

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**ESTUDIO COMPARATIVO DE LOS SERVICIOS WEB RESTFULL JERSEY Y SOAP JAX-WS PARA EL DESARROLLO DE UNA APLICACIÓN ANDROID CON WIKITUDE APLICADA A LA GESTIÓN DE INFORMACIÓN GEOLOCALIZADA DEL TURISMO DE LA PROVINCIA DE CHIMBORAZO.**

Tesis presentada para optar al grado académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

AUTORES:

**LORENA NATALY HIDALGO MACAS**

**MILTON EDISON JIMÉNEZ ACARO**

**TUTOR:** ING. JULIO SANTILLÁN

Riobamba – Ecuador

2016

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

Tribunal de Tesis certifica que: El trabajo de investigación: “ESTUDIO COMPARATIVO DE LOS SERVICIOS WEB RESTFULL JERSEY Y SOAP JAX-WS PARA EL DESARROLLO DE UNA APLICACIÓN ANDROID CON WIKITUDE APLICADA A LA GESTIÓN DE INFORMACIÓN GEOLOCALIZADA DEL TURISMO DE LA PROVINCIA DE CHIMBORAZO.“, de responsabilidad de Lorena Nataly Hidalgo Macas y Milton Edison Jiménez Acaro, ha sido minuciosamente revisado por los Miembros del Tribunal de Tesis, quedando autorizada su presentación.

Dr. Miguel Tasambay S. Ph. D

DECANO \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Julio Santillán

DIRECTOR DE ESCUELA \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Julio Santillán

DIRECTOR DE TESIS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ing. Jorge Menéndez

MIEMBRO DEL TRIBUNAL \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**NOTA DE LA TESIS………………………**

Nosotros, **Lorena Nataly Hidalgo Macas y Milton Edison Jiménez Acaro** somos responsables de las ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

**................................. .................................**

Lorena Nataly Hidalgo Macas Milton Edison Jiménez Acaro

**AGRADECIMIENTO**

*Agradezco a Dios por su infinita bondad, y por haber estado junto a nosotros en todo momento, por darnos salud, fortaleza, responsabilidad y sabiduría, por habernos permitido culminar un peldaño más de nuestras meta. A la Escuela Superior Politécnica de Chimborazo, a sus autoridades y profesores, por abrir sus puertas y darnos la confianza necesaria para triunfar en la vida y transmitir sabiduría para nuestra formación profesional. Igualmente a nuestro director de tesis Dr. Julio Santillán por su tiempo y por guiarnos de una manera adecuada en el desarrollo de este trabajo de tesis.*

*Lorena*

*Milton*

**DEDICATORIAS**

*Dedico esta Tesis a Dios por brindarme sus bendiciones cada día, agradezco a mis padres en especial a mi papi Marcos que supo infundir el amor al trabajo y por apoyarme en los buenos y malos momentos, a mis hermanos Maxi, Moni, Jhon, por su apoyo, y para mi princesa Nataly que me inspira día tras día.*

*Lorena*

*Dedico mi tesis a Dios a quién amo y admiro; a mis extraordinarios Padres, Sergio y Mercedes por su noble dedicación y amor, por ser mis amigos, mis consejeros, y por siempre guiarme y ser la voz y bendición de Dios como prioridad en mi vida. A mis Hermanos Deysdania, Sergio, Roosevet, Bryan, por ser mi apoyo fundamental e incondicional en los momentos muy difíciles de mi vida profesional y emocional, por ser un ejemplo a seguir y por sobre todas las cosas, por amar a Dios. Los llevo siempre en el corazón.*

*Milton*

**TABLA DE CONTENIDO**

|  |  |
| --- | --- |
| **ÍNDICE DE TABLAS………………………………………………………………………..** | **vii** |
| **ÍNDICE DE FIGURAS………………………………………………………………………** | **viii** |
| **ÍNDICE DE GRÁFICOS…………………………………………………………………....** | **ix** |
| **RESUMEN……………………………………………………………………………………** | **x** |
| **SUMARY……………………………………………………………………………………..** | **xi** |
| **INTRODUCCIÓN……………………………………………………………………………** | **1** |
| **CAPITULO I** |  |
| **MARCO TEÓRICO** |  |
| **1.1 Servicios web…………………………………………………………………………** | **8** |
| **1.1.1 *Definición*…………………………………………………………………………….** | **8** |
| **1.1.2 *Protocolos de los servicios web XML………………………………………………..*** | **9** |
| **1.1.3 *Ventajas y Desventajas de los Servicios Web………………………………………...*** | **10** |
| **1.2 Servicios web SOAP…………………………………………………………………** | **11** |
| **1.2.1 *Definición*…………………………………………………………………………….** | **11** |
| **1.2.2 *Objetivos de SOAP*…………………………………………………………………...** | **11** |
| **1.2.3 *Estructura de un mensaje*…………………………………………………..............** | **12** |
| **1.2.4 *Lenguaje de Marcado Extensible (XML)* …………………………………………** | **13** |
| **1.2.5 *Lenguaje de definición de servicios web WSDL.*…………………………................** | **14** |
| **1.2.6 *UDDI..*.……………………………………………………………………………….** | **15** |
| **1.2.7 *Protocolo HTTP*……………………………………………………………………..** | **15** |
| **1.2.8 *Tipos de datos en SOAP*……………………….…………………………………….** | **17** |
| **1.2.9 *JAX – WS*………………………………………………………………………….....** | **18** |
| **1.3 Servicios Web Restfull……………………….……………………………………...** | **19** |
| **1.3.1 *Definición*………………………………………………………………………….....** | **19** |
| **1.3.2 *Principios*………………………………………………………………………..……** | **19** |
| **1.3.3 *Características*…………………………………………………………………….…** | **22** |
| **1.3.4 *Web application Description Languaje (WALD)*.…………………………….……** | **22** |
| **1.3.5 *Jersey...*…………………………………………………………………………..…..** | **22** |
| **1.3.6 *Cuadro comparativo de los servicios web SOAP vs RESTFULL*…………………** | **24** |
| **1.4 Servicios web móviles……………………………………………………………....** | **25** |
| **1.4.1 *Tipos de MWS* …………………………………………………………………….....** | **26** |
| **1.4.2 *Fragmentación de Dispositivos*………………………………………………………** | **27** |
| **1.5 Realidad Aumentada……………………………………………………………..…** | **27** |
| **1.5.1 *Sistemas de localización de dispositivos móviles*…………………………………….** | **28** |
| **1.5.2 *GPS*……………………………………………………………………………………** | **28** |
| **1.5.3 *Android Location Services*…………………………………………………………...** | **28** |
| **1.5.4 *Google Maps*…………………………………………………………………............** | **29** |
| **1.5.5 *Qué es realidad aumentada*………………………………………………………….** | **29** |
| **1.5.6 *Realidad aumentada en móviles*………………………………………………..........** | **30** |
| **1.5.7 *Registro de objetos virtuales*………………………………………………………....** | **30** |
| **1.5.8 *Realidad aumentada basada en marcadores*……………………………………….** | **30** |
| **1.5.9 *Realidad aumentada basada en reconocimiento de objetos*………………………..** | **30** |
| **1.5.10 *Realidad aumentada basada en posición y orientación del dispositivo*……………** | **31** |
| **1.5.11 *APIS de realidad aumentada*………………………………………………………..** | **31** |
| **1.6 Android………………………………………………………………………………** | **32** |
| **1.6.1 *Características* ………………………………………………………………………** | **32** |
| **1.6.2 *Arquitectura Android* ………………………………………………………………** | **32** |
| **CAPÍTULO II** |  |
| **ANÁLISIS COMPARATIVO** |  |
| **2.1 Definición de parámetros a comprobar………………….…………………………** | **35** |
| **2.2 Determinación de las variables de comparación………………..…………………** | **35** |
| **2.3 Ambientes de pruebas………………………………………………….……………** | **36** |
| **2.4 Prototipo de los servicios web ………………………………………………..……..** | **38** |
| **2.4.1 *Prototipo móvil*………………………………………………………………….…….** | **39** |
| **2.5 Desarrollo de los prototipos………………………………………………………….** | **40** |
| **2.5.1 *Servicios web RestFull*…………………………………………………………….….** | **40** |
| **2.5.2 *Cliente RestFull*…………………………………………………………………..…..** | **41** |
| **2.5.3 *Servicios web SOAP*…………………………………………………………….…….** | **43** |
| **2.5.4 *Cliente SOAP*………………………………………………………………….………** | **44** |
| **2.6 Planteamiento de hipótesis…………………………………………………..….…..** | **46** |
| **2.7 Análisis comparativo……………………………………………………………...…** | **46** |
| **2.8 Pruebas y resultados………………………………………………………………...** | **47** |
| **CAPITULO III** |  |
| **DESARROLLO DE LA APLICACIÓN PARA TURISMO EN LA PROVINCIA DE CHIMBORAZO** |  |
| **3.1. Metodología de desarrollo…………………………………………………………** | **54** |
| **3.1.1 *Estudio Preliminar*…………………………………………………………………..** | **54** |
| **3.1.2 *Planificación*…………………………………………………………………………** | **54** |
| **3.1.3 *Product Backlog*……………………………………………………………………..** | **56** |
| **3.1.4 *Sprint Backlog* ……………………………………………………………………….** | **59** |
| **3.1.5 *Sprint 1*. ……………………………………………………………………………...** | **62** |
| **3.1.6 *Sprint 2*……………………………………………………………………………….** | **66** |
| **3.1.7 *Sprint 3*……………………………………………………………………………….** | **67** |
| **3.1.8 *Sprint 4*……………………………………………………………………………….** | **69** |
| **3.1.9 *Sprint 5*…………………………………………………………………………….…** | **71** |
| **3.1.10 *Gestión del proyecto*…………………………………………………………………** | **73** |
| **CONCLUSIONES…………………………………………………………………………...** | **75** |
| **RECOMENDACIONES………………………………………………………………….....** | **76** |

**BIBLIOGRAFIA…………………………………………………………………………… 80**

**ANEXOS…………………………………………………………………………………….. 82**

**ÍNDICE DE TABLAS**

[Tabla 1-1 Protocolos de Servicios Web 9](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535593)

[Tabla 2-1 Tipos de datos SOAP 17](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535594)

[Tabla 3-1 Tipos MIME 22](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535595)

[Tabla 4-1 Características de SOAP vs RestFull 25](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535596)

[Tabla 5-1 Comparación de SOAP vs RestFull 25](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535597)

[Tabla 6-1 Arquitectura Android 33](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535598)

[Tabla 1-2 Descripción de Parámetro 35](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535599)

[Tabla 2-2 Variable Independiente 35](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535600)

[Tabla 3-2 Equivalencias 48](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535601)

[Tabla 4-2 Valores Estadísticos RESTFULL Obtenidos 49](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535602)

[Tabla 5-2 Valores Estadísticos SOAP Obtenidos 50](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535603)

[Tabla 6-2 Resultados Estadísticos SOAP RESTFULL 50](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535604)

[Tabla 7-2 Muestras Relacionadas 51](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535605)

[Tabla 1-3 Product Backlog 56](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535606)

[Tabla 2-3 Sprint Backlog 60](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535607)

[Tabla 3-3 Tarjeta de Tareas 62](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535608)

**ÍNDICE DE FIGURAS**

[Figura 1-1 Estructura de un mensaje SOAP 12](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535616)

[Figura 2-1 Estructura WSDL 14](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535617)

[Figura 3-1 Proceso de JAX-WS 18](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535618)

[Figura 1-2 Ambiente de Pruebas 36](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535619)

[Figura 1-3 Tarjeta de Usuario 60](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535620)

**ÍNDICE DE GRÁFICOS**

[Gráfico 1-2 Histograma RestFull 49](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535629)

[Gráfico 2-2 Histograma SOAP 49](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535630)

[Gráfico 3-2 Prueba Bilateral de Dos Colas 52](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535631)

[Gráfico 1-3 Diagrama Entidad Relación 64](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535632)

[Gráfico 2-3 Metas Cumplidas del Primer Sprint 66](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535633)

[Gráfico 3-3 Metas Cumplidas del Segundo Sprint 68](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535634)

[Gráfico 4-3 Metas Cumplidas del Tercer Sprint 69](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535635)

[Gráfico 5-3 Metas Cumplidas del Cuarto Sprint 71](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535636)

[Gráfico 6-3 Metas Cumplidas del Quinto Sprint 73](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535637)

[Gráfico 7-3 Gráfica de Seguimiento del Proyecto 74](file:///E:\Tesis%20SOAP%20RESTFULL.docx#_Toc443535638)

**RESUMEN**

Los servicios web proveen de información para mejorar la escalabilidad de las aplicaciones, y la interoperabilidad entre ellas, esta información es presentada en cualquier medio, para ello se considera realizar un análisis comparativo entre dos tipo de servicios y que muestran diferencias en su implementación, RestFull con Jersey y SOAP con JAX-WS, el objetivo es analizar el rendimiento entre estos dos servicios y seleccionar el que presente un menor tiempo de respuesta. Los experimentos fueron realizados en un dispositivo móvil con sistema operativo Android el cual consume servicios web desde el servidor de aplicaciones Glassfish. Los resultados de estos experimentos muestran que los servicios web RestFull con Jersey tienen un promedio con una diferencia del 5.88% menos que el promedio del tiempo de respuesta obtenido en SOAP JAX-WS, la investigación muestra que RestFull tiene mejor rendimiento que SOAP al momento de implementarlo en un dispositivo Smartphone mostrando eficiencia y mejorando la experiencia del usuario al momento de consumir un servicio. Los servicios web RestFull son implementados para la creación del sistema de turismo de la Provincia de Chimborazo por su rendimiento en el consumo de los servicios web. Se recomienda la implementación de este estudio para la creación y consumo de servicios web en dispositivos móviles.

PALABRAS CLAVE:

<SERVICIOS WEB><SOAP [SIMPLE OBJECT ACCESS PROTOCOL]>< RESTFULL [REPRESENTATIONAL STATE TRANSFER]><ANDROID><JAX-WS><JERSEY>< JSON [JAVASCRIPT OBJECT NOTATION]>

**SUMARY**

Web services provide information to improve the scalability of applications, and interoperability between them, this information is presented in any medium, this is considered to perform analysis between two kinds of services and that show differences in their implementation, Restfull whit Jersey and SOAP with JAX-WS, the objective is to analyze performance between these two services and select which present a lower response time.

Experiments were performed on a mobile device with Android operating system which consumes web services from the Glassfish application server. The results of these experiments show that Jersey with Restfull web services have an average with a difference of the 5.88% less than the average of the response time obtained in JAX-WS SOAP, research shows that Restfull has better performance than SOAP at the time of implementing it on a device Smartphone showing efficiency and improving the user experience when consuming a service. Restfull web services are implemented for the creation of the system of tourism of the province of Chimborazo by his performance in the consumption of web services. The implementation of this study to the creation and consumption of web services in mobile devices is recommended.

# CAPITULO I

**INTRODUCCIÓN**

La máxima expresión de las tecnologías de información se define en un contexto digital accesible y manipulable, dirigiendo información continua y verificable mediante sistemas funcionales, didácticos, inteligentes, prácticos, sencillos y modernos, este punto divergente en el desarrollo tecnológico muestra el resultado de años y años de evolución de métodos y procedimientos para acceder a la información, presentando diversas opciones que se adaptan a las necesidades de los diferentes tipos de usuario, renovando la manera en que la información llega hasta ellos.

La intención de subir a ese tren llamado tecnología no es más que buscar medios para propiciar el desarrollo colectivo, cultural y educativo de la sociedad en que nos desenvolvemos, en tiempos remotos en que se pretendía divulgar un producto o únicamente proveer información a usuarios, buscando y creando aparece la tecnología web, este medio resultó inicialmente ser una interesante técnica para que la información de alguna manera sea accesible a usuarios privilegiados, de aquí en adelante la utilización de este medio se ha convertido en la base de la comunicación y propagación de la información que hasta el día de hoy presenta aplicaciones dirigidas a campos específicos como redes sociales, videos, enciclopedias en general, juegos, etc. impulsando la investigación en áreas como medicina, turismo, electrónica, mercados, entre otros, que en conjunto con el desarrollo de dispositivos móviles inteligentes buscan disminuir tiempo de respuesta y aumentar productividad impulsando a empresas, buscando un equilibrio en el desarrollo de que forma parte de nuestro diario vivir.

Los llamados servicios web que mediante métodos y funciones distribuyen programas a los usuarios, utilizan protocolos y tecnologías que son estándares en internet para diseño, solicitud y transmisión de documentos, en este aspecto para establecer la comunicación entre un servidor de aplicaciones web y un dispositivo móvil inteligente se busca la mejor tecnología para que el rendimiento del dispositivo móvil no se vea afectado, teniendo en cuenta que las aplicaciones integran diversos servicios como los de ubicación mediante GPS, Mapas, mencionado también una nueva tecnología que utilizada realidad aumentada mediante geoposicionamiento, esto y más puede reducir el rendimiento o en peores casos colapsar la memoria de un dispositivo móvil.

Por esto se considera importante investigar los protocolos de comunicación SOAP y RESTFULL, aplicado a dispositivos móviles con sistema operativo Android, presentado la información general del presente proyecto, en el primer capítulo definiciones y características de las tecnologías de comunicación mencionadas anteriormente. En el segundo capítulo se definen los parámetros de comparación para medir, analizar, y comprobar la hipótesis, finalmente el cuarto capítulo contiene el desarrollo de una aplicación web y móvil dirigida al turismo de la provincia de Chimborazo.

**Formulación general del proyecto de tesis**

**Antecedentes**

La nueva generación de dispositivos móviles ya no presentan las restricciones con las que inicialmente fueron desarrollados, en hardware integran mecanismos de localización, cámaras cada vez con mayor resolución, acelerómetros, etc., en software presentan mejoras en la capacidad de almacenamiento y ejecución de aplicaciones debido al control de procesos de memoria. Las aplicaciones presentadas mediante la web difieren cuando los usuarios las consumen desde una PC y desde un Smartphone, sin embargo en ninguno de los dos casos los servicios dejarían de ser funcionales, rápidos, seguros, actuales, dinámicos con la diferencia en un diseño perpendicular y acoplable para móviles.

La versatilidad que demuestran los servicios web de presentarse a cualquier tipo de usuario sin importar la plataforma en hardware y software que este utilice se debe a que las tecnologías y protocolos utilizados son estándares, de aquí que para la comunicación es utilizado el protocolo SOAP (Simple Object Access Protocol) que utiliza el formato de mensaje lo que permite adaptarse para satisfacer la variedad de necesidades que se pueden presentar para servicios de mayor complejidad, una especificación de servicios que define este estándar es el API para java JAX-WS.(Snell, Tidwell, & Kulchenko, 2010, p. 78), Por otro lado los servicios REST (Representational State Transfer), han desarrollado un estilo más sencillo de usar por ello está teniendo un gran impacto ante los servicios tradicionales SOAP que por su formato de cabecera que lleva en los mensajes es bastante seguro en comparación con REST.

REST, por su innovación se crea una versión para java denominada REST(FULL), que está orientada a recursos, con cuatro operaciones principales basadas HTTP que son: GET, POST, DELETE, UPDATE. En esta arquitectura se utiliza el API Jersey que permite trabajar con servicios web. Desde que REST salió a la luz, existe un debate en comparación con SOAP. Ambos difieren en muchos aspectos comenzando porque REST fue concebido en el ámbito académico y SOAP es un estándar en la industria, creado por un consorcio del cual Microsoft formaba parte. Según Paul Prescod programador independiente que escribió el libro más importante sobre la familia de los estándares: THE XML HANDBOOK, de alguna manera el modelo SOAP volvería a ser suficientemente bueno y el uso de REST se limitaría a ser el hipertexto de la web, otro posible futuro podría ser que la metodología SOAP revelaría paulatinamente sus defectos hasta que fuesen intratables, en este caso REST o algo parecido se convertiría en una arquitectura dominante. En un tercer escenario, encontraríamos una manera de que ambas arquitecturas trabajasen juntas, tratando cada una de resolver diferentes problemas e incluso podrían interoperar.(Cabrera, 2013, p. 74)

La incertidumbre sobre qué servicio resulta en la actualidad más conveniente aumenta cuando se requiere un servicio de aplicaciones para dispositivos móviles, pues estos requieren que los servicios sean ágiles y ligeros, en este caso se referencia a Android como tecnología de desarrollo y la base para investigar, aplicar métodos e interpretar resultados para toma decisiones sobre qué servicio web será el apropiado para este tipo de interacción con el usuario. La tecnología Android desarrollada en java tiene el lema “desarrolla una vez y ejecuta varias veces”, quiere decir que las aplicaciones funcionan en cualquier versión de dispositivo móvil inteligente. La posibilidad de utilizar al máximo las características que presenta los dispositivos actuales, nos permite adicionar uno de los servicios más interesantes que es la realidad aumentada considerada como un tipo de tecnología donde la visión de la realidad se amplía con elementos virtuales, mostrado además información digital. Existen aplicaciones de realidad aumentada que ganaron mucha popularidad desde su introducción muchas basadas en localización como Wikitude, Layar y Junaio.

Wikitude es una aplicación de Realidad Aumentada basada en geoposicionamiento que usa un diseño de pestañas como patrón para cambiar entre las vistas, recientemente está siendo rediseñado para Android, cuenta con gráficos e impecables iconos y textos en inglés. (Wikitude, 2015. www.kikitude.com)

Actualmente no se cuenta con un análisis de rendimiento de los servicios web SOAP con la API JAX-WS; con el servicio web RESTFULL con la API JERSEY, para implementar en un sistema móvil Android utilizando Wikitude para geolocalización, puesto que la falta de información sobre que arquitectura aplicar, provoca que los desarrolladores tengan dificultades en elegir la mejor tecnología para crear aplicaciones sea web o móviles, que cumplan con los requerimientos necesarios en el ámbito de desarrollo.

El Consejo Provincial de Chimborazo, se crea el primero de enero de 1946, bajo un análisis somero y efectivo de los gobernantes de aquella época, como medio de solución a las necesidades y requerimientos de los pueblos más alejados por el Poder Central. Dentro del organigrama institucional esta la Coordinación de Gestión de Fomento Productivo en donde se encuentra la unidad de turismo, encargada de gestionar los planes y proyectos turísticos de la provincia.

La provincia de Chimborazo se ubica en pleno centro del callejón interandino ecuatoriano conocida como la “provincia de las altas cumbres” como el Altar, el Cariguairazo, los Cubillines siendo la más alta el volcán Chimborazo con altura de 6310 metros que da el nombre a la provincia, el tren Ecuador con la ruta Nariz del Diablo. Las manifestaciones culturales del 6 de Enero del Niño Rey de Reyes, del 2 de noviembre Día de los difuntos, entre otras hacen de la provincia un lugar óptimo para el desarrollo de la actividad turística.

