI SERP - NEWSERVICE 1", nPerCtaCodigo);
Log.i("ADRI SERP - NEWSERVICE 2", fCtaCteImporte);
Log.i("ADRI SERP - NEWSERVICE 3", nSerCodigo);
Log.i("ADRI SERP - NEWSERVICE 4", cPerCodigoUsuario);
Log.i("ADRI SERP - NEWSERVICE 5", nCtaCteComCodigo);
Log.i("ADRI SERP - NEWSERVICE 6", cCtaCteComNumero);
Log.i("ADRI SERP - NEWSERVICE 7", nCantidadSMS);
    }
    catch (JSONException e)
    {
        e.printStackTrace();
    }
}

try
{
    DataOutputStream dStream = new DataOutputStream(conex.getOutputStream());
    dStream.writeBytes(String.valueOf(Ventas)); //Writes the string to the underlying output stream as a sequence of bytes.
    dStream.flush(); // Flushes the data output stream.
    dStream.close(); // Closing the output stream.

    int responseCode = conex.getResponseCode();

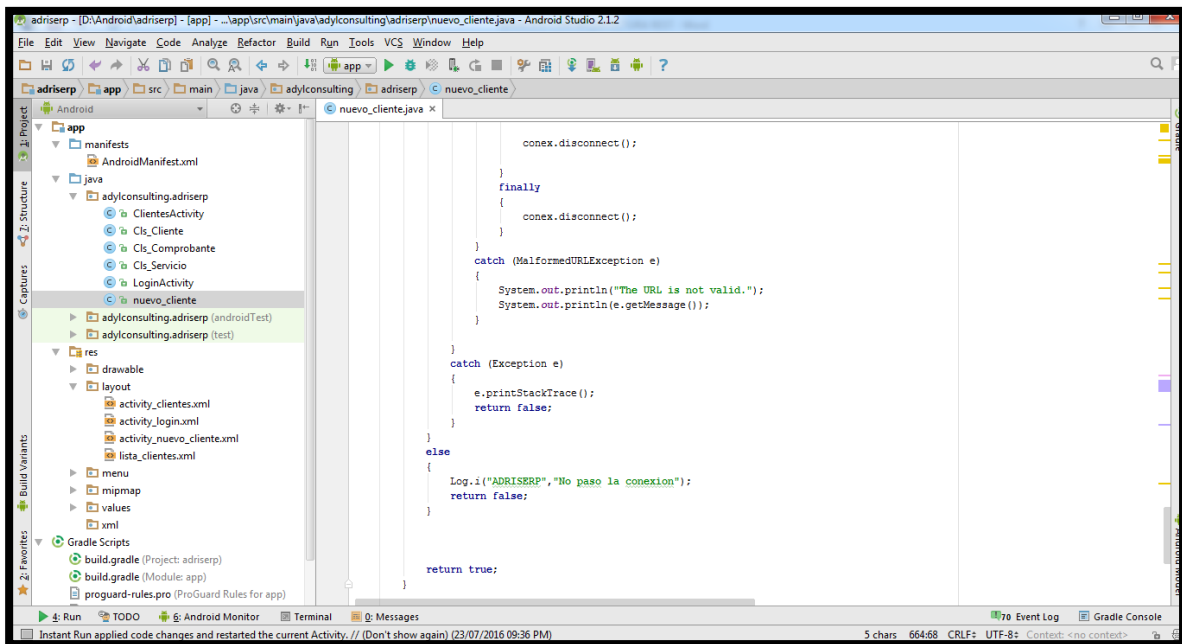
    // Código que permite mostrar todos los valores enviados
    //*****
    //final StringBuilder output = new StringBuilder("Request URL " + urlapiprest);
    //output.append(System.getProperty("line.separator") + "Response Code " + responseCode);
    //output.append(System.getProperty("line.separator") + "Type " + "POST");
    //*****