El turismo representa el primer ingreso de divisas en el país, por esa razón se impulsa su desarrollo con proyectos a nivel nacional. La satisfacción de viajar, conocer nuevos lugares, sumergirse en lo desconocido, conocer nuevas culturas, hace que la provincia de Chimborazo sea un destino turístico diverso pero con la necesidad de dar a conocer la gran cantidad de información en todo momento de aquellas zonas y lugares propicios para el turismo. La importancia de contar con herramientas para satisfacer la necesidad de divulgar la información, portable y oportuna lleva a crear una aplicación móvil con Wikitude para georreferenciación de lugares turísticos de la Provincia.

## Justificación del proyecto de tesis

### Justificación investigativa

El desarrollo de la tecnología móvil ha ido evolucionando a pasos agigantados, es por ello que con la aparición de los Smartphone en la actualidad muchas son las actividades u operaciones que una persona puede hacer, el incremento de las aplicaciones para Smartphone hoy en día marca un hito importante, pues cuán más es necesario la utilización de las mismas agilitando las operaciones de manera portable y ágil para los usuarios. Los servicios web junto a los dispositivos móviles tienen gran importancia puesto que presentan una buena interoperabilidad, su alcance va más allá de barreras locales y nacionales, permite proporcionar atención a tiempo a los usuarios desarrollando un diseño adaptativo para todo tipo de dispositivos.

El presente estudio comparativo entre los servicios web SOAP JAX-WS y servicio web RESTFULL JERSEY determinará qué servicio es el mejor en rendimiento la hora de implementar una aplicación Android, lo cual servirá como apoyo a los desarrolladores informáticos en sus implementaciones. El proyecto se encuentra dentro de la línea de investigación de la ESPOCH en el área TIC’S y la líneas y programa TIC y procesos industriales literal (e). Programa de conectividad y telecomunicaciones para la sociedad de la Información y conocimiento. Dentro del Plan Nacional del Buen vivir cumple con el objetivo 11. Asegurar la soberanía y eficiencia de los sectores estratégicos para la transformación industrial y tecnológica, con su política y lineamientos estratégicos 11.3 Democratizar la prestación de servicios públicos de telecomunicaciones y de tecnologías de información y comunicación (TIC), incluyendo radiodifusión, televisión y espectro radioeléctrico, y profundizar su uso y acceso universal.

### Justificación metodológica

Para el reciente estudio se implementará el método científico que tiene su base y postura sobre la teoría mecanicista (todo es considerado como una máquina, que se dividirá en partes pequeñas que permitan estudiar, analizar y comprender sus nexos, interdependencia y conexiones entre el todo y sus partes) el cual será implementado en la parte investigativa. Para la parte aplicativa se desarrollará dos prototipos, el primero enfocado a los servicios SOAP y el segundo a los servicios RESTFULL, cada uno implementado en una aplicación Android, el proceso se alineará de acuerdo a la metodología Scrum en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible de un proyecto. La parte aplicativa implementará esta metodología, la cual constará de un análisis del problema, un listado de requerimientos y los hitos o entregables.

### Justificación práctica

El trabajo a realizar pretende solucionar algunos de los problemas derivados de la necesidad de divulgar información relevante de los lugares de importancia por medio de una aplicación Android con Wikitude aplicada a la gestión de información geolocalizada del turismo en la Provincia de Chimborazo. La creación de una aplicación móvil de georreferenciación con realidad aumentada utilizando Wikitude en Android, se dispone a mejorar de forma adecuada y ágil el turismo de nuestra provincia con lo que daremos a conocer las tradiciones, nuestro pasado y nuestra cultura.

Para ello se considera importante desarrollar el sistema con la función principal de ubicación con realidad aumentada gracias a la tecnología Wikitude añadiendo todos los puntos de interés de un cantón, luego la ruta a seguir para llegar al destino, ubicación en el mapa, si escogemos un lugar específico se mostrará información detallada. Todo esto en el módulo de visualización a través de una aplicación móvil. Luego el módulo para la administración del sitio con operaciones de ingreso, modificaciones y eliminación de información de las principales características, todo esta información será mostrada en un dispositivo móvil Android gracias a los servicios web, que permitirá al usuario la posibilidad de interactuar con ellos.

**Objetivos**

**Objetivo general**

Realizar el análisis comparativo de los servicios web SOAP JAX-WS Y RESTFULL JERSEY, para el desarrollo de una aplicación Android con Wikitude aplicada a la gestión de información geolocalizada del turismo de la provincia de Chimborazo.

**Objetivos específicos**

* Realizar el Estudio de servicios web basados en el protocolo SOAP y arquitectura RESTFULL, para aplicaciones móviles Android
* Establecer herramientas y parámetros de evaluación de servicios web para determinar el rendimiento de una aplicación móvil.
* Realizar el análisis comparativo de los servicios web SOAP con la API JAX-WS y RESTFULL con la API JERSEY.
* Desarrollar el sistema web de administración y la aplicación móvil con Wikitude aplicada a la gestión de información geolocalizada del turismo de la provincia de Chimborazo.

**Hipótesis**

La utilización de los servicios web RESTFULL JERSEY ofrecen un mayor rendimiento que los Servicios Web SOAP JAX-WS aplicado al sistema de turismo del Consejo Provincial de Chimborazo.

# CAPITULO I

**MARCO TEÓRICO**

**1.1 Servicios web**

Los servicios web son la principal estrategia o herramienta para que los usuarios conozcan una idea sobre un producto o utilizada ampliamente para la autoeducación, muchos pueden ser los beneficios que aporta esta tecnología, en la actualidad prestar servicios ya no depende de que el servidor y el cliente cuenten con la misma plataforma, además se redujo al mínimo la inconsistencia en la comunicación que antes reprimían a usuario explotar al máximo los beneficios que ofrecía esta tecnología. En el siguiente capítulo se detallará concepto, y características de un servicio web además de las ventajas que tiene esta tecnología y las plataformas que conforman la presente investigación.

***1.1.1 Definición***

Para definir que es un servicio web se toma como referencia el concepto que propone World Wide Web Consortium (W3C, 2014, p 3) que define al servicio web como una aplicación software identificada por un URI (Uniform Resource Identifier), cuyas interfaces se pueden definir, representar y descubrir mediante documentos XML, esto hace posible la interacción de aplicaciones, utilizando mensajes XML, invocados mediante protocolos estándares en internet. Los servicios web exponen funcionalidades que son enviados y recibidos por un agente, en el caso de la persona u organización que envía y recibe mensajes pueden ser proveedor (proporciona un agente para implementar un servicio) o solicitante (utiliza el servicio que proporciona el agente del proveedor). Además de web presentar aplicaciones informáticas mediante tecnologías y protocolos web estándares, proporcionan mecanismos de comunicación para presentar información dinámica al usuario, aquí la interoperabilidad va más allá de la capacidad de intercambiar información entre dos máquinas diferentes sino que también proporciona mecanismos para que los servicios sigan presentando la misma funcionalidad aunque los agentes hayan cambiado.

### 1.1.2 Protocolos de los servicios web XML

Para que los servicios web se integren y complementen utilizan algunos protocolos en el proceso de comunicación y transmisión de datos, estos protocolos utilizados son estándares en la web, esto quiere decir que todos estos normalizan esfuerzos para que los servicios lleguen a cualquier usuario que los requiera. Estos protocolos cumplen con una función específica y son los siguientes:(Jerez Fernández, 2015, p. 13)

* [XML (eXtensible Markup Language)](http://www.fisica.uson.mx/carlos/WebServices/WS_XML.htm), representa la forma en que estructuramos, describimos e intercambiamos información, derivada de la tecnología HTML (HyperText Markup Language) y SGML (Standard Generalized Markup Language), tiene la ventaja de que todas las tecnologías de servicios web están basadas en XML. El formato de datos estándar y flexible que ofrece reduce las opciones de utilizar más tecnologías para realizar la misma operación.
* [UDDI (Universal Description, Discovery and Integration)](http://www.fisica.uson.mx/carlos/WebServices/WS_UDDI.htm), utiliza un tipo de directorio sección amarilla donde se puede registrar una empresa, y publicar sus servicios y así poder realizar transacciones, este registro de lo realiza a través de mecanismos basados en XML y HTTP(S), su función es describir, publicar y encontrar servicios web.
* [SOAP (Simple Object Access Protocol)](http://www.fisica.uson.mx/carlos/WebServices/WS_SOAP.htm) este protocolo nos indica cómo realizar el intercambio de mensajes proporcionando un mecanismo de referencia al mensaje mediante el uso de cabeceras, además se encarga de establecer la comunicación con la invocación de objetos que residen en sistemas remotos.
* [WSDL (Web Service Description Language)](http://www.fisica.uson.mx/carlos/WebServices/WS_WSDL.htm), describe los servicios web además nos muestra como podemos acceder a ellos y consiste en un lenguaje de definición de interfaz (IDL - Interface Definition Language) es apuntado en los registros UDDI y describe los mensajes SOAP que definen un servicio Web en particular.

En la tabla 1-2 se muestra el funcionamiento de los protocolos utilizados por los servicios web.

Tabla 1-2 Protocolos de Servicios Web

|  |
| --- |
| UDDIDirectorio de servicios |
| WSDLDescribe los servicios |
| SOAPInvocación de servicios a través de métodos |
| XMLEstructura del mensaje. Datos |
| HTTP, SMTP, TCPProtocolos de transporte |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

**Fuente:** http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb

Estos protocolos hacen posible que los clientes con características diferentes puedan publicar y obtener servicios esto porque son estándares en diseño, solicitud y transmisión de información. Existen más alternativas a los protocolos ya mencionados anteriormente, es importante investigar sobre las ventajas y funcionalidades que ofrecen estos protocolos puestos que algunas de ellos no serán solo alternativas sino que también complementarias.

### 1.1.3 Ventajas y Desventajas de los Servicios Web

Es inevitable mencionar las ventajas de los servicios web, a pesar de tener una noción de cuán importante se han vuelto este tipo de tecnología.(Carmona Barbero, 2013, p. 75)

* Principalmente una de las ventajas es que los servicios web aseguran interoperabilidad entre los servicios sin importar la plataforma en que estas son desarrolladas, esto debido a que utiliza XML para definir mensajes.
* Un concepto importante que se maneja en SOAP es que evitan problemas de firewall por utilizar HTTP como protocolo de comunicación.
* Además de facilitar la comunicación SOAP, dirige y orienta de la mejor manera la invocación de métodos y funciones de los servicios.
* Las herramientas utilizadas para la creación de servicios web actualmente son de fácil implementación, lo que produce un fácil, rápido y mejor desarrollo de nuevas aplicaciones por supuesto con la mejor tecnología que apunta a reducir recursos y mejorar beneficios.
* Los servicios web permiten centralizar los datos, distribuirlos sobre internet y con las nuevas herramientas pueden ser accedidos a través de una gran variedad de dispositivos.

Definitivamente los servicios web apuntan a un desarrollo ágil, independiente, que ha cambiado el tradicional software empaquetado a un servicio que puede ser accedido a través cualquier máquina.

Las ventajas mencionadas sobre los servicios web están consideradas según los protocolos más utilizados para la web, sin embargo los nuevos servicios que utilizan los dispositivos móviles específicamente dirigen un interés especial ya no solo por obtener un servicio sino por integrarlos sin perder agilidad, dinamismo, seguridad, etc., por esto es recomendable escoger la mejor tecnología de acuerdo al el servicio que el usuario a utilizar.

## 1.2 Servicios web SOAP

Los servicios ofrecidos mediante la web necesitan poder comunicarse mediante protocolos, con un marco de trabajo efectivo y un modelo de computación distribuida de objetos; para ello ese presenta uno de los objetos de estudio como es SOAP (Simple Object Access Protocol), que definiremos a continuación.

### 1.2.1 Definición

SOAP es considerado como el formato para definir el intercambio de datos XML entre dos usuarios, independiente de la plataforma o lenguaje de programación de una forma simple y ligera mediante un modelo de empaquetado de datos modular y una serie de mecanismos de codificación de datos.(Silva de la Cruz & Silva de la Cruz, 2013, p. 24). Su estructura cuenta con variadas especificaciones y extensiones como son la seguridad, formato de entrega, procesamiento del mensaje, enrutamiento, etc. Es considerado junto con el lenguaje de definición de servicios WSDL, un estándar, completamente dependiente del formato XML para la codificación de datos suponiendo una sobrecarga de trabajo para su procesado, también para la transmisión de datos el protocolo HTTP(Hiper Text Transport Protocol), diseñado para trabajar bajo el esquema RPC, invocando funciones remotas.

Las tecnologías que implementa SOAP hace que sea muy difícil ser adaptado a un ambiente móvil porque hasta ahora se conoce que son limitados y consumen muchos recursos, el encapsulado del mensaje que envía hace que consuma un mayor ancho de banda, requiere de más memoria y en este caso en procesamiento de un dispositivo.

### 1.2.2 Objetivos de SOAP

Para crear este modelo distribuido se considera algunos objetivos que permiten cumplir con el marco de trabajo con las que SOAP es considerado un modelo estándar e independiente. Estos son:

* Establecer un protocolo estándar de invocación a servicios remotos que estébasado en protocolos estándares de uso frecuente en Internet, como sonHTTP (Hiper Text Transport Protocol) para la transmisión y XML(eXtensible Markup Language) para la codificación de los datos.
* Independencia de plataforma hardware, lenguaje de programación e implementación del servicio Web.

La utilidad que presta SOAP es considerablemente útil por los protocolos ligeros y estándares que este utiliza para la conexión y transmisión de datos.

### 1.2.3 Estructura de un mensaje

El formato de mensajes SOAP define una estructura de sobre o envoltura que se compone de un encabezado o cuerpo, los datos con los que se define el encabezado aumenta de la funcionalidad del mensaje tales como direccionamiento, seguridad y mensajería confiable. El cuerpo contiene datos a ser transmitidos y soporta bloques de elementos XML, texto u otro contenido como se muestra e la figura 1-2

SOAP header

Bloque de cabecera

Bloque de cabecera

SOAP envelope

SOAP body

Cuerpo del mensaje

Figura 1-2: Estructura de un mensaje SOAP

**Realizado por:** Milton Jiménez y Lorena Hidalgo

**Fuente:** (Carmona Barbero, 2013)

**Encabezado del mensaje**

El encabezado es opcional pero en un menaje SOAP es de gran ayuda para su flexibilidad y el control que lleva de los bloques de encabezado. Para un recurso en REST no es de interés llevar un encabezado puesto que una de sus ventajas es utilizar lo mínimo en recursos.

Esta información puede tener modificaciones en toda su vida útil, refiriéndose al modelo SOAP que permite el tránsito por puntos intermedios.

**Cuerpo del Mensaje**

En esta sección se transportan los datos en XML simples, texto o estructuras complejas que no sean binaria o multimedia, aprovechando al máximo las funcionalidades de la codificación XML.

**Código especial adjunto**

Tratar de transmitir contenido misceláneo, como documentos multimedia y binario es muy costoso en términos de procesamiento y tamaño de datos, para ello SOAP crea una estructura compuesta, que separa al contenido especial y al mensaje en dos secciones bien definidas.

Para transportar este tipo de contenido SOAP utiliza el estándar Extensiones Multipropósito de Correo de Internet (MIME), para adjuntar documentos al mensaje

### 1.2.4 Lenguaje de Marcado Extensible (XML)

XML es un metalenguaje utilizado para la creación de lenguajes de marcado y para la representación, almacenamiento y estructuración de datos, en SOAP se utiliza para estructurar la información necesaria para los mecanismos de descubrimiento, descripción y mensajería. Es una plataforma independiente lo que permite ser utilizado tanto en SOAP como en REST. Para que un documento XML sea bien definido debe cumplir correctamente con la sintáctica de XML, y para que sea válido debe cumplir son un esquema según la naturaleza del mensaje que se esté estructurando.

Las partes o secciones de un documento XML se definen a continuación(Corales Muñoz Victor, 2012, p. 18)

* *Declaración:* indica la versión de XML utilizada y el tipo de codificación del documento.
* *Elementos:* son datos etiquetados que sirven para la estructuración de tipos de datos e información.
* *Atributos:* los atributos están formados por un nombre y un valor respectivamente, utilizados como variables que almacenan información adicional.
* *Espacios de nombres:* Los espacios de nombres se utilizan para la organización del contenido y son útiles para la importación de contenido y adaptabilidad de SOAP.
* *Comentarios:* los comentarios son anexados al documento para mejorar la legibilidad del mensaje.

### 1.2.5 Lenguaje de definición de servicios web WSDL

El WSDL nos permite tener una descripción de un servicio web. Especifica la interfaz abstracta a través de la cual un cliente puede acceder al servicio y los detalles de cómo se debe utilizar. (Carmona Barbero, 2013, p. 24)

**Elementos del WSDL**

* Types; es el contenedor de definiciones del tipo de datos que utiliza algún sistema de tipos (por ejemplo XSD).
* Message: definición abstracta y escrita de los datos que se están comunicando.
* Operation: descripción abstracta de una acción admitida por el servicio.
* Port Type: conjunto abstracto de operaciones admitidas por uno o más puntos finales.
* Binding: especificación del protocolo y del formato de datos para un tipo de puerto determinado.
* Port: punto final único que se define como la combinación de un enlace y una dirección de red.
* Service: colección de puntos finales relacionados

La estructura que tiene el lenguaje de definición de servicios se observa en la figura 2-2

OPERATIONS

WSDL portType

Tipos de datos

WSDL Types

Servicios

WSDL binding

WSDL service

Mensajes

WSDL Mensajes

Figura 2-2 Estructura WSDL

**Fuente:**(Carmona Barbero, 2013)

**Realizado por** Milton Jiménez y Lorena Hidalgo

Las herramientas de desarrollo construyen el documento WSDL automáticamente, es necesario, que a partir de la versión 2.0 define la interfaz de los servicios y esto hace que es usuario pueda ingresar de manera dinámica a los servicios.

### 1.2.6 UDDI

Universal Description, Discovery and Integration (UDDI), son las siglas de Catálogo de negocios en Internet, es uno de los estándares cuyo objetivo es ser accedidos por los menajes SOAP y dar paso a documentos WSDL, en los que se describe los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.(Carmona Barbero, 2013, p. 72), UDDI constituye un recurso para que los servicios web desarrollados por las empresas se publiquen, a fin de que posibles usuarios interesados en hacer uso de estos servicios para realizar determinadas acciones en sus negocios puedan encontrarlos sin demasiada dificultad.

### 1.2.7 Protocolo HTTP

Hyper Text Transfer Protocol (HTTP). Protocolo cliente/servidor utilizado en la web inicialmente para transferir paginas HTML hoy puede utilizarse para varias peticiones, estableciendo comunicación entre un cliente y un servidor mediante el puerto reservado 80, maneja el esquema cliente/servidor.

Una petición HTTP consta de:

* URL
* Métodos de acceso GET, POST, PUT, DELETE
* Cabecera meta información de la petición
* Cuerpo del mensaje

**Métodos de acceso**

**GET:** solicita una representación del recurso especificado. No debe causar modificaciones en el recurso.

**POST:** envía datos para que sean procesados al recurso indicado. Puede crear un nuevo recurso, modificarlo si existente o ambas cosas.

**PUT:** carga en el servidor una representación de un recurso.

**DELETE:** elimina un recurso especificado.

De estos métodos las peticiones delete, post, put a múltiples peticiones deben tener el mismo efecto que una sola; las peticiones get no tienen cuerpo del mensaje, se especifican con parámetros dentro de la URL como pares campo=valor separados por el carácter ´&´, no deben causar modificaciones en los recursos; y las peticiones post llevan datos en el cuerpo del mensaje, este puede crear nuevos recursos, o modificar un recurso ya existente. Se debe tener muy en cuenta las funciones que cumple cada método.

**Repuesta HTTP**

Una respuesta HTTP contiene:

* **Códigos de status**
* 200: ok
* 404: recurso no encontrado
* 500: error en el servidor
* 403: erros de autorización, etc
* **Cabeceras:** meta información de la respuesta.
* **Cuerpo del mensaje**
* Representación del recurso invocado o mensaje de error.

**Cabecera HTTP**

Especifica:

* El tipo de datos esperado
* Codificación esperada
* Lenguaje esperado
* Antigüedad de la repuesta
* Control de cache
* Credenciales de autorización
* Información para proxies
* Información ara autenticación
* Agente de usuario ejm. navegador utilizado.

SOAP está basada en protocolos estándares, como XML para definir el lenguaje, WSDL utilizado para describir el servicio web, UDDI para la publicación del servicio, HTTP para transporte, además de que ese último protocolo proporciona los métodos básicos para referirse a las funciones de la base de datos conocidos como CRUD(Create, Read, Update, Delete) para definir los servicios web.

### 1.2.8 Tipos de datos en SOAP

Los datos que integran el mensaje SOAP siguen ciertas normas que deben ser cumplidas, estos datos son conocidos como datos simples que pueden ser datos complejos y estructurados pero empaquetados. Esto surge ante la necesidad de realizar un mapeo de los tipos de datos de java y el documento SOAP, la cantidad de tipos de datos es considerable los más destacados se muestran en la tabla 2-1

Tabla 2-1 Tipos de datos SOAP

|  |  |
| --- | --- |
| **Tipo SOAP** | **Tipo Java** |
| SOAP-ENC:int | java.lang.Integer |
| SOAP-ENC:long | java.lang.Long |
| SOAP-ENC:short | java.lang.Short |
| SOAP-ENC:string | java.lang.String |
| SOAP-ENC:boolean | java.lang.Boolean |
| SOAP-ENC:float | java.lang.Float |
| SOAP-ENC:double | java.lang.Double |
| SOAP-ENC:byte | java.lang.Byte |

**Realizado por** Milton Jiménez y Lorena Hidalgo

**Fuente:**(Snell et al., 2010)

La serialización de datos consiste en empaquetar los datos complejos en datos simples, existen otros tipos de datos además de los ya mencionados anteriormente que son tipos mucho más complejos y estos son:

**Structs:** Un structno es más que un elemento que contiene un conjunto de elementos *hijos* almacenados cada uno de ellos en un campo propio.

**Arrays:** Un arrayen un mensaje **SOAP** es representado mediante un elemento cuyo tipo es:

**SOAP-ENC:Array:** Aunque en Java sea obligatorio que dentro de un array haya únicamente elementos del mismo tipo, SOAP no presenta esta restricción, sino que es posible albergar elementos de distintos tipos.

### 1.2.9 JAX – WS

JAX- WS (Java API for XML- Based Web Services )Es un API estándar en java para implementar e invocar Servicios Web en SOAP, permite operaciones asíncronas(no espera respuestas y en lugar de quedarse bloqueado realiza un callback), JAX-WS se utiliza normalmente en combinación con otras tecnologías(Vohra, 2012, p. 12). Java JAX-WS está diseñado para remplazar a Java-RPC(Remote Procedure Call), se comunican a través de XML utilizando llamadas a procedimientos remotos, los desarrolladores utilizan esta API para definir elementos de las clases implicadas en la implementación del servicio y simplificar el desarrollo de los servicios web, además de dar cierto soporte para RestFull aunque no es su objetivo.

JAX – WS especifica un mapping de WSDL a Java proporcionando un compilador que genere stubs (proxies) para invocar servicios web, también proporciona un compilador de Java a WSDL que genere el documento WSDL correspondiente a un interfaz o clase de implementación de Java que está sujeta a ciertas restricciones. El proceso de desarrollo de JAX-WS de define en la figura 3-1

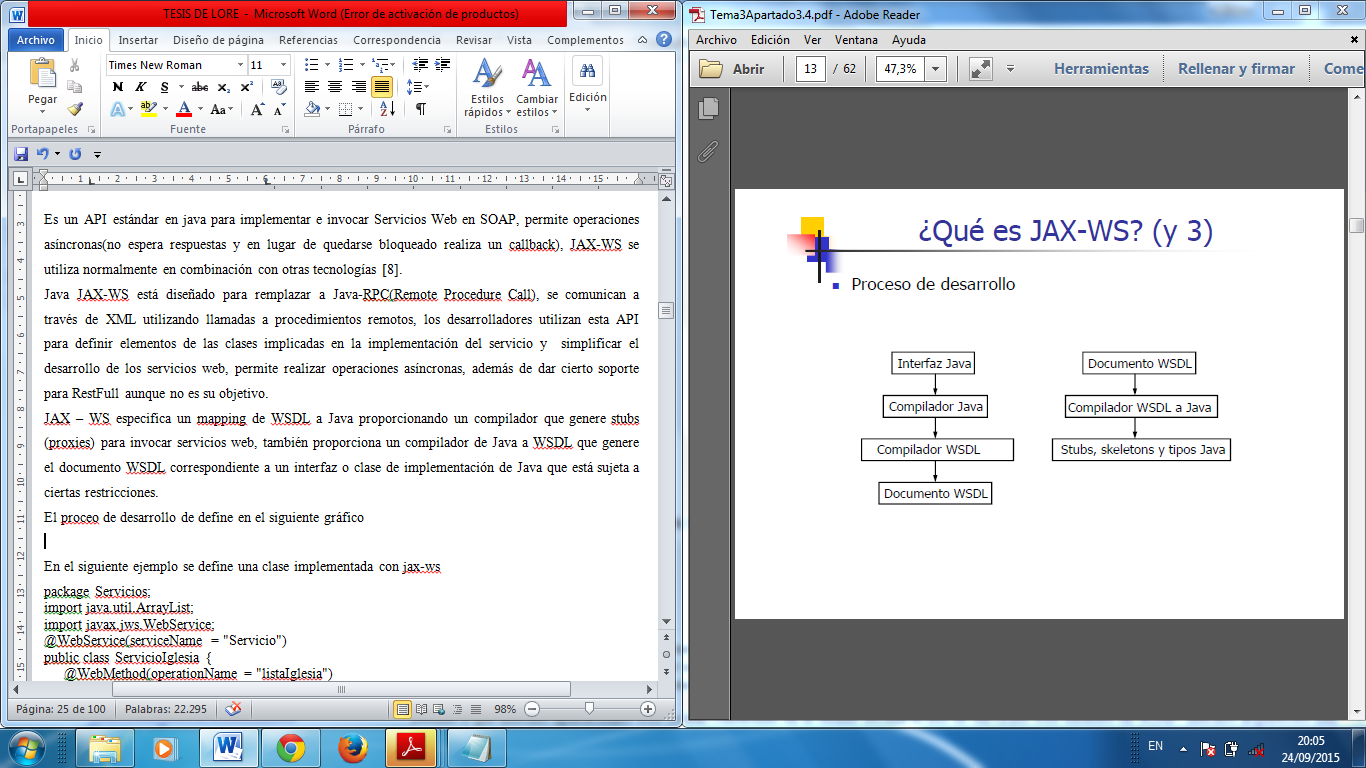


Figura 3-1 Proceso de JAX-WS

**Fuente:**(Burke &Jo, 2013)

**Realizado por**: Milton Jiménez y Lorena Hidalgo

En el siguiente ejemplo se define una clase implementada con jax-ws

package Servicios;

import java.util.ArrayList;

import javax.jws.WebService;

@WebService(serviceName = "Servicio")

public class ServicioIglesia {

@WebMethod(operationName = "listaIglesia")

public ArrayList<Iglesia> listaIglesia() throws Exception {

ArrayList<Iglesia> listaIg=CatalogoIglesia.ListadoIglesia();

return listaIg;

} }

Los servicios web creados con JAX-WS son desplegados como un servlet encargado de gestionar las peticiones HTTP que contenga mensajes SOAP dirigidos al servicio web.

## 1.3 Servicios Web Restfull

Un estilo de arquitectura para sistemas distribuidos de hipermedia está teniendo un avance dentro del desarrollo de aplicaciones web es el modelo de Transferencia de Estado Representacional o REST (RESTFULL como implementación de REST), conocido como un servicio orientado a recursos, ayudando a mejorar y facilitar el trabajo en la web.

### 1.3.1 Definición

El termino REST fue implementado por primera vez por Roy Fielding en una conferencia en el Universidad de California, al tratar de principios arquitectónicos distribuidos. Este tipo de servicios web exponen datos y funcionalidades mediante recursos identificados por URI, los clientes interactúan con los recursos mediante métodos de ingresos. Para Fielding REST es un Estilo Arquitectónico que consiste en clientes y servidores. Los clientes generan solicitudes a los servidores, estos la procesan, generando una respuesta apropiada. Las solicitudes y respuestas se construyen alrededor de la transferencia de representaciones de los recursos. Un recurso puede ser esencialmente cualquier concepto que pueda ser tratado. Una representación de un recurso es típicamente un documento que captura el estado actual o previsto de un recurso(Fielding & Taylor, 2002, p. 82). A diferencia del Protocolo SOAP, REST consume un poco menos el ancho de banda porque no se analiza el documento XML como lo hace SOAP, además de que no requiere de cabeceras en el mensaje. Se puede decir que un servicio web RESTFULL es un diseño basado en la arquitectura REST direccionada a construir aplicaciones distribuidas, orientada a publicar e identificar recursos, utilizando de manera explícita las operaciones del protocolo HTTP y transfiriendo recurso XML y JSON.

### 1.3.2 Principios

REST se define como un estilo arquitectónico, no es un estándar aunque hace uso de varios estándares como son HTTP, XML, URL, HTML entre otros. El diseño de sistemas basados en REST generalmente se denomina RESTFULL y satisfacen los siguientes principios.

* Interfaz uniforme para la identificación de recursos
* Utilización de métodos estándar de HTTP
* Comunicación sin mantener estados
* Recursos con múltiples representaciones
* **Interfaz uniforme para la identificación de los recursos**

La identificación de los recursos en RESTFULL se lo hace a través de URI (Uniform Resource Identifier).

* **Utilización de métodos estándar de HTTP**

HTTP expone los métodos estándar: GET, PUT POST, DELETE, que son utilizados por desarrolladores estableciendo relaciones con las operaciones de Leer, Actualizar, Crear y Borrar de la siguiente manera.(Cabrera, 2013, p. pp 45)

GET: Para obtener un recurso del servidor

PUT: Para cambiar el estado de un recurso o actualizarlo

POST: Para crear un recurso en el servidor

DELETE: Para eliminar un recurso en el servidor.

Es importante tomar en cuenta que en método GET no debe ser utilizado para ejecutar alguna operación sobre el servidor, como se define solo se utiliza para obtener un recurso del servidor no para realizar modificación alguna sobre él.

Los siguientes ejemplos muestran algunos casos prácticos que permiten tener una idea más específica de lo que se describe en la tabla anterior:

1. Este ejemplo logra mediante un HTTP POST adicionar un nuevo recurso a la colección.

*POST /ordenes HTTP/1.1*

*Host: ejemplo.com*

*Content-type: application/xml*

*<orden>*

*<identificador>12480</identificador>*

*<tipo>CI</tipo>*

*</orden>*

2. Este ejemplo logra mediante un HTTP GET obtener una representación del recurso.

*GET /ordenes/12480 HTTP/1.1*

*Host: ejemplo.com*

*Content-type: application/xml*

*<orden>*

*<identificador>12480</identificador>*

*<tipo>CI</tipo>*

*</orden>*

3. Este ejemplo logra mediante un HTTP PUT actualizar el recurso con los nuevos valores.

*PUT /ordenes/12480 HTTP/1.1*

*Host: ejemplo.com*

*Content-type: application/xml*

*<orden>*

*<identificador>11400</identificador>*

*<tipo>CE</tipo>*

*</orden>*

**Comunicación sin mantener estados**

Se exige que el estado sea transformado en estado del recurso y sea mantenido en el cliente. Un servidor no debería guardar el estado de la comunicación de cualquiera de los clientes que se comunican con el más allá de una petición única. La razón más obvia de esto es la escalabilidad, el número de clientes que pueden interactuar con el servidor se ven significativamente afectados si fuese necesarios mantener el estado del cliente.(Cabrera, 2013, p. 56)

De esta forma es mucho mejor que las aplicaciones basadas en HTTP no utilicen cookies para mantener el estado de la sesión pues los servicios REST no lo soportan, la forma de garantizar que los servicios web tengan un mejor rendimiento es incluyendo los parámetros, contexto y datos precisos dentro de la cabecera y cuerpo del HTTP.

**Recursos con múltiples representaciones`**

Los clientes pueden manejar el formato en que desean que sus datos sean devueltos, para ello es necesario utilizar el atributo HTTP Accept en el encabezado del mensaje y definir el tipo en el content-type. Se utilizan tipo MIME (Multipurpose Internet Mail Extensions) como posibles tipos.(Cabrera, 2013, p. 60), en los cuales se retomara la respuesta en dicho formato. Entre los más utilizados se mencionan en la tabla 3-1.

Tabla 3-1Tipos MIME

|  |  |
| --- | --- |
| **Tipos MIME** | **Formato** |
| Application/xml | XML |
| Application/json | JSON |
| Application/xhtml+xml | XHTML |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

**Fuente**:(Carmona Barbero, 2013)

Los diferentes tipos de datos que se pueden utilizar permiten que los servicios puedan ser consumidos desde cualquier otra plataforma.

### 1.3.3 Características

Se pueden identificar las principales características de los servicios web RESTFULL.

* Estos servicios se orientan recursos identificados por un id único.
* Para la transferencia de datos utiliza los formatos XML y JSON.
* La ventaja de estos servicios de no tener un estado es que mejora el rendimiento en cuestión de almacenamiento en memoria cache, además de ser asertivo en ambientes con un bajo ancho de banda.
* Intenta emular las operaciones del protocolo HTTP además de transferir las peticiones.
* Utiliza un URL para identifica cada recurso, los clientes lo invocan directamente.
* Con RESTFULL pueden reusarse directamente los mecanismos de autorización (permisos de acceso), cifrado y autenticación HTTP.

### 1.3.4 Web application Description Languaje (WALD)

Los servicios web RESTFULL necesitan un lenguaje de descripción que se ajuste a sus requisitos, para ello se desarrolló WALD similar al WSDL. El estudio de esa tecnología se debe al incremento del interés que los desarrolladores por utilizar los servicios web.

### 1.3.5 Jersey

Es una librería preparada únicamente para REST, no utiliza el encapsulado de SOAP y de hecho es mucho más limitado que este. Para implementar jersey se necesitan librerías que tienen un peso total de una mega.El principal problema son las versiones que a partir de la versión de java 6 puede empezar a causar problemas, además de no tener documentación se utilizan ejemplos creados con jersey para poder impleméntalo en la aplicación.

En la declaración de un servicio con jersey existe un cambio en la clase del servlet, luego queda configurar el servicio en a clase java, ya no necesita el archivo de configuración service.xml ni el WSDL.

import javax.ws.rs.GET;  
import javax.ws.rs.Path;  
import javax.ws.rs.Produces;  
import javax.ws.rs.QueryParam;  
@Path("/hola")  
public class SimpleService {

@GET

@Produces("text/plain")

public String hola(@QueryParam("nombre") String palabra) {

return "Hola " + palabra;

}

}

**Anotaciones que utiliza Jersey**

**@Path(“/hola”):** el path ayuda en la invocación del servicio. En la configuración de Jersey se declara de la siguiente manera:

**http://servidor:puerto/services/hola?nombre=Pepe**

**@GET:** identifica al método que será accesible mediante una petición http GET.

**@Produces(“text/plain”):** devuelve texto plano

**@QueryParam(“nombre”):** el parámetro nombre mapea al valor del parámetro de entrada al método del servicio web. Esto quiere decir que los parámetros en la URI no tienen que llamarse igual que los parámetros declarados en los servicios.

Es importante recordar que cada clase de un servicio web tiene un solo path es decir una sola ruta para llegar hasta el, este path de define a nivel de clase y no a nivel de método. Así en una clase solo se puede definir un método GET y un método de entrada tipo POST. Eso sí en algunas versiones de servidor en necesario declarar (al menos) todos los métodos del CRUD.

Se diferencia en la sencillez de crear estos métodos, además de no generar un wsdl, es más ligero y fácil de configurar.

### 1.3.6 Cuadro comparativo de los servicios web SOAP vs RESTFULL

Los servicios web pueden implementarse en distintos estilos de arquitecturas, tales como, SOAP(Simple Object Access Protocol) y RESTFULL(servicios web de estilos REST). Estos servicios web indican la forma en que deben enviar resultados, y como se deben publicar o dar a conocer los servicios orientados a mensajes.

En el caso de los servicios web RESTFULL, el cliente realiza un esfuerzo importante para invocar el servicio, mientras que la invocación de servicios en SOAP es mucho más simple. Sin embargo la invocación se servicios RESTFULL consume menos recursos y tiempo para el cliente y proveedor de servicios, que los servicios web SOAP; donde la capa de middleware tiene que ser implementada en ambos lados (cliente y proveedores de servicios).

Un resumen de las características de ambos servicios se muestra en la tabla 4-2:

Tabla 4-2 Características de SOAP vs RestFull

|  |  |  |
| --- | --- | --- |
|  | RESTFULL | SOAP |
| Características | Las operaciones se definen en los mensajes.  Una dirección única para cada instancia del proceso.  Cada objeto soporta las operaciones estándares definidas. | Las operaciones son definidas como puertos WSDL  Dirección única para todas las operaciones.  Múltiples instancias del proceso comparten la misma operación. |
| Ventajas declaradas | Bajo consumo de recursos  Las instancias del proceso son declaradas explícitamente.  El cliente no necesita información de enrutamiento a partir de la URI inicial.  Los clientes pueden tener una interfaz “listener” genérica para las notificaciones.  Generalmente fácil de construir y adoptar | Fácil (generalmente de utilizar).  La depuración es posible.  Las operaciones complejas pueden ser escondidas detrás de una fachada.  Envolver APIs existentes es sencillo.  Incrementa la privacidad.  Herramientas de desarrollo. |
| Posibles desventajas | Gran número de objetos.  Manejar un espacio de nombres (URIs) puede ser engorroso.  La descripción sintáctica/semántica muy informal (orientada al usuario).  Pocas herramientas de desarrollo. | Los clientes necesitan puertos dedicados para diferentes tipos de notificaciones.  Las instancias del proceso son creadas implícitamente. |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

**Fuente:**(Steven Davelaar, 2015)

Oro criterio de comparación es tomado de un artículo científico (AlShahwan, Moessner, & Carrez, 2010, p. 449) y consiste en seleccionar el framework que mejor se desenvuelve en un ambiente móvil y la capacidad de cumplir las metas de los servicios web móviles como se indica en la tabla 5-1.

Tabla 5-1Comparación de SOAP vs RestFull

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Servicios web basados en SOAP** | **Servicios web basados en RestFull** |
| Cliente/ servidor | Perfectamente acoplados | Débilmente acoplados |
| URI | Representa un único servicio en una URI | Representa cada recurso con una URI |
| Capa de transporte | Todos | Únicamente HTTP |
| cache | No soporta | Soporte |
| Interface | Interfaz no uniforme (WSDL) | Interfaz uniforme |
| Contexto de información | Informa al cliente el funcionamiento del servicio web | Funcionamiento de servicios web implícito |
| Tipo de datos | Necesitan conversión de datos | Soporta todos los tipos de datos directamente |
| Método de información | Body entity of HTTP | URI – HTTP |
| Descripción de web services | WSDL | WADL |
| Expandability | No expandible | Expandible sin necesidad de crear nuevos WS |
| Estándares usados | Especifica estándares (WDSLD, UDDI, WS- security) | Estándares web (URL, métodos HTTP, tipos MIME XML) |
| Seguridad/confidencialidad | Especifica estándares WS-security | HTTP secutiry |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

**Fuente:**(AlShahwan et al., 2010)

RestFull Y SOAP tienen enfoques diferentes. RestFull es un estilo de arquitectura para generar aplicaciones cliente-servidor. SOAP es una especificación de protocolo para intercambiar datos entre dos extremos. De la comparación anterior se deduce que RestFull es más ligero y se basa en HTTP, además de no tener estado permitiendo escalabilidad, es compatible con los marcadores y el almacenamiento en memoria cache y mejora el rendimiento además de soportar todo topo de datos.

## 1.4 Servicios web móviles

Los servicios web móviles también llamados por sus siglas MWS se han convertido en una solución ante las necesidad de presentar información, aplicaciones y servicios a diversidad de usuarios, todos propuestos y presentados para dispositivos móviles, tratando de mejorar la experiencia del usuario utilizando herramientas y características del dispositivo, en este tema trataremos sobre las principales características de los servicios web aplicados a un dispositivo móvil

**1.4.1 *Tipos de MWS***

Los servicios web en un dispositivo móvil se presentan en dos formas:

Cualquier servicio web que puede ser consumido por un sistema móvil y cuando un sistema proveedor de servicios Web está albergado en una terminal móvil. Los servicios web móviles deben cumplir con la interoperabilidad e integración en sistemas heterogéneos permitiendo desplegar aplicaciones y servicios a clientes interesados en la utilización de plataformas celulares.

Básicamente los MWS están basados en las tecnologías de XML, HTTP, SOAP y WSDL, se considera importante aprovechar servicios y plataformas nativos y estándares para así extender su productividad, pues el uso de tecnologías para ambientes fijos no es óptimo en ambientes móviles, esto a pesar del avance en hardware cada vez más potente. Las principales dificultades se presentan cuando una tecnología diseñada para un ambiente fijo es aplicada a uno móvil, a pesar de que las herramientas de desarrollo presentan opciones de crear este tipo de aplicaciones aun la tecnología móvil presenta restricciones en el uso de este tipo de aplicaciones como los siguientes (Crespo & Eduardo, 2010ª, p18):

• Redes inalámbricas: este tipo de redes se caracteriza por alta latencia y errores de transmisión. Además, por las características de movilidad de los terminales, las redes inalámbricas poseen topologías altamente cambiantes.

• Procesamiento y memoria limitados: las prestaciones de poder computacional y almacenamiento de dispositivos Smartphone han mejorado notablemente, pero las terminales móviles siempre serán relativamente limitadas comparadas a sus contrapartes fijas.

• Batería: la mejora en las prestaciones de hardware y software en un dispositivo móvil conlleva el mayor uso de energía. Sin embargo, la evolución de la tecnología de baterías no va a la par con la de electrónica móvil; aunque diversas técnicas se pueden aplicar alargar la duración de la batería, estas afectan la experiencia del producto.

• Dispositivos: la gama de celulares es amplia y consiste de aparatos con características altamente integradas.

***1.4.2 Fragmentación de Dispositivos***

La principal dificultad en el desarrollo y despliegue de aplicaciones móviles es la diversidad de dispositivos Smartphone, este es considerado el principal obstáculo para el desarrollo de aplicaciones genéricas para dispositivos móviles. Según Carlos Eduardo Jerves las principales causas de la fragmentación de dispositivos es las necesidades del ambiente externo, estos contextos tienen gran influencia sobre la aplicación. En hardware y software existen variedades de características físicas y de sistemas operativos móviles que presentan interfaces y soportes variados para la implementación de tecnologías.

Puntualizando la variedad de software esta propuesto que se debe elegir un patrón de despliegue de aplicaciones para esto se considera los siguientes enfoques: (Crespo & Eduardo, 2010b, p. 15)

• Manual-Multi: consiste en el desarrollo separado de diferentes aplicaciones según la necesidad. Es la opción más costosa en términos de desarrollo, y aunque facilita el rápido despliegue, dificulta el mantenimiento.

• Derive-Multi: consiste en el desarrollo de aplicaciones que compartan parcialmente bases de código. Esta aliviana el mantenimiento, pero aún se considera que las partes no compartidas son versiones fragmentadas del mismo programa.

• Single-Adapt: consiste en habilitar a una para diferentes ambientes a través de una capa de abstracción. Aunque esta es la mejor solución para mantener una sola aplicación, dificultades podrían surgir con el middleware de abstracción.

## 1.5 Realidad Aumentada

La disponibilidad de ubicarnos en un lugar o ubicar elementos interesantes en el momento preciso, la disposición de aplicaciones en teléfonos móviles y de componentes como la cámara en estos dispositivos, se unen para presentar formas de ubicar objetos superpuestos, con ayuda de marcadores llenos de información para identificar a objetos o lugares. Además la posibilidad de utilizar aplicaciones como los mapas unido con la ubicación de objetos nos ayuda a encontrar aquellos puntos que son de interés.

### 1.5.1 Sistemas de localización de dispositivos móviles

Los dispositivos móviles pueden ser localizados en un medio según la disponibilidad del dispositivo. De aquí se puede diferenciar tres formas de ubicar un dispositivo móvil.

Estos sistemas de localización son de diferentes tipos y se definen como:

**Basados en red:** En este tipo de ubicación es necesario utilizar un sistema de proveedor de servicios, con la desventaja de que ese proveedor siempre debe estar cerca.

**Basados en terminal:** Para ello es necesario de una aplicación, que junto con el receptor de señales del dispositivo, determina la posición del terminal.

**Híbridos**

Son una combinación entre basados en terminal y basados en red, no es más fiable puesto que al igual que los anteriores tiene problemas de adaptación de proveedor y la necesidad de instalar una aplicación.

### 1.5.2 GPS

GPS son las siglas del inglés Global Position System, es un sistema de posicionamiento basado en terminal que permite conocer la situación de un objeto o persona en cualquier lugar del mundo. Se trata de una red de 27 satélites que emiten una señal con el tiempo de emisión y su posición. Esta señal llega al GPS con un cierto retraso, lo cual nos permite calcular de una manera aproximada la distancia del satélite, ya que sabemos que esta señal viaja a la velocidad de la luz.

### 1.5.3 Android Location Services

Es un API para geolocalización creada por Google en el área de desarrollo para Android, ubicada en el paquete *android location*, esta API permite conocer la última ubicación del dispositivo, además de monitorear la posición real o hacer uso de una aplicación para mostrar información de algún lugar u objeto que este cerca o en el rango de una zona marcada. Trabaja con la tecnología GPS, además de utilizar fácilmente los servicios de Google Maps.(Alan Bover Argelaga, 2010, p. 82)

Para utilizar estas posiciones el GPS del teléfono debe estar activad y para que se actualice esta ubicación de debe girar el teléfono, así el acelerómetro de este se actualizara y junto con él se obtendrá la nueva posición del dispositivo para poder referenciar ese punto en un mapa o con realidad aumentada herramientas que componen las funcionalidades dela aplicación.