    // Obtengo la información de retorno del POST
}
    
```

```

//*****
if (responseCode == HttpURLConnection.HTTP_OK)
{
    /*
    BufferedReader br = new BufferedReader(new InputStreamReader(conex.getInputStream()));
    String line = "";
    StringBuilder responseOutput = new StringBuilder();
    System.out.println("Output=" + br);
    while((line = br.readLine()) != null)
    {
        responseOutput.append(line);
    }
    br.close();
    output.append(responseOutput.toString());
    */
    return true;
}
else
{
    BufferedReader br = new BufferedReader(new InputStreamReader(conex.getInputStream()));
    String line = "";
    StringBuilder responseOutput = new StringBuilder();
    System.out.println("Output=" + br);
    while((line = br.readLine()) != null)
    {
        responseOutput.append(line);
    }
    br.close();
    output.append(responseOutput.toString());
    Log.i("ADRI SERP - NEWSERVICE", output.toString());
}
//*****
    
```



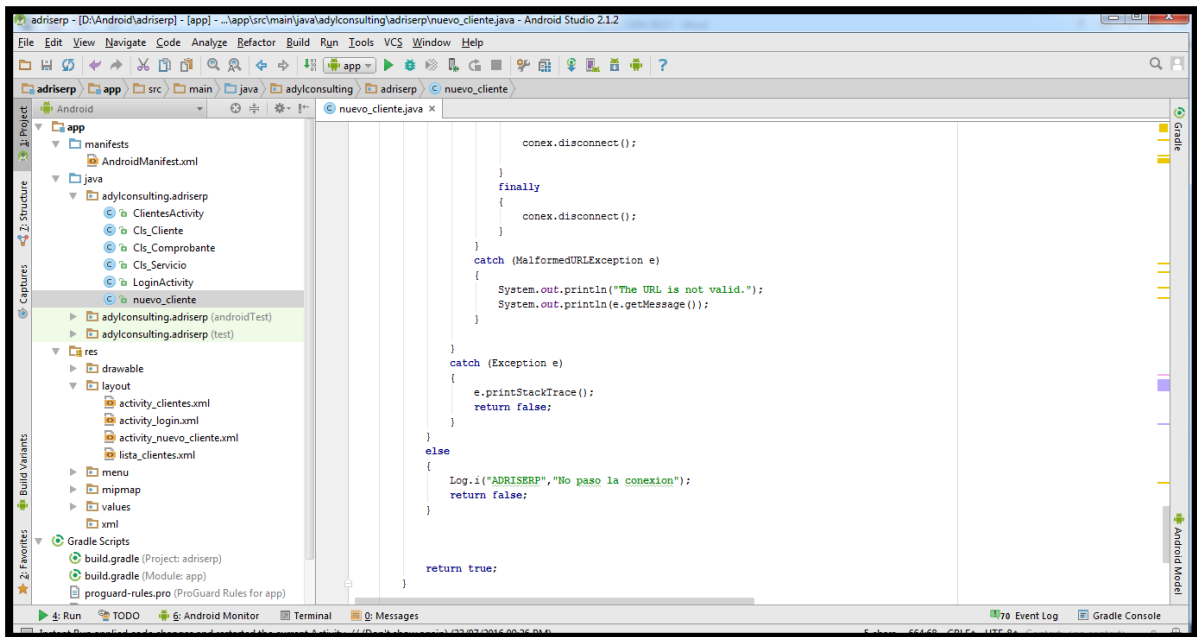


```

conex.disconnect();
    }
    finally
    {
        conex.disconnect();
    }
}
catch (MalformedURLException e)
{
    System.out.println("The URL is not valid.");
    System.out.println(e.getMessage());
}
}
catch (Exception e)
{
    e.printStackTrace();
    return false;
}
}
else
{
    Log.i("ADRISERP", "No paso la conexion");
    return false;
}
}

return true;
    }
}

```



```

conex.disconnect();
    }
    finally
    {
        conex.disconnect();
    }
}
catch (MalformedURLException e)
{
    System.out.println("The URL is not valid.");
    System.out.println(e.getMessage());
}
}
catch (Exception e)
{
    e.printStackTrace();
    return false;
}
}
else
{
    Log.i("ADRISERP", "No paso la conexion");
    return false;
}
}

return true;
    }
}

```