### 1.5.4 Google Maps

Es un servicio que ofrece mapas para dispositivos móviles de tercera generación, para poder utilizarlos con un dispositivo móvil es necesario encender el GPS para determinar la posición real y poder mostrar una ubicación en el mapa. Del mismo modo que otras aplicaciones web desarrolladas por Google, para implementar Google Maps, se usan un gran número de ficheros JavaScript. Cuando este busca un punto determinado, la ubicación está marcada por un indicador en forma de pin, el cual es una imagen PNG transparente sobre el mapa. Para conseguir la conectividad sin sincronía con el servidor, y así proporcionar al usuario mayor interactividad con el mapa, mediante la realización de peticiones asíncronas a la red con JavaScript y XML HttpRequest, se usa una técnica reciente, conocida como AJAX. Esta herramienta es de gran utilidad para el proyecto por el motivo de que para una aplicación de turismo consideramos importante mostrar en un mapa la ubicación de los lugares de interés.

### 1.5.5 ¿Qué es realidad aumentada?

Es una tecnología que combina elementos reales con virtuales añadiendo información de estos elementos, esta función lo realiza en tiempo real utilizando gráficos en dos y tres dimensiones para hacer la sobrexposición virtual en un mismo espacio.

Para poder hacer uso de esa tecnología necesitamos las siguientes especificaciones:

* Un dispositivo que cuente con los recursos necesarios para realizar el procesamiento de las imágenes que se reconocen en el entorno, principalmente una cámara.
* Software especializado para el reconocimiento, análisis y procesamiento de imágenes en 3D.
* Después de relacionar los elementos virtuales con la realidad se necesita información del porque estos dos objeto son relacionados

Realidad aumentada es un término que define un entorno a través de la virtualización, aplicada a un dispositivo móvil, sus diferentes métodos de presentación se vuelve una herramienta interactiva que ayuda al usuario a buscar y ubicar lugares u objetos de interés en tiempo real.

### 1.5.6 Realidad aumentada en móviles

En los últimos años se ha puesto especial interés en realizar aplicaciones para dispositivos móviles que presenten la facilidad de utilizar la tecnología de realidad aumentada, esto por un lado es posible gracias a la evolución que ha tenido el hardware y software para soportar esta y más tecnologías, sin la restricción que existía antes de los costos elevados.

### 1.5.7 Registro de objetos virtuales

La parte más complicada de la realidad aumentada es saber adquirir información para reconocer la posición del dispositivo y los objetos a su alrededor. La información en mostrada en marcadores, reconocimiento de objetos y posición y orientación del dispositivo.(Alan Bover Argelaga, 2010, p. 106)

### 1.5.8 Realidad aumentada basada en marcadores

Los marcadores son imágenes del entorno que pueden ser reconocidas por el dispositivo, al reconocer esta imagen se ubica al dispositivo y se puede conseguir la información según el tamaño de la figura, ángulo de visión con respecto a la cámara, calculando distancias, con solo una cámara. No es muy conveniente llenar un entorno de marcadores.Para el reconocimiento de estos patrones los cuales pueden ser letras o símbolos, se utiliza la técnica L2-Norm que calcula la relación entre dos objetos con cada uno de los marcadores guardados en memoria, la desventaja que esta comparación se lo realiza en escala de grises y se puede llegar a confundir un marcador con otros.

### 1.5.9 Realidad aumentada basada en reconocimiento de objetos

Se basa en, a través de la cámara web, reconocer un objeto en particular, y compararlo con una base de datos de objetos según su forma para descubrir de qué objeto se trata. Claramente, este sistema no requiere disponer más que una cámara en él dispositivo, y no necesita modificar el entorno para que funcione, lo que la hace totalmente portable de un entorno a otro con toda facilidad.

### 1.5.10 Realidad aumentada basada en posición y orientación del dispositivo

Este tipo de realidad aumentada requiere de un sistema de localización como el GPS y de sistemas que reconozcan la orientación del dispositivo como el acelerómetro. Aquí es necesario utilizar coordenadas para marcar los puntos y de acuerdo a la ubicación del dispositivo se visualizaran los marcadores en la dirección a los puntos de referencia.

### 1.5.11 APIS de realidad aumentada

Es importante reconocer las distintas opciones de aplicaciones de realidad aumentada que se presentan, y escogerlas de acuerdo a las funcionalidades que ofrecen.

**Junaio**

Inicialmente se creó para iPhone, ahora existe una versión disponible para Android, basada en PHP, soporta marcadores y tiene la funcionalidad de reconocer el entorno a través de la posición y orientación del dispositivo.(«Junaio», 2012, www.junaio.com)

**Layar**

Esta aplicación es accesible mediante un servidor y funciona realizando una petición HTTP para guardar puntos de interés a través de la API de código abierto que ofrece, además de ofrecer filtros importantes para búsquedas de punto de interés. («Augmented Reality | Interactive Print | Layar», 2015, p 3)

**Wikitude**

Wikitude es una API de realidad aumentada que funciona para IPhone, Android y algunos móviles con el SO Symbian. Está escrita en Java, y lanza como una llamada a la aplicación, por lo que Wikitude es adaptable a cualquier desarrollo de software en un móvil compatible. Se requiere pedir una KEY registrada para poder acceder al sistema completo de Wikitude, sin ella, se muestra una marca de agua en la pantalla y no se podrá comercializar la aplicación.

Para el funcionamiento de esta API inicialmente se necesitaba tener instalado Wikitude, hoy se puede llamar a la aplicación en código de manera sencilla, sin necesidad de instalar la librería. Los puntos de interés son cargados manualmente en la llamada a la aplicación, lo que nos permite más flexibilidad, pero si trabajamos con bases de datos grandes en red, deberemos solicitar primero los puntos de interés con un protocolo propio y luego cargarlos manualmente. No se recomienda cargar más de 50 puntos de interés en una llamada. Por último, nos permite visualizar modelos 3D.(Wikitude, 2015, www.wikitude.com)

## 1.6 Android

Android es un sistema operativo de código abierto para dispositivos móviles que facilita el trabajo de los desarrolladores, está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma, que proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono. (Como GPS, las llamadas, agenda, acelerómetro, etc.) De una forma sencilla en el lenguaje de programación java. Esta plataforma separa el hardware del software que se ejecuta en él, esto significa que un aplicación Android puede ser ejecutado en un gran número de dispositivos. (Darwin, 2012, p. 89)

### 1.6.1 Características

Pueden ser definidas las siguientes características.(Darwin, 2012, p. 95)

* Open source

Una de las características de este sistema operativo es que es libre, para la programación y para la instalación de aplicaciones no es necesario pagar nada, aumentando su popularidad y bajando los costes de aplicaciones para teléfono móviles. Cualquiera puede tener acceso al código fuente, inspeccionarlos, compilarlo e incluso cambiarlo, este permite adaptar mejor el sistema operativo.

* Soporte para Multimedia y Sensores

Dispone de soporte para medios con formatos comunes de audio, video e imágenes planas, además de cámara, gps, brújula y acelerómetro.

* Utiliza la máquina virtual Dalvik

### 1.6.2 Arquitectura Android

Android se forma de cuatro elementos que se indican en la tabla 6-1:

Tabla 6-1Arquitectura Android

|  |  |
| --- | --- |
| Aplicaciones | |
| Armazón de Aplicaciones | |
| librerías | Android Runtime |
| Kernel de Linux | |

Realizado por: Milton Jiménez

**Fuente:**(Darwin, 2012)

**Runtime de Android:** se encuentran las librerías que ayudan a las funcionalidades del sistema operativo.

**Librerías:** están las librerías que se incluyen en la base de datos expuestas a los desarrolladores a través del framework de las aplicaciones Android System C library

**Armazón de aplicaciones:** expone el código fuente usado en las aplicaciones base.

**Aplicaciones:** están expuestas las aplicaciones que utiliza el cliente como mapas, navegadores, contactos, entre otros.

Android complementa el núcleo que contiene Linux ya que es una modificación de este para compartir código y entre todos mejorarlo y así formar en conjunto algo grande.

Como se ha mencionado los servicios web exponen aplicaciones a ser utilizadas por varios y distintos usuarios a la vez, sin cambiar funcionalidades, mostrando compatibilidad a los distintos tipos de dispositivos a los cuales van dirigidas estas aplicaciones. Las evolución de estos servicios web han cambiado la forma en que los usuarios acceden a la información, así también se amplían los estudios de tecnologías que hacen que esta información llegue en el menor tiempo posible y a mayor cantidad de usuarios, ante esto se crea una comparación entre SOAP como protocolo de acceso a objetos y RESTFULL como arquitectura orientado a recursos, los dos tienen la finalidad de crear y exponer servicios, sin embargo hoy los estos servicios son presentados en una gran variedad de dispositivos, y para ello los dos tratan de presentar mejoras en sus funcionalidades para consumir la menor cantidad de recursos posibles.

SOAP se concentra en encapsular el mensaje que tiene que ser enviado/recibido, llevando un control sobre los datos y sobre todo en la seguridad e integridad de estos. RESTFULL está dirigido a recursos con características y métodos, no le interesa llevar un control de sesión, tampoco sobre qué tipo de datos lleva, su principal objetivo en entregar/recibir información lo más rápido posible, es decir no cuenta con un encabezado que describe las características del mensaje y su situación de recurso ligero hace que la transmisión no sea confiable en cuestión de seguridad pero eso sí, los datos por más extensos que sean podrán llegar en el menor tiempo posible.

Entre las aplicaciones que ofrecen estos servicios están las de dispositivos móviles, dedicados a mejorar la portabilidad de la información, además de ofrecer aplicaciones de ubicación, adicionan la posibilidad de agregar marcadores de interés de cada usuario, combinando la realidad con lo virtual dentro de una zona específica, para ello utiliza frameworks de realidad aumentad, entre algunos se pudo distinguir a Wikitude.Android es una plataforma open source, lo cual facilita el desarrollo de todo tipo de aplicaciones que pueden ser ejecutadas en cualquier dispositivo móvil con ese sistema operativo.

De esto surge un inconveniente; en SOAP todo el encabezado que adjunta a los datos, ocupan recursos, y la ligereza de RESTFULL no asegura la integridad de la información, qué forma se puede adoptar para escoger la mejor opción de servicios web aplicados a un dispositivo móvil; esto se definirá en el siguiente capítulo.

# CAPÍTULO II

# ANÁLISIS COMPARATIVO

En la actualidad el usuario de un entorno de computación móvil será capaz de acceder a datos, información u otros objetos lógicos desde cualquier dispositivo en cualquier red mientras está en movimiento. De aquí la necesidad de buscar la mejor solución en servicios web para aplicaciones móviles que ofrezcan un menor tiempo de respuesta al momento de consumir un servicio web, para ello es importante establecer un mecanismo de comunicación entre la aplicación móvil y el servicio de datos que al instante de proporcionar información esta sea entregada en el menor tiempo posible, buscando la mejor tecnología en ofrecer eficiencia en suministro de datos. Para ello utilizaremos las características que nos presentan SOAP y RESTFULL, dos tipos de arquitecturas orientadas a servicios web.

## 2.1 Definición de parámetros a comprobar

El parámetro que a continuación se define para el análisis entre los servicios web para el desarrollo de aplicaciones móviles está basado en criterios de los tesistas y de un estudio en la implementación de web services(Steven Davelaar, 2015), el cual se va a considerar para este análisis de se indica en la Tabla 1-2.

Tabla 1-2Descripción de Parámetro

|  |  |
| --- | --- |
| PARÁMETRO | DESCRIPCIÓN |
| Rendimiento | El menor tiempo que utiliza un servicio web para suministrar datos a una aplicación móvil. |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

## 2.2 Determinación de las variables de comparación

Para demostrar que servicio web cumple con la función de suministrar información en el menor tiempo posible mediante un dispositivo móvil se analiza la variable de la cual depende el rendimiento en este caso el tipo de servicio que se muestra como variable independiente en la tabla 2-2

Tabla 2-2 Variable Independiente

|  |  |
| --- | --- |
| **Variable Independiente** | **Descripción** |
| Tipo de servicio | El rendimiento depende del tipo de servicio que se implemente, se establece más eficiente el servicio web que provea información en el menor tiempo posible. |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

## 2.3 Ambientes de pruebas

El ambiente de pruebas es un punto importante, ya que las pruebas a realizar deben ejecutarse en un ambiente homogéneo, buscando la mejor opción para aplicar las herramientas de desarrollo utilizadas, evitando perdidas de conexiones y alteraciones en los resultados.

Se muestra en la siguiente figura 1-2 la arquitectura de la aplicación mediante un diagrama de despliegue.

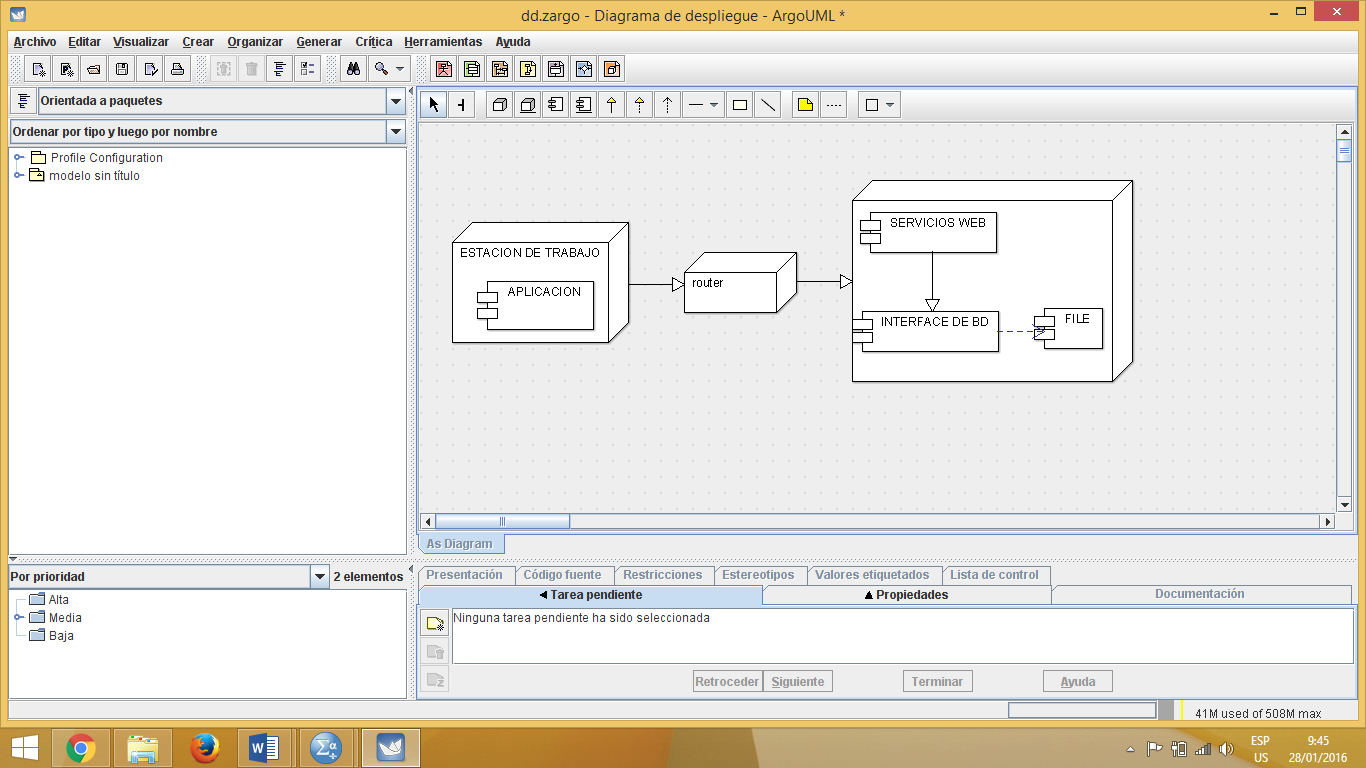


Figura 1-2Ambiente de Pruebas

**Realizado por:** Milton Jiménez y Lorena Hidalgo

El diagrama de despliegue muestra cómo a través de una aplicación móvil con una comunicación mediante un dispositivo de red es posible mostrar información proporcionada por un servicio web.

Para las pruebas se tiene una base de datos de 1000 registros, se elige un dispositivo de gama alta con características como su RAM, el tamaño de pantalla, o la buena recepción de señal al momento de escoger cual tipo de servicio WEB es el recomendable. Este tipo de dispositivo posee todos los elementos más evolucionados de la telefonía móvil, pantalla, acceso a redes de máxima velocidad, conectividad, procesador y la frecuencia en GHz, memoria RAM, batería, servicios gratuitos de almacenamiento, además de la cámara fotográfica y video, para las pruebas se escogió la versión de sistema operativo Lollipop 5.0.(Android, 2015, www.andorid.com)En cuanto a la red se aplica una red wifi para poder conectarse al punto de red más.

Para realizar las pruebas de tiempo de respuesta se crea un método en Android que utiliza  las clases **javax.swing.Timer** esta clase se encarga de mostrar un aviso cada vez que queramos (por ejemplo, un aviso cada vez que todos los servicios sean cargados en el caso del proyecto en desarrollo) y se encarga de llamar a un método que se ha implementado. El resultado de este método es el tiempo en milisegundos en que se demora la aplicación en cargar los servicios web. Basta con instanciarla pasándole cada cuánto tiempo (en milisegundos) queremos que nos avise y un **ActionListener**, cuyo método **actionPerformed()** se ejecutará periódicamente. Luego sólo hay que llamar al método **start()** cuando sea necesario que el **Timer** empiece a contar. El siguiente código demuestra cómo se implementó timer para contar el tiempo de respuesta.

Utilizamos un método público

public void metodo() {

Inicializamos contadores tipo enteros en cero, estos nos servirán para obtener el tiempo que demora cada listado para luego calcular cuánto se demora la aplicación en cargar todos los servicios necesarios.

start=0;end=0;dat1=0;dat2=0;dat3=0;dat4=0;dat5=0;dat6=0;dat7=0;dat8=0;

Asignamos a la variable el tiempo exacto en mili segundos en que comenzamos a correr la aplicación

comienzo= System.currentTimeMillis();

En la variable start obtenemos nuevamente el tiempo exacto en que la aplicación empieza acorrer esto para poder llevar el tiempo de carga de un listado, en este caso el listado de iglesias.

start= System.currentTimeMillis(); CargarServicios.obtenerListadoIglesias();

Obtenemos el tiempo en que el servicio termina de cargarse en la variable end.

end = System.currentTimeMillis();

Luego se calcula la diferencia que existe entre el tiempo de fin y el inicio de la carga del servicio, el resultado es el tiempo en mili segundos de un solo listado.

val1=Calcular(end, start);

De esta manera de obtiene tiempos de respuesta de cada listado que para la aplicación móvil son necesarios, el siguiente listado en cargar y registrar el tiempo de carga es el listado de una galería con imágenes tipo logo.

dat1= System.currentTimeMillis();

CargarServicios.ListadoGaleriaLogos();

dat2 = System.currentTimeMillis();

val2=Calcular(dat2, dat1);

Para la aplicación en total fue necesario cargar 8 listados que tienen como resultado el tiempo que demora cada uno para luego sumar y obtener el resultado final, esto constituye en las pruebas un tiempo de respuesta y para las 383 restantes se realiza el mismo procedimiento tanto para obtener tiempos en RestFull como en SOAP. Las pruebas fueron realizadas simultáneamente y los resultados de una muestra de 384 pruebas de tiempo de respuesta en milisegundo se encuentran en la tabla del anexo A.

## 2.4 Prototipo de los servicios web

Se considera importante realizar un demo del proyecto final, estos prototipo simularán las funcionalidades del proyecto final como realizar administración de galerías, puntos de ubicación, para la organización de la información,

De esta manera se crea un Sistema de Administración y Servicios WEB que consta de:

* Gestión de Iglesias: Con información detallada de la misma el lugar y dirección donde se encuentran.
* Gestión de Rutas: Para una mejor disponibilidad de visitas según su orden.
* Gestión de Galería: Imágenes de Puntos de Interés de cada una de las Iglesias.

La Gestión de las Entidades Iglesias, rutas y galería serán desarrolladas tanto para el servicio web SOAP así como también para Restfull.

### 2.4.1 Prototipo móvil

De la misma forma para la aplicación móvil se considera importante crear una aplicación que represente funcionalidades más cercanas al sistema original.

El sistema consta de:

* Listado de Iglesias: Indica la Información de todas las Iglesias ingresadas.
* Listado de Rutas: Muestra la ruta en un Mapa de google de las Iglesias establecidas.
* Realidad Aumentada: Indica de manera virtual la posición georreferenciada de una Iglesia Especifica.
* Detalle Iglesia: Indica información de la Iglesia, mapas, galería y como llegar al punto.

El sistema Móvil será desarrollado tanto para el consumo de los Servicios Web SOAP como también los Servicios Restfull.

## 2.5 Desarrollo de los prototipos

Para el desarrollo de los prototipos móviles se contó con el uso del IDE Netbeans 8.0, que nos permite crear la aplicación web y los servicios tanto RestFull como SOAP, gracias a ello detallamos a continuación el proceso de desarrollo de cada una de las entidades desarrolladas.Para la creación de una Aplicación web, se implementó el IDE de Netbeans que es un entorno de desarrollo integrado libre, además es un producto libre y gratuito sin restricciones de uso utilizando JAVA como lenguaje de desarrollo para crear aplicaciones con servicios web, se trabaja con el servidor de ***Glassfish***, con el framerwork ***Java Server Faces.***

Se muestra a continuación la forma en que los servicios fueron creados en RestFull y en SOAP, y la forma en que estos servicios son consumidos desde una aplicación móvil.

### 2.5.1 Servicios web RestFull

Para el desarrollo de los Servicios WEB RestFull se utiliza la librería Jersey con la máquina virtual de Java. Las operaciones que se detallan en este tipo de servicio son similares a las de REST, operaciones como: GET, POST, PUT, y DELETE.

Para la creación de los servicios web declaramos un encabezado que consta de

@Path("servicioIglesia")

@Produces({"application/json; charset=utf-8"})

@Consumes({"application/json; charset=utf-8"})

Declaramos una clase con el nombre referente al servicio que voy a realizar la cual tendrá dos atributo que son objeto iglesia y un lisado de iglesias, en el constructor de la clase instanciamos el objeto iglesia y cargamos el lisado de iglesias como se indica a continuación:

public class ServicioIglesia {

private Iglesia iglesia;

private ArrayList<Iglesia> ListaIglesia;

public ServicioIglesia() {

iglesia=new Iglesia();

ListaIglesia= CatalogoIglesia.ListadoIglesia();

}

Para crear un servicio web en RestFull se realiza la invocación a los métodos Post, Put, Get y Delete, con una anotación @ que describe un contenido informativo de un objeto, necesario para la creación del servicio, estos métodos están declarados a continuación:

@POST

public void Insertar( Iglesia entity) {

try {

} catch (Exception ex) {

Logger.getLogger(ServicioIglesia.class.getName()).log(Level.SEVERE, null, ex);

}

}

@PUT

public void Modificar( Iglesia entity) {

try {

} catch (Exception ex) {

Logger.getLogger(ServicioIglesia.class.getName()).log(Level.SEVERE, null, ex);

}

}

@DELETE

@Path("{id}")

public void Eiminar(@PathParam("id") Integer id) {

try {

} catch (Exception ex) {

Logger.getLogger(ServicioIglesia.class.getName()).log(Level.SEVERE, null, ex);

}

}

@GET

@Path("lista")

public ArrayList<Iglesia> Lista() throws Exception{

try {

} catch (Exception ex) {

Logger.getLogger(ServicioIglesia.class.getName()).log(Level.SEVERE, null, ex);

}

}

La operación que vamos a utilizar según la funcionalidad de la aplicación es el método @GET luego definimos el path del método que indica la ruta para llegar al servicio y en el ejemplo propuesto devuelve un listado de tipo Iglesia, dentro de esta operación se invoca a cargariglesia() en donde según la arquitectura de la aplicación se hace uso del catálogo listar iglesia desde la base de datos.

@GET

@Path("lista")

public ArrayList<Iglesia> Lista() throws Exception{

cargariglesia();

return ListaIglesia;

}

public void cargariglesia() throws Exception{

ListaIglesia= CatalogoIglesia.ListadoIglesia();

}

Esta es una manera de desarrollar un servicio web RestFull, para que pueda ser consumida por un cliente.

### 2.5.2 Cliente RestFull

Para el desarrollo de un prototipo móvil se utilizó el IDE de Eclipse, en el cual viene integrado con el sistema operativo Android, que facilita la creación de aplicaciones Móviles desarrolladas en un entorno open source. En RestFull, las llamadas al servicio no se harán a través de una única URL, sino que se determinará la acción a realizar según la URL accedida y la acción HTTP utilizada para realizar la petición (GET, POST, PUT o DELETE).

Iniciamos creando una clase que contenga los métodos de los listados de la entidades dadas, cada método tendrá un objeto HttpClient, que será el encargado de realizar la comunicación HTTP con el servidor a partir de los datos que nosotros le proporcionemos. Tras esto crearemos la petición POST creando un nuevo objeto HttpPost e indicando la URL de llamada al servicio, establecemos el tipo de datos que vamos a utilizar mediante setHeader() en formato json.

public class CargarServicios {

public static void obtenerListadoIglesias() {

DIRECCION\_IP=getDIRECCION\_IP();

HttpClient httpClient = new DefaultHttpClient();

HttpGet del = new HttpGet("http://"+ DIRECCION\_IP+":"+PUERTO+"/PrototipoRESTfull/webServices/servicioIglesia/lista");

del.setHeader("content-type", "application/json");

Declaramos un array de tipo Iglesia para contener el resultados de la consulta, para ello ejecutamos la petición del servicio con execute(). Luego creamos un objeto JSONArray a partir del resultado textual de getEntity() y un objeto JSONObjectpara acceder a los atributos del objeto recorrido utilizando para cada uno de ellos el método get(), según el tipo de cada atributo (getInt(), getString(), etc) y agregamos a la lista de objetos.

ArrayList<Iglesia> lista = null;

try {

HttpResponse resp = httpClient.execute(del);

String respStr = EntityUtils.toString(resp.getEntity());

JSONArray respJSON = new JSONArray(respStr);

lista = new ArrayList<Iglesia>();

for (int i = 0; i < respJSON.length(); i++) {

JSONObject obj = respJSON.getJSONObject(i);

Iglesia miiglesia = new Iglesia();

miiglesia.setId(obj.getInt("id"));

miiglesia.setNombre(obj.getString("nombre"));

miiglesia.setDecripcion(obj.getString("decripcion"));

miiglesia.setDireccion(obj.getString("direccion"));

miiglesia.setLatitud(obj.getDouble("latitud"));

miiglesia.setLongitud(obj.getDouble("longitud"));

lista.add(miiglesia);

}

} catch (Exception ex) {

Log.e("ServicioRest obtenerListadoIglesias", "Error!", ex);

}

Listados.setListaIglesia(lista);

}

Para la construcción de un cliente que consuma servicios web Restfull debe aplicarse las clases que hemos implementado anteriormente con lo que lograremos clientes desde Android capaces de consumir dichos servicios.

### 2.5.3 Servicios web SOAP

En lo que respecta a los Servicios WEB SOAP se utiliza la API de JAX-WS para crear servicios en java, con esta api se desarrolla varias operaciones que fueron desarrolladas en el servicio.

Se crea una clase que tendrá una anotación de @webservice para identificar el nombre del servicio, se declara un objeto y un array del servicio que se va a utilizar. Realizar la instancia del objeto y el array en el constructor de la clase, así mismo se crea un método que cargará el listado de iglesia, como se indica a continuación.

@WebService(serviceName = "ServiciosSoap")

public class ServiciosSoap {

private Iglesia iglesia;

private ArrayList<Iglesia> ListaIglesia;

public ServiciosSoap() throws Exception {

iglesia = new Iglesia();

ListaIglesia = CatalogoIglesia.ListadoIglesia();

}

public void cargariglesia() throws Exception {

ListaIglesia = CatalogoIglesia.ListadoIglesia();

}

La anotación @webMethod indica que el método debe ser publicado como operación del servicio, en necesario para los métodos que quieren ser publicados y si es necesario se puede indicar como parámetro el nombre con el que puede aparecer la operación en el documento WSDL, en el ejemplo siguiente publicamos un listado de tipo iglesias:

@WebMethod(operationName = "listaIglesia")

public ArrayList<Iglesia> listaIglesia() throws Exception {

cargariglesia();

return ListaIglesia;

}

En RestFull declaramos todos los recursos del CRUD, sin embargo en SOAP solo se implementa la invocación de operaciones que sean necesarias, es el caso del ejemplo de servicio propuesto, declaramos la operación y su nombre de ahí únicamente se obtiene un array desde la base de datos listo para ser consumido desde un cliente.

### 2.5.4 Cliente SOAP

Para desarrollar el cliente SOAP en Android es necesario incorporar la librería kSOAP puesto que Android no incluye ningún tipo de soporte para el acceso a los servicios SOAP, entonces el framework permitirá utiliza los servicios web que utilicen el estándar SOAP. Lo primero es definir cuatro constantes que nos servirán para realizar el código:

* NAMESPACE. Espacio de nombres utilizado en nuestro servicio web.
* URL. Dirección URL para realizar la conexión con el servicio web.
* METHOD\_NAME. Nombre del método web concreto que vamos a ejecutar.
* SOAP\_ACTION. Equivalente al anterior, pero en la notación definida por SOAP.

En el ejemplo se muestra la forma en un cliente SOAP es declarado, primero una clase pública y dentro de ella los métodos para invocar los servicios para la explicación utilizaremos el método que contiene un listado de iglesias desde los servicios utilizando las constantes antes mencionadas:

public class CargarServicios {

public static void obtenerListadoIglesias2() {

Listados.setListaIglesia(null);

final String NAMESPACE = "http://Servicios/";

final String URL="http://"+ DIRECCION\_IP+ ":"+PUERTO+"/AplicacionSOAP/ServiciosSoap?WSDL";

final String METHOD\_NAME = "listaIglesia";

final String SOAP\_ACTION = "";

Los siguientes pasos del proceso serán crear la petición SOAP al servicio web, enviarla al servidor y recibir la respuesta.

En primer lugar encontramos los datos de la petición Request que contiene el nombre del método al que queremos llamar y los valores de los parámetros de entrada. Rodeando a esta información se añaden otra serie de etiquetas y datos a modo de contenedor estándar que suele recibir el nombre de Enveloper. La información indicada en este contenedor no es específica de nuestra llamada al servicio, pero sí contiene información sobre formatos y esquemas de validación del estándar SOAP. Por último, durante el envío de esta petición SOAP al servidor mediante el protocolo HTTP se añaden determinados encabezado. Todo esto junto hará que el servidor sea capaz de interpretar correctamente nuestra petición SOAP, se llame al método web correcto, y se devuelva el resultado en un formato dado.

ArrayList<Iglesia> lista = null;

SoapObject request = new SoapObject(NAMESPACE, METHOD\_NAME);

SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);

envelope.dotNet = true;

envelope.setOutputSoapObject(request);

HttpTransportSE transporte = new HttpTransportSE(URL);

Después de obtener el resultado desde el servicio web en el siguiente código recorremos el objeto SOAP con lo cual obtenemos cada una de las propiedades con el método getPropetty la cual le asignamos cada una de los atributos del objeto que estamos solicitando, y asignamos al listado que almacena los objetos.

try

{

transporte.call(SOAP\_ACTION, envelope);

lista = new ArrayList<Iglesia>();

java.util.Vector<SoapObject> rs = (Vector<SoapObject>) envelope.getResponse();

if (rs != null)

{

for (SoapObject cs : rs)

{

Iglesia obj = new Iglesia();

obj.setId(Integer.parseInt(cs.getProperty(2).toString()));

obj.setNombre(cs.getProperty(5).toString());

obj.setDecripcion(cs.getProperty(0).toString());

obj.setDireccion(cs.getProperty(1).toString());

obj.setLatitud(Double.parseDouble(cs.getProperty(3).toString()));

obj.setLongitud(Double.parseDouble(cs.getProperty(4).toString()));

lista.add(obj);

}

}

}

catch (Exception e)

{

e.printStackTrace();

Log.e("ServicioRest ListadoIglesias", "Error!", e);

}

Listados.setListaIglesia(lista);

}

Con este código, es posible consumir un servicio SOAP para ser utilizado dentro de una aplicación móvil como Android.

## 2.6 Planteamiento de hipótesis

La hipótesis puede ser aceptada o rechazada en los dos casos se puede cometer errores, la hipótesis formulada con la intención de rechazarla se llama hipótesis nula y se representa von H0. Rechazar H0 implica aceptar una hipótesis alternativa H1.

El contraste de la hipótesis es bilateral:

H0 ≡ μ1 = μ2

H0= La utilización de los servicios web RESTFULL JERSEY ofrecen igual rendimiento que los Servicios Web SOAP JAX-WS aplicado al sistema de turismo del Consejo Provincial de Chimborazo.

Se establece la hipótesis alternativa

H1 ≡ μ1 ≠ μ2

H1= La utilización de los servicios web RESTFULL JERSEY ofrecen rendimiento diferente que los Servicios Web SOAP JAX-WS aplicado al sistema de turismo del Consejo Provincial de Chimborazo.

## 2.7 Análisis comparativo

El análisis comparativo permite determinar si existe alguna diferencia significativa entre los dos tipos de servicio WEB estadísticamente comprobable en lo que respecta a rendimiento, a la hora de consumir un servicio para aplicaciones móviles. La cantidad de veces que se pueda consultar el servicio web es infinita así se considera una población infinita. La muestra debe permitir el mínimo error posible en los resultados de un procedimiento matemático asignado para una población infinita.

El nivel de confianza es del 95% con un margen de error del 5%.

Nivel de significancia cuando el nivel de confianza es el 95% α=1.96.

Tamaño de la muestra

Donde se indica en la tabla 3-2:

Tabla 3-2 Equivalencias

|  |  |
| --- | --- |
| **Z=** | Nivel de confianza correspondiente con la tabla de valores de Z |
| **n=** | Tamaño de la muestra |
| **p=** | Es la variabilidad positiva |
| **q=** | Es la variabilidad negativa |
| E= | Margen de error del 5% |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

Entonces: =384 muestras

Como es población infinita se pueden realizar infinitas pruebas, esto se considera costoso por eso con un análisis se obtiene una muestra de 384 muestras de tiempo de respuesta.

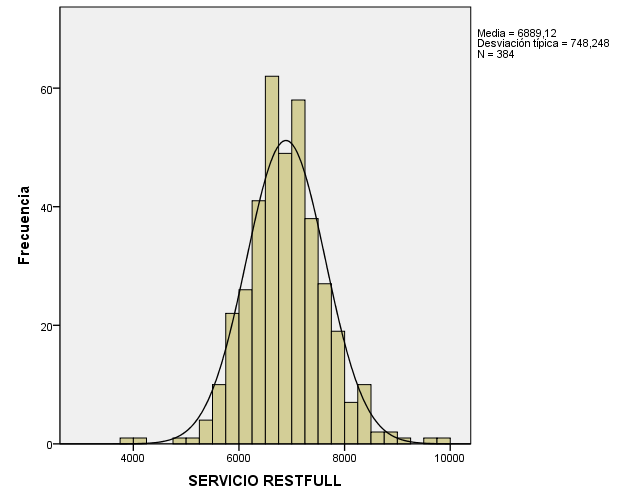
## 2.6 Pruebas y resultados

Se realizaron 384 pruebas de tiempos de respuestas según lo establecido y se obtuvieron resultados que están en el anexo A.

Para determinar si los datos de una población dada provienen de una población normal utilizamos el paquete SPSS que demuestra la normalidad mediante histogramas y un cálculo IQR/S, para ello utilizaremos los datos obtenidos de la muestra de 384 tiempos de respuesta

Histogramas

La demostración de normalidad mediante histogramas en SPSS de los resultados del tiempo de respuesta de los servicios web RestFull se muestra en el gráfico 1-3, y de los servicios web SOAP en el gráfico 2-3.



**Gráfico 1-2** Histograma RestFull

**Realizado por:** Milton Jiménez y Lorena Hidalgo

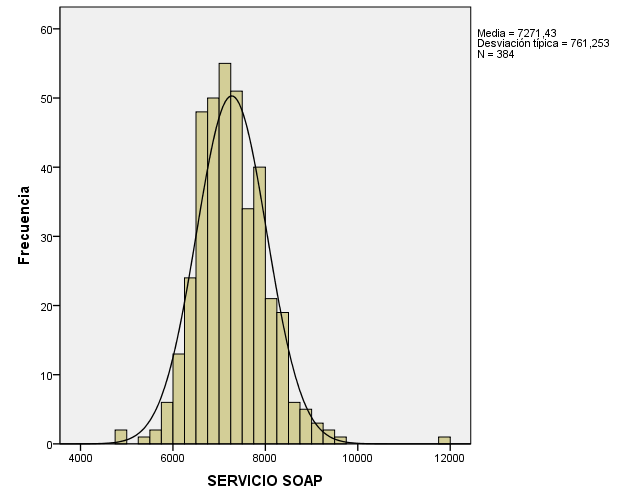


Gráfico 2-2 Histograma SOAP

Realizado por: Milton Jiménez y Lorena Hidalgo

La frecuencia presentada en las dos figuras tienen la forma de una campana de gauss, aunque no se puede afirmar con precisión que la muestra proviene de una población normal podemos concluir que se asemeja a una curva normal teórica.

Calculo IQR/S

El segundo paso para demostrar normalidad es calcular el intervalo intercuartiles (IRQ), la desviación estándar (s) y el cociente entre ellas IQR/S. si los datos son aproximadamente normales IRQ/S ≈ 1.3, esta propiedad se cumple para las distribuciones normales.

Para obtener el intervalo de intercuartiles se obtiene la diferencia entre los percentiles 75º y 25º, estos datos y la desviación estándar de los Servicios web RestFull se obtienen en el paquete SPSS y se muestra en la tabla 4-2.

Tabla 4-2 Valores Estadísticos RESTFULL Obtenidos

| N | Válidos | 384 |
| --- | --- | --- |
| Perdidos | 0 |
| Desv. típ. | | 748,248 |
| Percentiles | 25 | 6442,00 |
| 50 | 6830,50 |
| 75 | 7350,75 |

Realizado por: Milton Jiménez y Lorena Hidalgo

Haciendo el cálculo obtenemos que IRQ/S= 7350.75-6442/748.248= 1.21. Dado que el valor obtenido es aproximadamente igual a 1.3 decimos que los datos obtenidos en RestFull son aproximadamente normales.

De la misma manera calculamos para los estadísticos de los resultados obtenidos en SOAP presentados en la tabla 5-2.

Tabla 5-2Valores Estadísticos SOAP Obtenidos

| N | Válidos | 384 |
| --- | --- | --- |
| Perdidos | 0 |
| Desv. típ. | | 761,253 |
| Percentiles | 25 | 6749,25 |
| 50 | 7213,00 |
| 75 | 7752,25 |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

Haciendo el cálculo obtenemos que IRQ/S= 7752.25-6749.25/761.253= 1.31. Dado que el valor obtenido es muy aproximado a 1.3 decimos que los datos obtenidos en SOAP son aproximadamente normales.

Dado los resultados de las gráficas y el cálculo para probar normalidad podemos concluir que los datos son normalizados y podemos aplicar la prueba T de Sudent para la demostración estadística de la hipótesis nula.

Luego de demostrar a normalización de datos obtenemos la tabla 6-3 donde se muestran los resultados de media, desviación estándar, mínimos y máximos de los datos obtenidos, en RestFull y en SOAP.

Tabla 6-3 Resultados Estadísticos SOAP RESTFULL

|  | | SERVICIO SOAP | SERVICIO RESTFULL |
| --- | --- | --- | --- |
| N | Válidos | 384 | 384 |
| Perdidos | 0 | 0 |
| Media | | 7271,43 | 6889,12 |
| Desv. típ. | | 761,253 | 748,248 |
| Mínimo | | 4967 | 3788 |
| Máximo | | 11820 | 9928 |

**Realizado por:** Lorena Hidalgo y Milton Jiménez

Los datos obtenidos en RestFull tienen una media de 6889.12 y una desviación estándar de 748.25 en comparación con los resultados en SOAP que tiene una media de 7271.43 y desviación estándar de 761.25 lo que de lo que nos muestra una mayor dispersión de los datos en SOAP, en milisegundos RestFull con una media menor que en SOAP lo que indica que los resultados en milisegundos el menor tiempo de respuesta lo tiene RestFull.

Además en el visor de resultados del paquete SPSS obtenemos estadísticos descriptivos del procedimiento prueba T para muestras relacionadas que recoge para cada variable la media, el número de casos, la desviación típica y el error de la media. En la siguiente tabla 13-3 se muestra contiene el intervalo de confianza para la diferencia en las medias. La segunda mitad de la tabla informa sobre el valor del estadístico t y sus grados de libertad (gl) y el nivel crítico bilateral (sig bilateral).

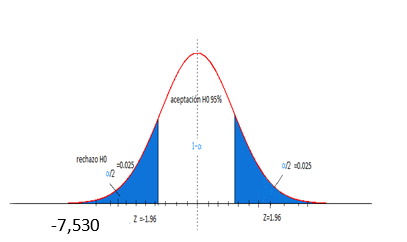
Tabla 7-2 Muestras Relacionadas

|  | | Diferencias relacionadas | | | | t |
| --- | --- | --- | --- | --- | --- | --- |
| 95% Intervalo de confianza para la diferencia | | | |
| Inferior | | Superior | |
| Par 1 | SERVICIO RESTFULL - SERVICIO SOAP | -482,135 | | -282,485 | | -7,530 |
| **Prueba de muestras relacionadas** | | | | |
|  | | Gl | Sig. (bilateral) | |
| Par 1 | SERVICIO RESTFULL - SERVICIO SOAP | 383 | ,000 | |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

El valor de nivel crítico es muy pequeño (0.000), por lo que podemos rechazar la hipótesis de igualdad de medias y con un nivel de confianza del 95% en el gráfico 3.3 que muestra la Prueba bilateral de dos colas y se encuentra que el valor asignado a 0.25 es 1.96 entonces la zona de rechazo de la hipótesis nula se encuentra en la zona menor a -1.96 y mayor a 1.96, como se muestra en la siguiente gráfico 3-2.

Gráfico 3-2 Prueba Bilateral de Dos Colas



**Realizado por:** Milton Jiménez y Lorena Hidalgo

El valor de Z normalizada tiene una probabilidad de -7.53 y se demuestra estadísticamente que existe una diferencia entre el rendimiento de los servicios web RetsFull con Jersey y SOAP con JAX-WS.

Recordamos la hipótesis alterna

H1= La utilización de los servicios web RESTFULL JERSEY ofrecen rendimiento diferente que los Servicios Web SOAP JAX-WS aplicado al sistema de turismo del Consejo Provincial de Chimborazo.

De lo anterior basados en el tiempo de respuesta en milisegundos Restfull con una media de 6889.12 tiene un menor tiempo de repuesta que SOAP con una media de 7271.43, y entre menor sea el tiempo promedio de respuesta a las peticiones del servicio web mayor es la eficiencia y mejor el rendimiento en el mecanismo de comunicación utilizando, se concluye que la utilización de los servicios web RESTFULL JERSEY ofrecen un mayor rendimiento que los Servicios Web SOAP JAX-WS aplicado al sistema de turismo del Consejo Provincial de Chimborazo.

Al invocar servicios web SOAP desde un dispositivo móvil de alta gama, establece un tiempo de espera mayor que el tiempo que presentan los servicios web RestFull, es probable que los servicios web SOAP sean descartados para la implementación en aplicaciones móviles, los resultados están definidos por la arquitectura que presentan SOAP, el formato XML es conciso, lleva información detallada sobre el mensaje, además de que WSDL utiliza este tipo de formato, lleva el registro de tipos de datos, mensajes, operaciones, port type, etc. lo que se supone consume demasiados recursos. Al contrario de RestFull que utiliza dos tipos de formatos xml y json y tiene más sencillez en su implementación.

Una elección dependerá del tipo de Aplicación, la arquitectura implementada, la disponibilidad de documentación, la integración con otros estándares y la compatibilidad con los navegadores.

A partir de ello se puede decir que Restfull es de mayor rendimiento en cuanto a SOAP, por lo que se procederá al desarrollo del sistema de Turismo de Chimborazo implementado los servicios WEB con Restfull.

**CAPITULO III**

**DESARROLLO DE LA APLICACIÓN PARA TURISMO EN LA PROVINCIA DE CHIMBORAZO**

La nueva visión en el Ecuador es incentivar el turismo de sus cuatro mundos (Galápagos, Costa, Andes y Amazonía), dentro de las aplicaciones móviles no se ha desarrollado una que se oriente a brindar información al turista sobre los lugares de interés, combinando servicios de ubicación y mapas, mejorando la experiencia de los turistas, alcanzando un beneficio común tanto de los visitantes como del lugar o persona que presta el servicio.

Chimborazo cuenta con lugares exóticos, con una historia y una cultura rica que tradicionalmente se ha presentado al turista mediante agencias de viaje o un programa web a nivel nacional que informa al usuario sobre lugares propicios para el turismo. La información será administrada mediante un sitio web y la información será presentada de forma detallada, actualizada sobre lugres turísticos de interés, mediante una aplicación móvil, que presente datos específicos que ayude a un sector definido como lo es la provincia de Chimborazo, así las principales funcionalidades de la aplicación contará con las siguiente características:

* Separar información por cantones.
* Mostrar rutas por donde el turista puede llegar a su destino.
* Llamar la atención del usuario mediante una galería de fotos del lugar y los servicios que ofrece además de utilizar otros medios multimedia interesantes como la realidad del entorno mediante gráficos virtuales.

La mejor estrategia para motivar al turista a visitar la provincia de Chimborazo es presentar información específica, siendo agradable a la vista del usuario y fácil de utilizar.

## 3.1 Metodología de desarrollo

La metodología de desarrollo en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Para el desarrollo de la aplicación se utiliza Scrum por su descripción ágil y comprensible para gestionar el desarrollo de software, permitiendo fomentar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

## 3.1.1 Estudio Preliminar

Para determinar si el proyecto es factible o no se realiza el estudio preliminar, en el cual se hizo un análisis del ámbito del sistema, donde analizamos el alcance del proyecto, la gestión de riesgos en la cual se identifica los principales riesgos que podrían retrasar el proyecto además se muestra las posibles soluciones en el caso que estos se presenten, el estudio comercial donde se utiliza la herramienta Cocomo implementando la técnica de puntos de función con la se determinó la cantidad de 65000 líneas de código, implementadas en un tiempo de 26.5 semanas con un esfuerzo de 2 personas y con lo cual se obtuvo un costo aproximado de 6647.62 dólares. También existen los recursos de hardware y software necesarios para el desarrollo del proyecto. Este estudio preliminar se encuentra detallado en el manual de usuario.

Gracias a estudio realizado se concluye que si es factible, por lo que recomendamos la ejecución del proyecto.

## 3.1.2 Planificación

Con el fin de determinar objetivos se realiza una planificación del desarrollo del proyecto, para esto se realiza 3 reuniones con el cliente, donde se definen los roles, los requerimientos funcionales y no funcionales dando como resultado el siguiente listado:

**Requerimientos Funcionales Básicos**

* La autenticación de Usuario para la Administración.
* La Administración de los Cantones.
* La Administración de Precios para Iglesias y Museos de un Cantón.
* La Administración de Servicios para Iglesias.
* La Administración de Cocinas para Restaurantes.
* La Administración de Festividades.
* La Administración de Contactos.
* La Administración de Iglesias.
* La Administración de Museos.
* La Administración de Lugares Turísticos.
* La Administración de Hoteles.
* La Administración de Restaurantes.

**Requerimientos no funcionales**

* El cliente considera importante a largo plazo incrementar las funcionalidades del sistema y adaptarse a procesos cambiantes por lo cual el sistema debe ser escalable.
* Para la implementación el sistema debe estar disponible 7 días por 24 horas, garantizando un esquema adecuado que permita ante una posible falla de la solución en cualquiera de sus componentes para que de esa manera no exista perdida de información.
* El usuario necesita que las tareas sean cumplidas en el menor tiempo posible y se espera un mejor rendimiento.
* La aplicación debe ser desarrollada en software libre.

Gracias a las reuniones realizadas se determina 12 requerimientos funcionales básicos. Para lo cual se recomienda realizar más reuniones y tomar en cuenta la importancia de definir todos los requerimientos que luego serán detallados en el product backlog.

**3.1.3 Product Backlog**

Para definir los requisitos del sistema se crea el Product Backlog gestionada por el cliente con la ayuda del Scrum Master. Este listado consta de un identificador, nombre de la historia, prioridad asignada por el cliente, la estimación que para poder calcular se aplica la técnica de Planing Pocker que involucra a todos los miembros del equipo obteniendo la estimación de cada requerimiento, y el responsable para la elaboración como se indica en la tabla 1-3.

Tabla 1-3 Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ Historia/tarea** | **Nombre Historia/ Nombre Tarea** | **Prioridad** | **Puntos** | **Responsable** |
| **(horas)** |
| HT1 | Como desarrollador quiero realizar un análisis y diseño de la base de datos para almacenar información | Alta | 48 | Milton Jiménez |
| HT2 | Como desarrollador quiero crear un sitio web para la administración de datos | Alta | 14 | Milton Jiménez |
| HT3 | Como desarrollador quiero definir un estándar de codificación, para utilizarlo como modelo o patrón de referencia | Alta | 4 | Milton Jiménez |
| HT4 | Como desarrollador quiero definir un diseño de interfaces para establecer una referencia en diseño | Alta | 8 | Milton Jiménez |
| HU1 | Como usuario quiero insertar un cantón para la administración del Sistema | Alta | 12 | Lorena Hidalgo |
| HU2 | Como usuario quiero modificar un cantón para la administración del Sistema | Alta | 10 | Milton Jiménez |
| HU3 | Como usuario quiero eliminar un cantón para la administración del Sistema | Alta | 10 | Lorena Hidalgo |
| HU4 | Como usuario quiero un listado de cantones para la administración del Sistema | Alta | 10 | Milton Jiménez |
| HU5 | Como usuario quiero insertar una categoría de precio para la administración de Iglesias y Museos | Alta | 10 | Lorena Hidalgo |
| HU6 | Como usuario quiero modificar una categoría de precio para la administración de Iglesias y Museos | Alta | 10 | Milton Jiménez |
| HU7 | Como usuario quiero eliminar una categoría de precio para la administración de Iglesias y Museos | Alta | 10 | Lorena Hidalgo |
| HU8 | Como usuario quiero listar las categorías de precio para la administración de Iglesias y Museos | Alta | 10 | Milton Jiménez |
| HU9 | Como usuario quiero insertar un Servicio para la administración de Iglesias | Alta | 10 | Lorena Hidalgo |
| HU10 | Como usuario quiero modificar un Servicio para la administración de Iglesias | Alta | 10 | Milton Jiménez |
| HU11 | Como usuario quiero eliminar un Servicio para la administración de Iglesias | Alta | 10 | Lorena Hidalgo |
| HU12 | Como usuario quiero listar los Servicios para la administración de Iglesias | Alta | 10 | Milton Jiménez |
| HU13 | Como usuario quiero insertar una categoría cocina para la administración de Restaurantes | Alta | 10 | Lorena Hidalgo |
| HU14 | Como usuario quiero modificar una categoría cocina para la administración de Restaurantes | Alta | 10 | Milton Jiménez |
| HU15 | Como usuario quiero eliminar una categoría cocina para la administración de Restaurantes | Alta | 10 | Lorena Hidalgo |
| HU16 | Como usuario quiero listar las categorías de cocina para la administración de Restaurantes | Alta | 10 | Milton Jiménez |
| HU17 | Como usuario quiero insertar una festividad para la administración de Iglesias | Alta | 10 | Lorena Hidalgo |
| HU18 | Como usuario quiero modificar una festividad para la administración de Iglesias | Alta | 10 | Milton Jiménez |
| HU19 | Como usuario quiero eliminar una festividad para la administración de Iglesias | Alta | 10 | Lorena Hidalgo |
| HU20 | Como usuario quiero listar una festividad para la administración de Iglesias | Alta | 10 | Milton Jiménez |
| HU21 | Como usuario quiero insertar un contacto para la administración una Entidad | Alta | 10 | Lorena Hidalgo |
| HU22 | Como usuario quiero modificar un contacto para la administración una Entidad | Alta | 10 | Milton Jiménez |
| HU23 | Como usuario quiero eliminar un contacto para la administración una Entidad | Alta | 10 | Lorena Hidalgo |
| HU24 | Como usuario quiero listar un contacto para la administración una Entidad | Alta | 10 | Milton Jiménez |
| HU25 | Como usuario quiero insertar una iglesia para la administración del Sistema | Media | 12 | Milton Jiménez |
| HU26 | Como usuario quiero eliminar una iglesia para la administración del Sistema | Media | 12 | Lorena Hidalgo |
| HU27 | Como usuario quiero un listado de iglesias para la administración del Sistema | Media | 18 | Milton Jiménez |
| HU28 | Como usuario quiero insertar un museo para la administración del Sistema | Media | 12 | Lorena Hidalgo |
| HU29 | Como usuario quiero eliminar un museo para la administración del Sistema | Media | 12 | Milton Jiménez |
| HU30 | Como usuario quiero un listado de museos para administración del Sistema | Media | 18 | Lorena Hidalgo |
| HU31 | Como usuario quiero insertar un lugar turístico para la administración del Sistema | Media | 12 | Milton Jiménez |
| HU32 | Como usuario quiero eliminar un lugar turístico para la administración del Sistema | Media | 12 | Lorena Hidalgo |
| HU33 | Como usuario quiero listar los lugares turístico para la administración del Sistema | Media | 18 | Milton Jiménez |
| HU34 | Como usuario quiero insertar un hotel para la administración del Sistema | Media | 12 | Lorena Hidalgo |
| HU35 | Como usuario quiero eliminar un hotel para la administración del Sistema | Media | 12 | Milton Jiménez |
| HU36 | Como usuario quiero listar los hoteles para la administración del Sistema | Media | 18 | Lorena Hidalgo |
| HU37 | Como usuario quiero insertar un Restaurante para la administración del Sistema | Media | 12 | Milton Jiménez |
| HU38 | Como usuario quiero eliminar un Restaurante para la administración del Sistema | Media | 12 | Lorena Hidalgo |
| HU39 | Como usuario quiero listar los Restaurantes para la administración del Sistema | Media | 18 | Milton Jiménez |
| HU40 | Como usuario quiero insertar un precio para la administración de las Iglesias | Baja | 10 | Lorena Hidalgo |
| HU41 | Como usuario quiero eliminar un precio para la administración de las Iglesias | Baja | 10 | Lorena Hidalgo |
| HU42 | Como usuario quiero listar los precios para la administración de las Iglesias | Baja | 10 | Lorena Hidalgo |
| HU43 | Como usuario quiero insertar un precio para la administración de los museos | Baja | 10 | Lorena Hidalgo |
| HU44 | Como usuario quiero eliminar un precio para la administración de los museos | Baja | 10 | Lorena Hidalgo |
| HU45 | Como usuario quiero listar precios para la administración de los museos | Baja | 10 | Lorena Hidalgo |
| HU46 | Como usuario quiero insertar la galería para administración del Sistema | Baja | 20 | Lorena Hidalgo |
| HU47 | Como usuario quiero eliminar la galería para administración del Sistema | Baja | 6 | Lorena Hidalgo |
| HU48 | Como usuario quiero listar la galería para administración del Sistema | Baja | 28 | Lorena Hidalgo |
| HU49 | Como usuario quiero autenticarme para acceso al Sistema | Baja | 10 | Lorena Hidalgo |
| HT5 | Como desarrollador quiero documentar el sistema para utilizarlo como fuente informativa | Baja | 34 | Lorena Hidalgo |
| HT6 | Como usuario quiero capacitación para la utilización del sistema | Baja | 4 | Lorena Hidalgo |
| HT7 | Como desarrollador quiero crear una aplicación móvil para mostrar información | Baja | 6 | Milton Jiménez |
| HT8 | Como desarrollador quiero definir un diseño de interfaces para establecer una referencia en diseño | Baja | 10 | Milton Jiménez |
| HU50 | Como Usuario quiero visualizar la información detallada de un Cantón | Baja | 23 | Milton Jiménez |
| HU51 | Como Usuario quiero visualizar un listado de Iglesias según el Cantón para Información Turística | Baja | 27 | Milton Jiménez |
| HU52 | Como Usuario quiero visualizar un listado de Museos según el Cantón para Información Turística | Baja | 25 | Milton Jiménez |
| HU53 | Como Usuario quiero visualizar un listado de Lugares según el Cantón para Información Turística | Baja | 28 | Milton Jiménez |
| HU54 | Como Usuario quiero visualizar un listado de Hoteles según el Cantón para Información Turística | Baja | 28 | Milton Jiménez |
| HU55 | Como Usuario quiero visualizar un listado de Restaurantes según el Cantón para Información Turística | Baja | 28 | Milton Jiménez |
| HU56 | Como Usuario quiero visualizar un mapa de puntos según la categoría escogida para Información Turística | Baja | 30 | Lorena Hidalgo |
| HU57 | Como Usuario quiero visualizar puntos de realidad aumentada según la categoría escogida para Información Turística | Baja | 48 | Lorena Hidalgo |
| HU58 | Como Usuario quiero visualizar la descripción de una Iglesia escogida para Información Turística | Baja | 19 | Milton Jiménez |
| HU59 | Como Usuario quiero visualizar la descripción de un Museo escogido para Información Turística | Baja | 19 | Milton Jiménez |
| HU60 | Como Usuario quiero visualizar la descripción de un Lugar escogido para Información Turística | Baja | 19 | Milton Jiménez |
| HU61 | Como Usuario quiero visualizar la descripción de un Hotel escogido para Información Turística | Baja | 18 | Milton Jiménez |
| HU62 | Como Usuario quiero visualizar la descripción de un Restaurante escogido para Información Turística | Baja | 19 | Milton Jiménez |
| HU63 | Como Usuario quiero visualizar la Galería de un Punto escogido para Información Turística | Baja | 58 | Lorena Hidalgo |
| HU64 | Como Usuario quiero visualizar el detalle un contacto para Información Turística | Baja | 18 | Milton Jiménez |
| HT9 | Como desarrollador quiero documentar el sistema móvil para utilizarlo como fuente informativa | Baja | 16 | Lorena Hidalgo |
| HT10 | Como usuario quiero capacitación para la utilización del sistema móvil | Baja | 3 | Lorena Hidalgo |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

Para la creación del Product Backlog se utilizó la técnica de las tarjetas de historias de usuario que consta de los siguientes campos:

Identificador (las historias de usuario como HU y las historias técnicas como HT), título, descripción, estimación, prioridad de implementación respecto a las otras historias de usuario, la prioridad alta será la primera en ser implementada, riesgo que corre una historia de tener algún tipo de retraso, sprint asignado, programador responsable y las pruebas de aceptación irán comprobando la funcionalidad del sistema como se indica en la figura 5-4

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: HT1** | **Usuario:** Desarrollador | |
| **Nombre historia:** Análisis y diseño de la base de Datos | | |
| **Prioridad en negocio:**  Alta | | **Riesgo en desarrollo:**  Baja |
| **Puntos estimados: 48** | | **Sprint asignado: 1** |
| **Programador responsable:** Lorena Hidalgo | | |
| **Descripción:** | | |
| Como desarrollador quiero un análisis y diseño de la base de datos para almacenar información para ello se necesita instalar herramientas, realizar un diagrama entidad relación, un diseño lógico y físico, realizar un diccionario de datos, y realizar procedimientos almacenados. | | |
| **Pruebas de aceptación** | | |
| Ingresar datos mediante la utilización de procedimientos almacenados Verificar el tipo de datos que serán almacenados | | |

Figura 1-3 Tarjeta de Usuario

**Realizado por**: Milton Jiménez y Lorena Hidalgo

Para el desarrollo del sistema de turismo se estima un tiempo de 1130 horas asignadas a dos desarrolladores encargados del avance del proyecto con 64 historias de usuario y 10 historias técnicas, de las cuales la mayoría son de prioridad alta, el siguiente paso define el sprintBacklog según la prioridad que tenga cada historia. Existen problemas en asignar la escala de prioridades con variables cuantitativas.

### 3.1.4 Sprint Backlog

Es la planificación que se va a hacer en el sprint, se asigna las tareas a cada persona y el tiempo que queda para terminarlas así el proyecto se divide en partes más pequeñas ordenadas por el cliente y permite tener una referencia diaria del tiempo que le queda a cada tarea como se indica en la tabla 2-3.

Tabla 7-4 Sprint Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint** | **№ Historia** | **Nombre Historia/ Nombre Tarea** | **fecha inicio** | **fecha fin** |
| **1** | HT1 | Como desarrollador quiero realizar un análisis y diseño de la base de datos para almacenar información | 15-09-2014 | 22-09-2014 |
| **1** | HT2 | Como desarrollador quiero crear un sitio web para la administración de datos | 22-09-2014 | 23-09-2014 |
| **1** | HT3 | Como desarrollador quiero definir un estándar de codificación, para utilizarlo como modelo o patrón de referencia | 23-09-2014 | 23-09-2014 |
| **1** | HT4 | Como desarrollador quiero definir un diseño de interfaces para establecer una referencia en diseño | 24-09-2014 | 24-09-2014 |
| **1** | HU1 | Como usuario quiero insertar un cantón para la administración del Sistema | 15-09-2014 | 16-09-2014 |
| **1** | HU2 | Como usuario quiero modificar un cantón para la administración del Sistema | 24-09-2014 | 26-09-2014 |
| **1** | HU3 | Como usuario quiero eliminar un cantón para la administración del Sistema | 16-09-2014 | 18-09-2014 |
| **1** | HU4 | Como usuario quiero un listado de cantones para la administración del Sistema | 26-09-2014 | 29-09-2014 |
| **1** | HU5 | Como usuario quiero insertar una categoría de precio para la administración de Iglesias y Museos | 18-09-2014 | 19-09-2014 |
| **1** | HU6 | Como usuario quiero modificar una categoría de precio para la administración de Iglesias y Museos | 29-09-2014 | 30-09-2014 |
| **1** | HU7 | Como usuario quiero eliminar una categoría de precio para la administración de Iglesias y Museos | 19-09-2014 | 22-09-2014 |
| **1** | HU8 | Como usuario quiero listar las categorías de precio para la administración de Iglesias y Museos | 30-09-2014 | 01-10-2014 |
| **1** | HU9 | Como usuario quiero insertar un Servicio para la administración de Iglesias | 22-09-2014 | 23-09-2014 |
| **1** | HU10 | Como usuario quiero modificar un Servicio para la administración de Iglesias | 01-10-2014 | 02-10-2014 |
| **1** | HU11 | Como usuario quiero eliminar un Servicio para la administración de Iglesias | 23-09-2014 | 24-09-2014 |
| **1** | HU12 | Como usuario quiero listar los Servicios para la administración de Iglesias | 2-10-2014 | 3-10-2014 |
| **1** | HU13 | Como usuario quiero insertar una categoría cocina para la administración de Restaurantes | 24-09-2014 | 25-09-2014 |
| **1** | HU14 | Como usuario quiero modificar una categoría cocina para la administración de Restaurantes | 3-10-2014 | 5-10-2014 |
| **2** | HU15 | Como usuario quiero eliminar una categoría cocina para la administración de Restaurantes | 25-09-2014 | 26-09-2014 |
| **2** | HU16 | Como usuario quiero listar las categorías de cocina para la administración de Restaurantes | 5-10-2014 | 6-10-2014 |
| **2** | HU17 | Como usuario quiero insertar una festividad para la administración de Iglesias | 26-10-2014 | 29-10-2014 |
| **2** | HU18 | Como usuario quiero modificar una festividad para la administración de Iglesias | 6-10-2014 | 7-10-2014 |
| **2** | HU19 | Como usuario quiero eliminar una festividad para la administración de Iglesias | 29-09-2014 | 30-09-2014 |
| **2** | HU20 | Como usuario quiero listar una festividad para la administración de Iglesias | 7-10-2014 | 8-10-2014 |
| **2** | HU21 | Como usuario quiero insertar un contacto para la administración una Entidad | 30-09-2014 | 1-10-2014 |
| **2** | HU22 | Como usuario quiero modificar un contacto para la administración una Entidad | 8-10-2014 | 9-10-2014 |
| **2** | HU23 | Como usuario quiero eliminar un contacto para la administración una Entidad | 1-10-2014 | 2-10-2014 |
| **2** | HU24 | Como usuario quiero listar un contacto para la administración una Entidad | 9-10-2014 | 10-10-2014 |
| **2** | HU25 | Como usuario quiero insertar una iglesia para la administración del Sistema | 13-10-2014 | 13-10-2014 |
| **2** | HU26 | Como usuario quiero eliminar una iglesia para la administración del Sistema | 3-10-2014 | 6-10-2014 |
| **2** | HU27 | Como usuario quiero un listado de iglesias para la administración del Sistema | 13-10-2014 | 15-10-2014 |
| **2** | HU28 | Como usuario quiero insertar un museo para la administración del Sistema | 6-10-2014 | 7-10-2014 |
| **2** | HU29 | Como usuario quiero eliminar un museo para la administración del Sistema | 15-10-2014 | 17-10-2014 |
| **2** | HU30 | Como usuario quiero un listado de museos para administración del Sistema | 7-10-2014 | 9-10-2014 |
| **2** | HU31 | Como usuario quiero insertar un lugar turístico para la administración del Sistema | 17-10-2014 | 20-10-2014 |
| **2** | HU32 | Como usuario quiero eliminar un lugar turístico para la administración del Sistema | 9-10-2014 | 13-10-2014 |
| **2** | HU33 | Como usuario quiero listar los lugares turístico para la administración del Sistema | 20-10-2014 | 22-10-2014 |
| **2** | HU34 | Como usuario quiero insertar un hotel para la administración del Sistema | 13-10-2014 | 14-10-2014 |
| **3** | HU35 | Como usuario quiero eliminar un hotel para la administración del Sistema | 22-10-2014 | 23-10-2014 |
| **3** | HU36 | Como usuario quiero listar los hoteles para la administración del Sistema | 14-10-2014 | 16-10-2014 |
| **3** | HU37 | Como usuario quiero insertar un Restaurante para la administración del Sistema | 23-10-2014 | 27-10-2014 |
| **3** | HU38 | Como usuario quiero eliminar un Restaurante para la administración del Sistema | 16-10-2014 | 17-10-2014 |
| **3** | HU39 | Como usuario quiero listar los Restaurantes para la administración del Sistema | 27-10-2014 | 29-10-2014 |
| **3** | HU40 | Como usuario quiero insertar un precio para la administración de las Iglesias | 17-10-2014 | 20-10-2014 |
| **3** | HU41 | Como usuario quiero eliminar un precio para la administración de las Iglesias | 20-10-2014 | 21-10-2014 |
| **3** | HU42 | Como usuario quiero listar los precios para la administración de las Iglesias | 21-10-2014 | 22-10-2014 |
| **3** | HU43 | Como usuario quiero insertar un precio para la administración de los museos | 22-10-2014 | 23-10-2014 |
| **3** | HU44 | Como usuario quiero eliminar un precio para la administración de los museos | 23-10-2014 | 24-10-2014 |
| **3** | HU45 | Como usuario quiero listar precios para la administración de los museos | 24-10-2014 | 27-10-2014 |
| **3** | HU46 | Como usuario quiero insertar la galería para administración del Sistema | 27-10-2014 | 30-10-2014 |
| **3** | HU47 | Como usuario quiero eliminar la galería para administración del Sistema | 30-10-2014 | 30-10-2014 |
| **3** | HU48 | Como usuario quiero listar la galería para administración del Sistema | 31-10-2014 | 5-11-2014 |
| **3** | HU49 | Como usuario quiero autenticarme para acceso al Sistema | 5-11-2014 | 6-11-2014 |
| **3** | HT5 | Como desarrollador quiero documentar el sistema para utilizarlo como fuente informativa | 6-11-2014 | 11-11-2014 |
| **3** | HT6 | Como usuario quiero capacitación para la utilización del sistema | 11-11-2014 | 12-11-2014 |
| **4** | HT7 | Como desarrollador quiero crear una aplicación móvil para mostrar información | 29-10-2014 | 30-10-2014 |
| **4** | HT8 | Como desarrollador quiero definir un diseño de interfaces para establecer una referencia en diseño | 30-10-2014 | 31-10-2014 |
| **4** | HU50 | Como Usuario quiero visualizar la información detallada de un Cantón | 31-10-2014 | 5-11-2014 |
| **4** | HU51 | Como Usuario quiero visualizar un listado de Iglesias según el Cantón para Información Turística | 5-11-2014 | 10-11-2014 |
| **4** | HU52 | Como Usuario quiero visualizar un listado de Museos según el Cantón para Información Turística | 11-10-2014 | 13-11-2014 |
| **4** | HU53 | Como Usuario quiero visualizar un listado de Lugares según el Cantón para Información Turística | 13-11-2014 | 17-11-2014 |
| **4** | HU54 | Como Usuario quiero visualizar un listado de Hoteles según el Cantón para Información Turística | 17-11-2014 | 19-11-2014 |
| **4** | HU55 | Como Usuario quiero visualizar un listado de Restaurantes según el Cantón para Información Turística | 19-11-2014 | 21-11-2014 |
| **4** | HU56 | Como Usuario quiero visualizar un mapa de puntos según la categoría escogida para Información Turística | 12-11-2014 | 19-11-2014 |
| **5** | HU57 | Como Usuario quiero visualizar puntos de realidad aumentada según la categoría escogida para Información Turística | 19-11-2014 | 25-11-2014 |
| **5** | HU58 | Como Usuario quiero visualizar la descripción de una Iglesia escogida para Información Turística | 21-11-2014 | 25-11-2014 |
| **5** | HU59 | Como Usuario quiero visualizar la descripción de un Museo escogido para Información Turística | 25-11-2014 | 27-11-2014 |
| **5** | HU60 | Como Usuario quiero visualizar la descripción de un Lugar escogido para Información Turística | 27-11-2014 | 1-12-2014 |
| **5** | HU61 | Como Usuario quiero visualizar la descripción de un Hotel escogido para Información Turística | 2-12-2014 | 4-12-2014 |
| **5** | HU62 | Como Usuario quiero visualizar la descripción de un Restaurante escogido para Información Turística | 4-12-2014 | 8-12-2014 |
| **5** | HU63 | Como Usuario quiero visualizar la Galería de un Punto escogido para Información Turística | 25-11-2014 | 2-12-2014 |
| **5** | HU64 | Como Usuario quiero visualizar el detalle un contacto para Información Turística | 8-12-2014 | 10-12-2014 |
| **5** | HT9 | Como desarrollador quiero documentar el sistema móvil para utilizarlo como fuente informativa | 2-12-2014 | 5-12-2014 |
| **5** | HT10 | Como usuario quiero capacitación para la utilización del sistema móvil | 9-12-2014 | 12-12-2014 |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

Se identifican 5 Sprints para la creación satisfactoria del proyecto, cada sprint tiene un tiempo estimado aproximado de treinta días laborables y para detallar Sprint Backlog se utiliza la tarjeta de tareas que contiene el nombre de la tarea, el avance previsto y el real, las observaciones la solución y el estado en que se encuentra la tarea, como se muestra en la tabla 3-3.

Tabla 3-3 Tarjeta de Tareas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tarea** | **Avance previsto** | **Avance real** | **Observaciones** | **Solución** | **Estado tarea** |
| Diagrama entidad relación | 4 | 4 | Diseño diagrama entidad relación | Correcto | Terminada |
| Diseño lógico y Físico | 4 | 6 | Diseño lógico y físico de la BD | Correcto | Terminada |
| Creación de tablas | 16 | 16 | Campo en Base de Datos | se agrega campo tipo de imagen | En proceso |
| Creación de procedimientos | 16 | 16 | Desarrollo de procedimientos almacenados | Correcto | Terminada |
| Diccionario de datos | 8 | 5 | Definir tipo de datos | Tipo de dato de imagen | En proceso |
| **Total avance** | **48** | **47** |  |  |  |

**Realizado por:** Milton Jiménez y Lorena Hidalgo

El registro de tareas muestra el avance real del proyecto y el tiempo real en que las tareas del Sprint son desarrolladas, si el tiempo es suficiente se agregarán tareas del siguiente sprint caso contrario si existe tiempo faltante las tareas se agregarán al siguiente.

### 3.1.5 Sprint 1

En la reunión inicial se definen 4 historias técnicas y 14 historias de usuario.Las reuniones diarias por cuestión de tiempo se realizan una vez por semana, en esta reunión el equipo comparte toda la información del proyecto, generando y adaptado cambios que se dan dentro del desarrollo y así aumentar la productividad del entregable.

HT1: Como desarrollador quiero realizar un análisis y diseño de la base de datos para almacenar información

Tiene un tiempo estimado de 48 horas y en este tiempo se desarrollarán 5 tareas, para iniciar se crea el diagrama entidad relación y esta se indica en la gráfico 1-3.

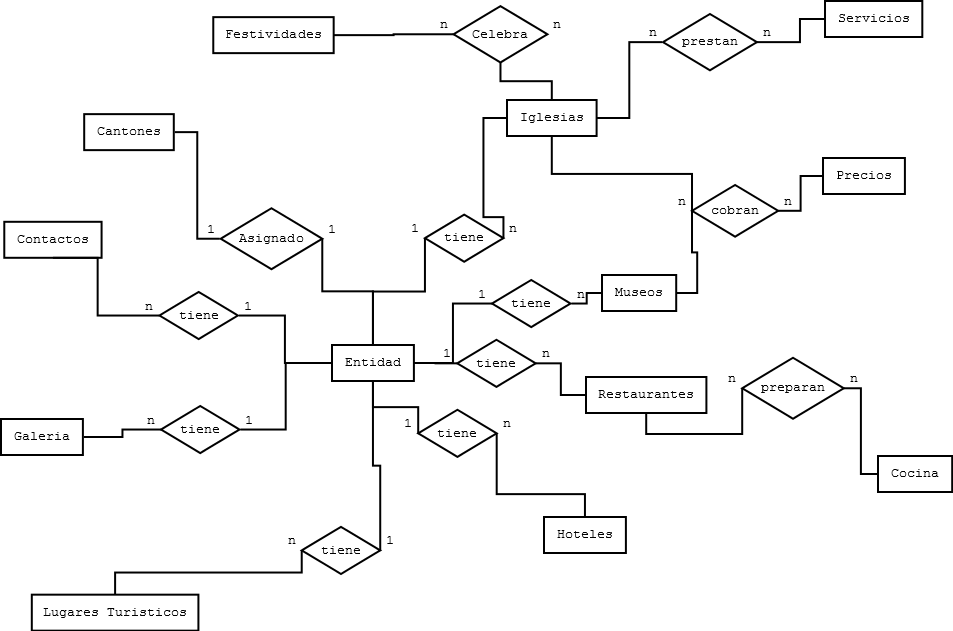


Gráfico 1-3 Diagrama Entidad Relación

**Realizado por:** Milton Jiménez y Lorena Hidalgo

El diagrama entidad relación identifica las entidades principales que son: Cantones, Festividades, servicios, precios, iglesias, muesos, restaurantes, hoteles, lugares turísticos, galería, entidad y contactos. Se realiza una normalización de los datos que llego hasta la tercera forma normal

De ahí se realiza el diseño lógico donde se define la cantidad de tablas que se agregan por la relación n a n es el caso de la tabla iglesia-festividad, iglesia-servicio, iglesia-precio, museo-precio, restaurante-cocina, entidad-contactos, dando un resultado de 22 tablas relacionadas

Se crea un diagrama físico que está desarrollada en el gestor de base de datos Postgres con tablas y procedimientos almacenados considerados importantes para mejorar las transacciones y evitar perdida de información, las funciones establecidas para la gestión de datos son implementadas para la creación, modificación, eliminación y listado de objetos, por último se genera el diccionario de datos cuyos principales tipos de datos son string (en Postgres character varying), por el tipo de información que se quiere mostrar en la aplicación móvil, además el tipo de dato Integer que define claves primarias y son utilizadas como llaves secundarias en las tablas n a n.

En conclusión la base de datos que va a gestionar la información del proyecto perteneciente al GAD de la provincia de Chimborazo no necesita una gran volumen de almacenamiento en disco con la capacidad de 500 GB es suficiente para 5 años de almacenamiento.

Se recomienda disponer de nuevos servidores de base de datos en el caso de gestionar imágenes de mayor calidad.

HT2: Como desarrollador quiero crear un sitio web para la administración de datos

En esta historia se asignan 14 horas para la creación de una aplicación web y para esto se definen 3 tareas de ingeniería. Para realizar las pruebas de aceptación se ejecuta la aplicación y se realiza una prueba de conexión de datos. Se toma en consideración los puntos definidos para buenas prácticas de codificación y se pretende hacer uso de la mejor manera de nuestros hábitos de programación.

HT3: Como desarrollador quiero definir un estándar de codificación, para utilizarlo como modelo o patrón de referencia.

Para esta historia se asigna un tiempo estimado de 4 horas y la tarea de ingeniería define un estándar de codificación para la aplicación, basado en el documento Convenciones de código para el lenguaje de programación JavaTM, (Sun Microsystem, 1999,p10), y considera importante definir un estándar por las siguientes razones:

* El 80% del coste del código de un programa va a su mantenimiento.
* Casi ningún software lo mantiene toda su vida el auto original.
* Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo.
* Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.
* Para que funcionen las convenciones, cada persona que escribe software debe seguir la convención.

HT4: Como desarrollador quiero definir un diseño de interfaces para establecer una referencia en diseño

Para esta historia se estima 8 horas para el desarrollo de las tareas asignadas donde se seleccionaron como colores principales el gris y el azul marino ya que se considera que son tonos sobrios y elegantes, a la vez que no cansan tanto la vista luego de varias horas de fijación ante la pantalla.

Historia de usuario de 1 a la 14

Corresponde al desarrollo de software se lo realiza utilizando la herramienta de desarrollo Netbeans, con el lenguaje de programación java, con lo que se obtiene el código que reposa en los servidores del GADCH, la codificación se basa en el estándar mencionado en la historia técnica 2.

Luego de terminado todas las tareas del primer sprint se realiza la reunión de revisión del sprint. En esta reunión, los desarrolladores presentan el producto entregable que han implementado y, los gestores, clientes, usuarios y Product Owner analizan esta entrega y escuchan al equipo sobre los problemas que han tenido durante el proceso.

En el desarrollo del primer Sprint se presentan las metas cumplidas, las funcionalidades y las críticas recibidas como se muestra en la gráfico 2-3.

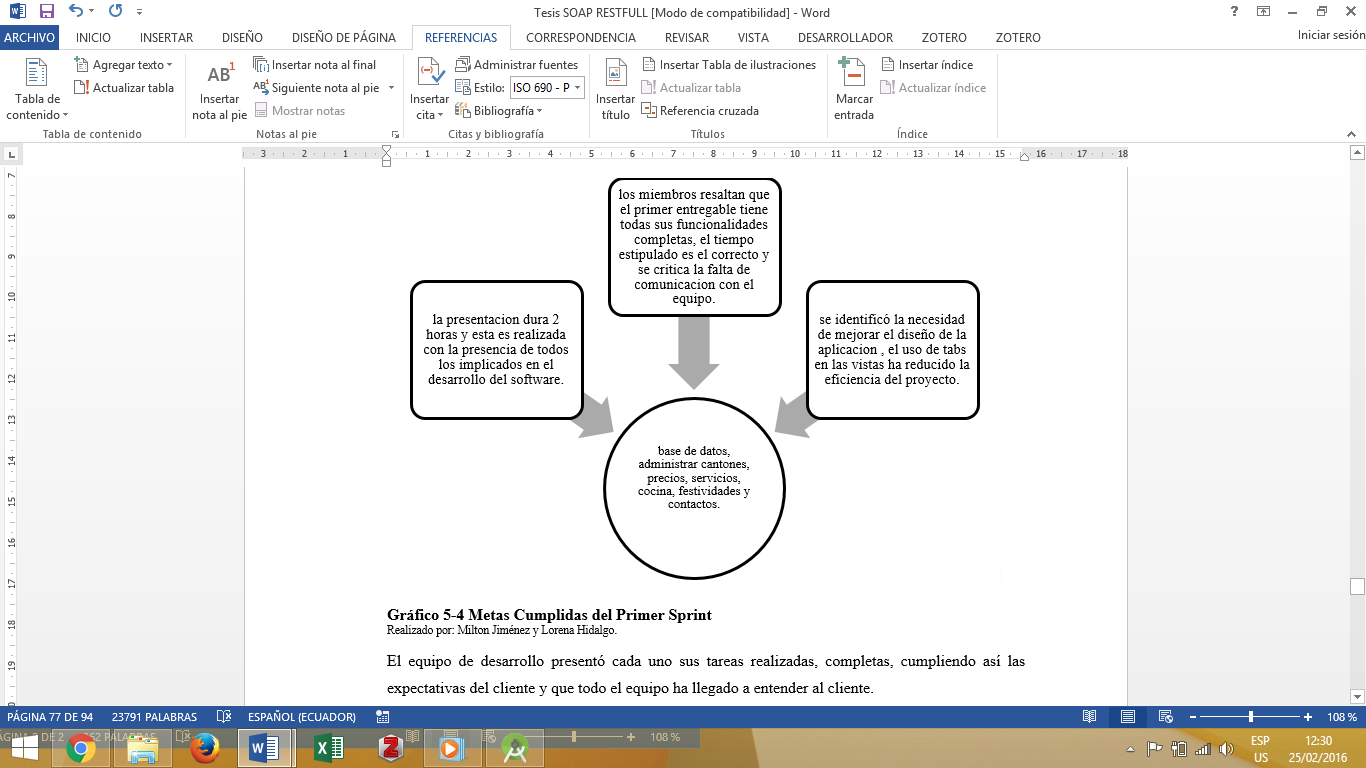


Gráfico 2-3 Metas Cumplidas del Primer Sprint

Realizado por: Milton Jiménez y Lorena Hidalgo.

El equipo de desarrollo presentó cada uno sus tareas realizadas, completas, cumpliendo así las expectativas del cliente y que todo el equipo ha llegado a entender al cliente. Es importante en este punto del desarrollo del proyecto tener amplios conocimientos e incentivar la investigación sobre las herramientas utilizadas y sobre las funcionalidades que estas prestan para la generación, diseño y presentación del proyecto. Las mejoras para el siguiente sprint se definen en reunión de retrospectiva

**Reunión de retrospectiva**

En esta reunión el equipo debate sobre el sprint ya finalizado y los cambios que se pueden aplicar al segundo sprint para que sea más productivo. Los resultados presentaron las posibles mejoras y se definen a continuación:

* El uso de tabs en diseño de la aplicación web, será corregido dentro del siguiente sprint: investigar y aplicar conocimientos sobre el framework utilizado se convierte en un avance para el siguiente sprint, además de llegar al acuerdo de reforzar la comunicación del grupo.
* El desarrollo asertivo de las tareas del primer sprint presentan un informe de los pasos que se están realizando y los problemas que ha traído realizarlos, además los resultados servirá como herramienta y base para el siguiente sprint.

### 3.1.6 Sprint 2

En la reunión realizada cada semana se muestran los cambios que se generaron en el primer sprint sobre investigar más sobre funcionalidades de las herramientas utilizadas resultan en una agradable presentación del diseño y color de la aplicación web. En cada reunión los componentes eran mejorados. El equipo comparte toda la información del segundo sprint, generando y adaptado cambios que se dan dentro del desarrollo y así aumentar la productividad del entregable, entregando un informe de las labores de los integrantes del equipo

Historia de Usuario 15 a la 34 corresponde al desarrollo de software cada una de las tareas se definen en el manual técnico, la codificación que reposa en los servidores del GADCH, la codificación se basa en el estándar mencionado en la historia técnica 2. Los resultados servirán como herramienta y base para las siguientes tareas.

**Reunión revisión del sprint (Sprint Review Meeting)**

En esta reunión, los desarrolladores presentan el producto entregable que han implementado y, los gestores, clientes, usuarios y Product Owner analizan esta entrega y escuchan al equipo sobre los problemas que han tenido durante el proceso.

El gráfico 3-3 siguiente representa las metas cumplidas, las funcionalidades y las críticas recibidas.

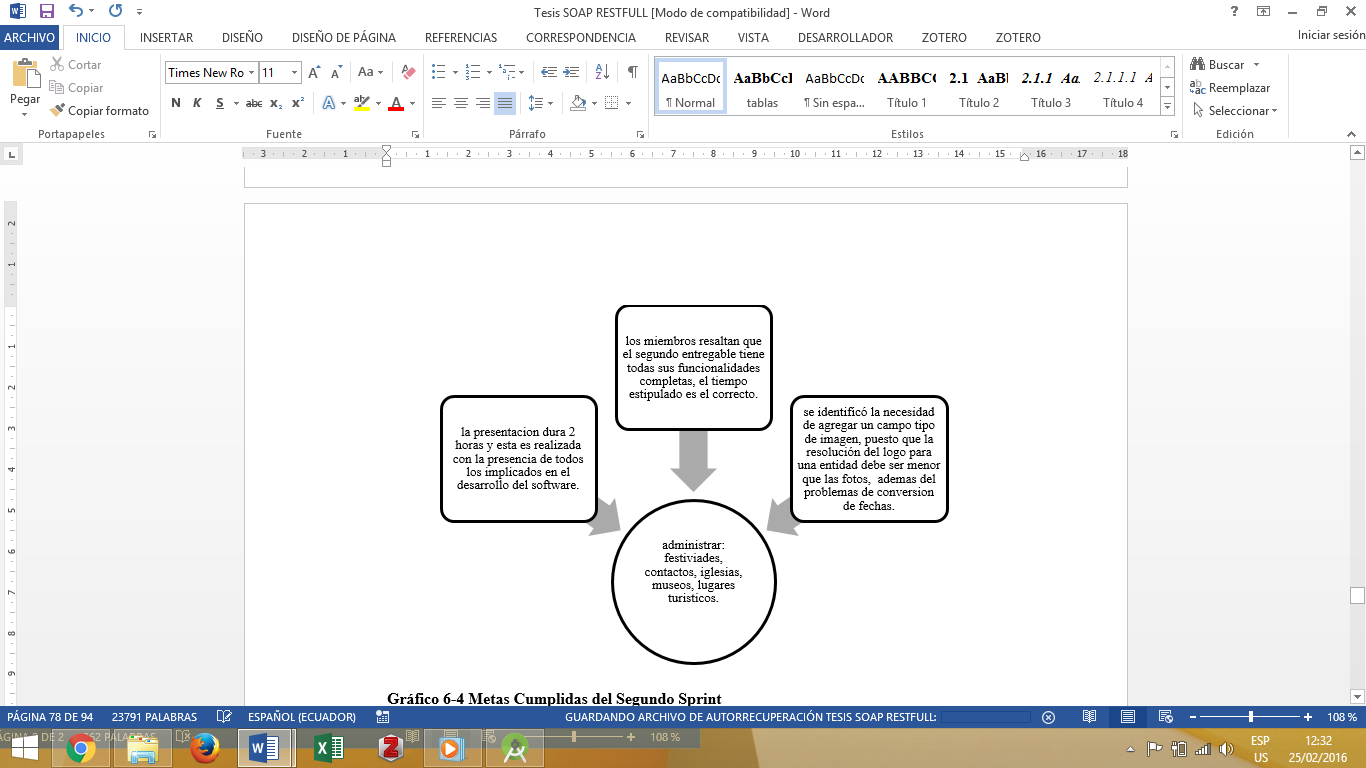


Gráfico 3-3 Metas Cumplidas del Segundo Sprint

**Realizado por:** Milton Jiménez y Lorena Hidalgo.

Las tareas realizadas por el equipo de desarrollo en este sprint fueron completadas en el tiempo establecido, las modificaciones se realizarán en el siguiente sprint.

**Reunión de retrospectiva**

La reunión dio como resultado la mejora en el ingreso de una entidad; una propiedad de un componente dentro del diálogo llamado Ajax es utilizado para ingresar la imagen que representa a una entidad, de esto resultó que la imagen debe ser de una resolución más baja que las fotografías, luego este logo de la entidad debe ser ingresada en el siguiente módulo de galería para evitar que el componente Ajax dentro del diálogo de ingreso de una entidad cause cambios en la funcionalidad de la propiedad action.

### 3.1.7Sprint 3

En este sprint se definen 14 historias de usuario y 2 historias técnicas, para ello se asignan un tiempo de desarrollo y se obtienen los siguientes resultados:

HU35 a la HU49

Corresponde al desarrollo de software cada una de las tareas se definen en el manual técnico, la codificación se basa en el estándar mencionado en la historia técnica 2.

HT5: Como desarrollador quiero documentar el sistema para utilizarlo como fuente informativa

Para la documentación del sistema se estima 34 horas. En esta etapa se termina con la creación de la aplicación web y además se presentan lo manuales de instalación, técnicos y de usuario estos manuales fueron desarrollados para presentarla información básica sobre la instalación y uso de la aplicación, además de esto se consideró importante crear el manual técnico para un mantenimiento futuro de la aplicación

HT6: Como usuario quiero capacitación para la utilización del sistema

La capacitación de la aplicación web se realiza en 2 horas y se prepara al usuario en la utilización el sistema web.

Las reuniones anteriores, las correcciones y mejoras realizadas permitieron que las tareas del tercer sprint pudieran ser completadas sin un mayor problema.

**Reunión revisión del sprint (Sprint Review Meeting)**

El producto es entregado por los desarrolladores a los clientes y usuarios. El grafico 4-3 representa las metas cumplidas, las funcionalidades y las críticas recibidas

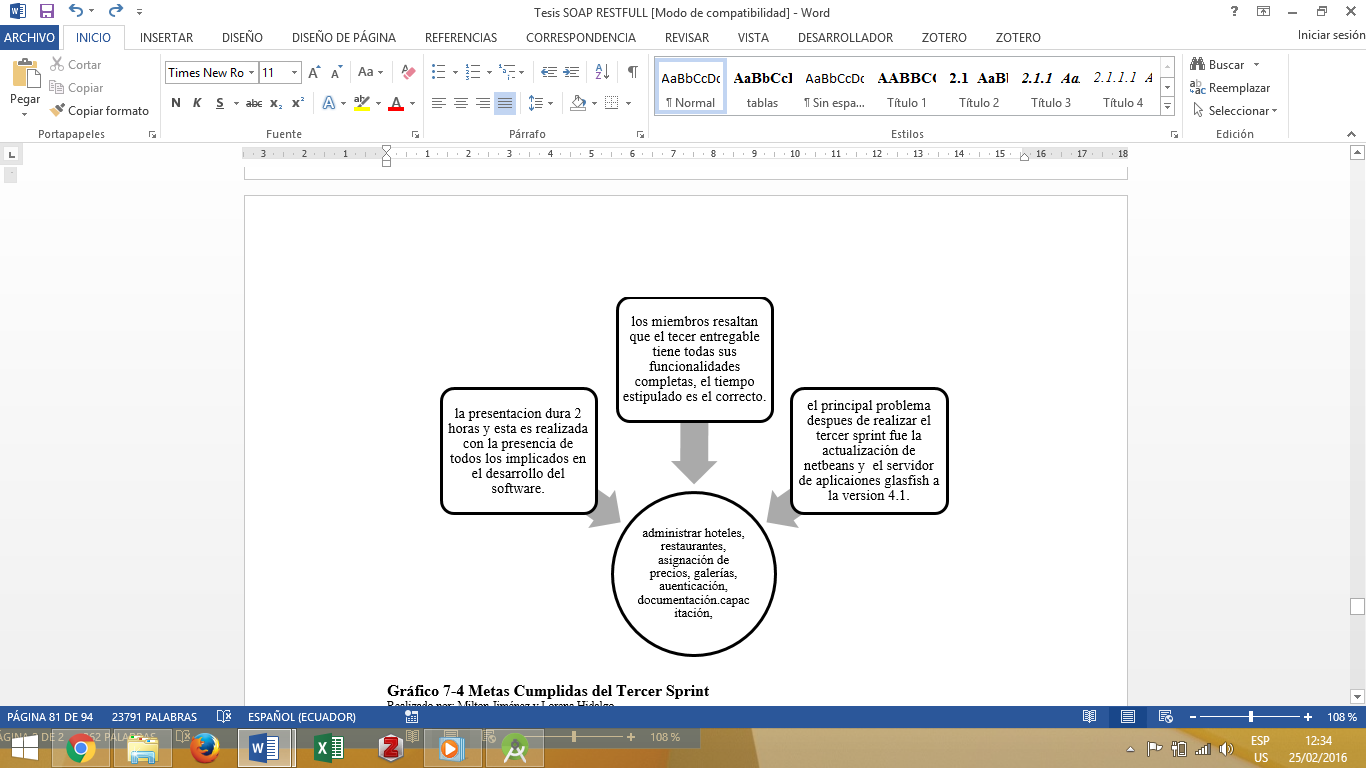


Gráfico 4-3 Metas Cumplidas del Tercer Sprint

**Realizado por:** Milton Jiménez y Lorena Hidalgo.

El equipo de desarrollo presentó la problemática existente al realizar las tareas, aún con las actualizaciones la aplicación web tiene cada una sus tareas realizadas, completas, cumpliendo así las expectativas del cliente.

**Reunión de retrospectiva**

Esta reunión se forma el debate de las actualizaciones de las versiones de las herramientas, en este caso el IDE de desarrollo Netbeans se actualizó a la versión 8.0 junto con el servidor de aplicaciones Glassfish a la versión 4.1. El principal problemas de las actualizaciones es que los frameworks utilizados no son compatibles y es necesario investigar y aplicar nuevas librerías para su correcto funcionamiento. Las funcionalidades del proyecto son correctas, en el siguiente paso se realizarán las pruebas y la implementación de la aplicación web.

**3.1.8 Sprint 4**

El desarrollo del Sprint 4 con sus respectivas historias de usuario, tareas y gráfica de seguimiento se encuentran en el manual técnico.En las reuniones del cuarto sprint en el desarrollo de la aplicación móvil se participa con el grupo el diseño de la aplicación y el consumo de los servicios web, esto se llevó a cabo hasta la finalización del cuarto sprint según el cronograma expuesto.

HT7: Como desarrollador quiero crear una aplicación móvil para mostrar información

Para la creación de la aplicación móvil de asigna 6 horas y dos tareas que se verifican al ejecutar la aplicación móvil.

HT8: Como desarrollador quiero definir un diseño de interfaces para establecer una referencia en diseño

Esta historia tiene un tiempo estimado de 8 horas, y para la aplicación de utiliza colores vivos y así aprovechar de una apariencia llamativa como fondo se utiliza un color fucsia, con una fuente para títulos Times New Roman número 14 y según sea la utilización llega hasta un tamaño de letra 8. Los botones se encuentran representados por iconos referentes a la acción a ejecutar y las imágenes son representativas al ámbito de la aplicación.

Historia Técnica 50 a 56:

Corresponde al desarrollo de software cada una de las tareas se definen en el manual técnico,la codificación que reposa en los servidores del GADCH, la codificación se basa en el estándar mencionado en la historia técnica 2.2.

Todas las reuniones realizadas y las correcciones mejoraron la apariencia y permitieron que las tareas del sprint sean completadas sin problemas.

**Reunión revisión del sprint (Sprint Review Meeting)**

Esta reunión se realizó con la presencia de los desarrolladores que presentan el producto entregable que han implementado y, los gestores, clientes, usuarios y Product Owner analizan esta entrega y escuchan al equipo sobre los problemas que han tenido durante el proceso.

El gráfico 5-3 siguiente representa las metas cumplidas, las funcionalidades y las críticas recibidas.

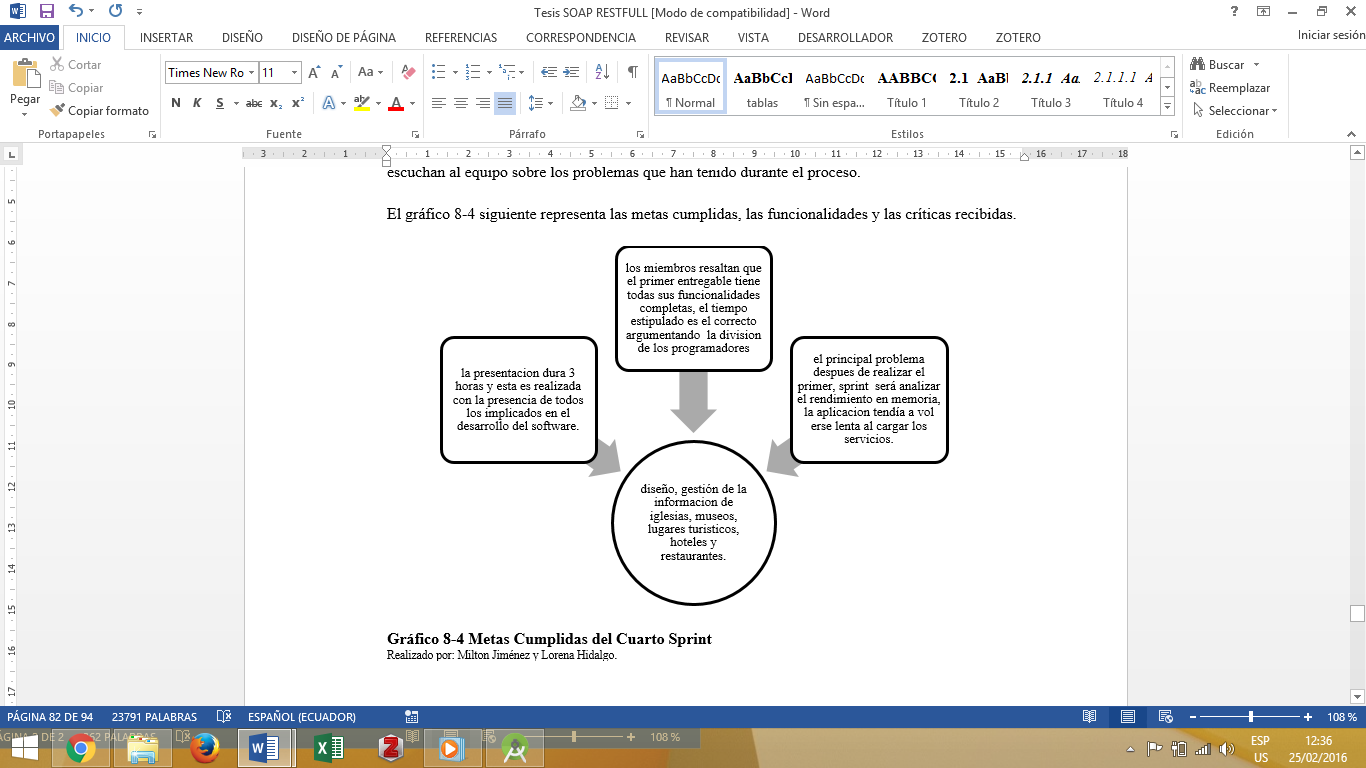


Gráfico 5-3 Metas Cumplidas del Cuarto Sprint

**Realizado por**: Milton Jiménez y Lorena Hidalgo.

El equipo de desarrollo presenta las funcionalidades de primer sprint completas, cumpliendo con el cronograma y con las expectativas del cliente.

**Reunión de retrospectiva**

La reunión se realiza con la disposición de mejorar el tema de rendimiento en la memoria del dispositivo, la investigación se realizará en el lapso que dure el proyecto, es importante especificar que en esta reunión se presentó los posibles temas a investigar para el desarrollo eficiente de la aplicación.

1. Es importante tener en cuenta la versión del dispositivo, el diseño presentado es compatible en una versión Android 2.3 en adelante, para preservar la definición de colores.
2. Además de la versión del dispositivo verificar la capacidad de memoria y como es asignada a diferentes procesos según el dispositivo.
3. La herramienta de desarrollo para Android está siendo actualizada así que los cambios al final de la aplicación deberán ser empleadas.

Las funcionalidades del proyecto son correctas, en el siguiente paso se presentarán lugares en realidad aumentada.

### 3.1.9 Sprint 5

El desarrollo del Sprint 4 con sus respectivas historias de usuario, tareas y gráfica de seguimiento se encuentran en el manual técnico. Las historias con prioridad media son 8 y serán realizadas según el cronograma expuesto.

Historia de usuario 57 a 64 corresponde al desarrollo de software cada una de las tareas se definen en el manual técnico, la codificación que reposa en los servidores del GADCH, la codificación se basa en el estándar mencionado en la historia técnica 2.2.

HT9: Como desarrollador quiero documentar el sistema móvil para utilizarlo como fuente informativa

Para la documentación del sistema se estima 16 horas. En esta etapa se termina con la creación de la aplicación móvil y además se presentan lo manuales técnico y de usuario estos manuales fueron desarrollados para presentarla información básica sobre la instalación y uso de la aplicación, además de esto se consideró importante crear el manual técnico para un mantenimiento futuro de la aplicación

HT10: Como usuario quiero capacitación para la utilización del sistema móvil

La capacitación de la aplicación móvil se realiza en 2 horas y se prepara al usuario en la utilización de la aplicación.

Las reuniones anteriores, las correcciones y mejoras realizadas permitieron que las tareas del tercer sprint pudieran ser completadas sin un mayor problema.

En las reuniones realizadas para el seguimiento de este sprint tuvieron como resultado diferentes análisis sobre las actualizaciones del software utilizado para la aplicación móvil, junto con ellas el cambio de librerías y frameworks. Además de esto se presenta un manual técnico y de usuario(ver cd anexo), importante para uso y mantenimiento de la aplicación móvil.

**Reunión revisión del sprint (Sprint Review Meeting)**

Para la entrega del sprint se consideró la solución de problemas de compatibilidad de software, en esta reunión, los desarrolladores presentan el producto entregable que han implementado y, los gestores, clientes, usuarios y Product Owner analizan esta entrega y escuchan al equipo sobre los problemas que han tenido durante el proceso.

El gráfico 9-4 siguiente representa las metas cumplidas, las funcionalidades y las críticas recibidas.

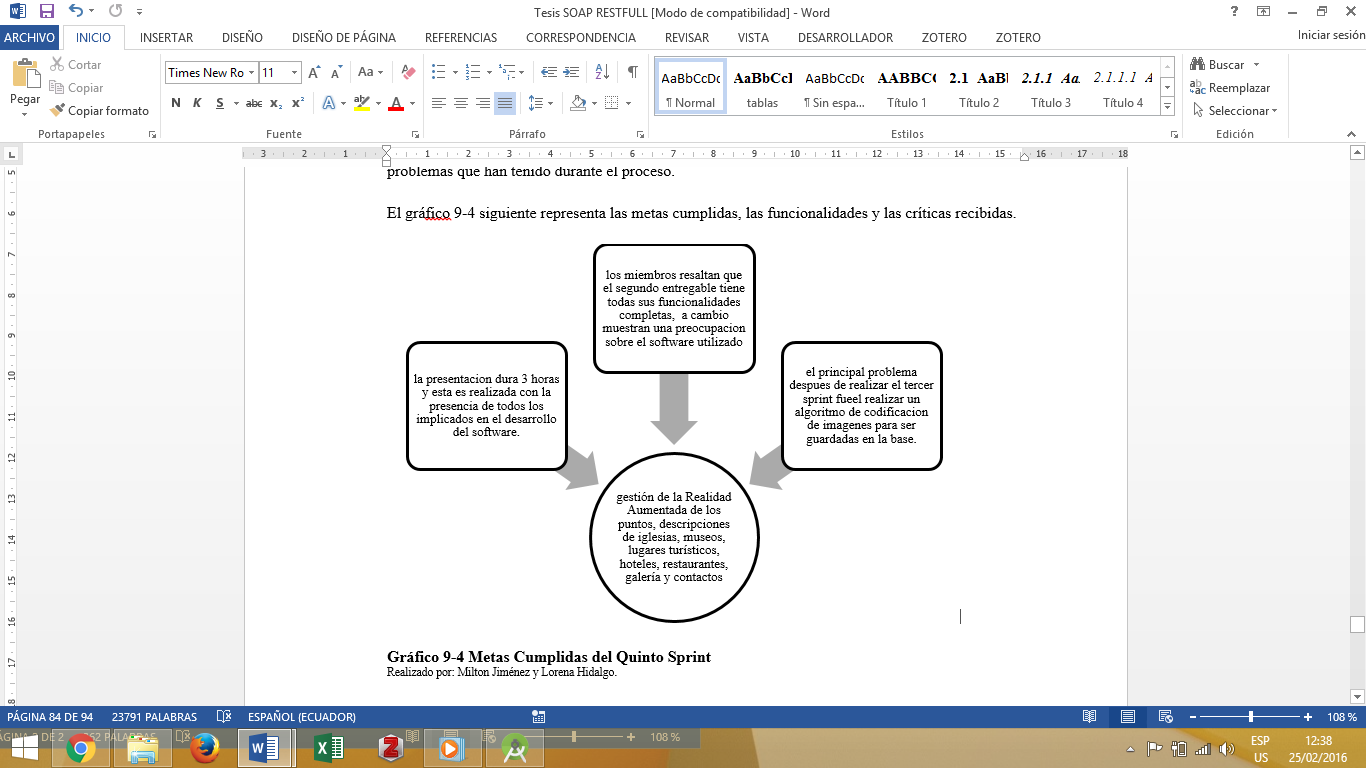


Gráfico 6-3 Metas Cumplidas del Quinto Sprint

**Realizado por:** Milton Jiménez y Lorena Hidalgo.

El equipo de desarrollo presentó cada uno sus tareas realizadas, completas, y resuelven los problemas en el desarrollo y presentación de la aplicación.

**Reunión de retrospectiva**

La reunión luego de analizar los problemas en el desarrollado son considerados para la implementación del proyecto:

1. El API de Android se actualiza y el servicio de mapas de google play services junto con él.
2. Para guardar una imagen subida desde el dispositivo a la base de datos se consideró al gestor de imágenes llamada glide, esta librería permite gestionar imágenes con el menor consumo de memoria del dispositivo.
3. La herramienta de realidad aumentada Wikitude es actualizada de la versión 4 a la versión 4.1, la forma de agregar esta librería es muy diferente a la de la versión anterior esto provoca problemas en cuestión de tiempo para investigar soluciones.

Las funcionalidades del proyecto se cumplen con el tiempo estipulado con las mejoras que sonaplicadas en los casos definidos anteriormente.

### 3.1.10 ***Gestión del proyecto***

Para medir la velocidad de desarrollo de las tareas del Sprint en días se desarrolló un gráfico de seguimiento del proyecto llamado Burdown chart que describe las tareas realizadas y el tiempo que lleva ejecutarlas y este se indica en el gráfico 7-3.

Gráfico 7-3 Gráfica de Seguimiento del Proyecto

**Realizado por** Milton Jiménez y Lorena Hidalgo

La gráfica se muestra cómo los ítems seleccionados para cada sprint del proyecto son suficientes para ser desarrollados en el tiempo establecido, las actividades son realizadas en casi según el tiempo ideal, pero al finalizar el proyecto se llega a cumplir con las tareas establecidas, se recomienda desarrollar todo el proyecto con el mismo ritmo de desarrollo para obtener mejores resultados

Es importante tomar en cuenta que ante actualizaciones de herramientas de desarrollo la incompatibilidad de librerías en muchas ocasiones es inevitable, para ello la capacidad de los desarrolladores para resolver este tipo de problemas resulta ser imprescindible; durante todo el proceso de desarrollo el equipo maneje destrezas de investigación y resolución de problemas, mejorando cada vez más la expectativa de éxito en la implementación del sistema a desarrollarse.

En el desarrollo de cualquier proyecto en este caso de aplicaciones es importante formar un equipo dedicado al avance del mismo, cada integrante toma su lugar y proyecta confianza en el proceso de investigación sobre todo cuando existen dificultades en las herramientas de software como actualización de frameworks, librerías, nuevas versiones, etc. Además tener en cuenta que así como representan un problema, estas actualizaciones pueden mejorar características del proyecto como apariencia, rendimiento, y minimizar código como en los casos de importación de librerías.

La aplicación web muestra una estructura escalable, cumple la función de administrar los sitios de interés dentro de una Provincia, gestiona puntos de referencia GPS utilizada por la aplicación móvil, presentando una galería del sitio de interés. Dentro de la aplicación móvil se presenta los datos obtenidos gracias a servicios web, en ese punto del proyecto es necesario desarrollar y presentar una interfaz llamativa y sobretodo con una navegabilidad dirigida a diferentes versiones de dispositivos.

Se recomienda administrar correctamente los puntos de posición del lugar turístico reconociendo latitud y longitud, esto por la referencia que se realiza en el mapa que se utiliza en la aplicación, si llegara a ocurrir una mala obtención de estos puntos, por la ubicación del Ecuador al tomar al contrario los puntos el mapa se direcciona en algún punto del Antártida. Para mejorar la aplicación móvil es interesante investigar sobre base de datos offline, para nuevas versiones aplicar la base de datos SQL lite es una buena opción.

**CONCLUSIONES**

* Con el estudio de los servicios web se identificaron las partes, elementos y estructura que lo conforma para mejorar la escalabilidad de las aplicaciones, y la interoperabilidad entre ellas.
* Las características que presentan los servicios web RestFull con Jersey en rendimiento para sistemas distribuidos hace que estos utilicen pocas operaciones y este tipo de arquitectura es particularmente útil en dispositivos con escasos recursos como PDAs o teléfonos móviles.
* Las funcionalidades de SOAP con JAX-WS se centran en el diseño de aplicaciones distribuidas con muchas operaciones, la sobrecarga de las cabeceras y capas adicionales de los elementos debe ser restringida puesto que encapsula el mensaje y adiciona datos para el envío de este.
* En el estudio del rendimiento realizada los servicios web RestFull con Jersey se obtiene un promedio con una diferencia de 832,3 milisegundos que corresponde a un 5.88% menos que el promedio del tiempo de respuesta obtenido en los servicios web SOAP JAX-WS.
* El uso de RestFull con Jersey para desarrollo de una aplicación Móvil, permitió obtener tiempos de respuesta bajos por lo que fue implementado en el sistema de Turismo de la Provincia de Chimborazo.

# RECOMENDACIONES

* Se recomienda la utilización de los Servicios web RestFull con Jersey en la implementación de aplicaciones móviles que por su estructura ligera que optimiza recursos y pueden ser aplicada en servicios de medios sociales, redes sociales, servicios web de chat y servicios móviles.
* Utilizar SOAP para servicios financieros, de telecomunicación y transacciones bancarias.
* Se recomienda mejorar la documentación del API Jersey para RestFull
* En la metodología ágil Scrum utilizada presentó dificultades en la priorización de historias de usuario, se recomienda priorizar en una escala del 1 al 10 para mejorar la selección para identificar el orden de desarrollo.
* El tiempo que demanda crear los componentes de la interfaz de usuario de la aplicación móvil fue bastante lo que se recomienda incorporar un framework para mejorar la apariencia visual.
* Para la compatibilidad de la aplicación en distinto sistemas operativos se recomienda la creación de una aplicación híbrida.

**BIBLIOGRAFÍA**

**Alan Bover Argelaga.** . *APLICACIÓN DE GESTIÓN DE INFORMACIÓN GEOLOCALIZADA EN ANDROID* [en línea]. Barcelona - España 2010 [Consulta: 17 de febrero de 2014]. [disponible en:https://www.researchgate.net/publication/277123988\_Aplicacion\_de\_gestion\_de\_informacion\_geolocalizada \_en\_Android]

**AlShahwan, F., Moessner, K., & Carrez** E*valuation of Distributed SOAP and RESTful MobileWeb Services. International Journal on Advances in Networks and Services* [en línea]. London-Surrey 2010 [disponible en: http://epubs.surrey.ac.uk/125638/], pp 447-461.

**Android, A.**| *Android Developers*.[en línea][consulta: 17 de febrero de 2014], [disponible en:http://developer.android.com/intl/es/about/dashboards/index.html]

**Augmented Reality** | *Interactive Print* | Layar. (2015). [en línea].[Consulta: 18 de febrero de 2014], [disponible en: https://www.layar.com/]

**Burke, B.,***RESTful Java with JAX-RS 2.0*. O’Reilly Media 2013, [en línea]. [disponible en: http://shop.oreilly.com/product/0636920028925.do]

**Cabrera, Y. V.** *Transferencia de estado representacional (REST): estilo de arquitectura para sistemas distribuidos de hipermedia. Serie Científica-Universidad de las Ciencias Informáticas,*.[En línea]. Cuba 2013. [Disponible en: http:/publicaciones.uci.cu/index.php/SC/article/ view/1076]

**Carmona Barbero,.***Plataformas De Integración.Servicios Web REST y SOAP.* [en línea] [consulta: 17 de febrero de 2014,][disponible en:https://www.google.com/search?q=Plataformas\_de\_ integraci%C3%B3n.Servicios\_Web\_REST\_y\_SOAP.pdf&ie=utf-8&oe=utf#q=Plataformas+De+Integraci%C3%B3n.Servicios+Web+REST+y+SOAP.pdf]

**Corales Muñoz Victor.**. *XML LENGUAJE DE PROGRAMACIÓN.*[en línea]. Madrid-universidad Carlos III. [consulta: 17 de febrero de 2014], disponible en: http://biblioteca.versila.com/]

**Crespo, J., & Eduardo.***Prototipo de servicios web móviles.*[en línea].Universidad de Cuenca. Cuenca, 2010, pp 3-18,[Disponible en: http://dspace.ucuenca.edu.ec/handle/123456789/2541]

**Darwin, I. F.***Android Cookbook*. O’Reilly Media, Inc.[en línea]. "O'Reilly Media, Inc.", 2012 [disponible en:https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxh bmRs ZXNzb25zfGd4OjE0OGYwNWVmMDE1ZjQ0OWY].

**Fielding, R. T., & Taylor,** *Principled design of the modern Web architecture. ACM Transactions on Internet Technology (TOIT),2*(2), 115-150.[En línea].  New York, NY, USA 2015. [Disponible en: http://doi.org/10.1145/514183.514185]

**Jerez Fernández,** *Implementación de un cliente REST para un servicio de almacenamiento de ficheros basado en metadatos.* [En línea]. Cartagena*,* febrero 2015*,* [Disponible en: http://repositorio.upct.es//handle/10317/4561]

**Junaio.** *What is Junaio?.* [Consulta: 18 de febrero de 2014] [ disponible en: https://my.metaio.com/dev/junaio/]

**Silva de la Cruz, A. da, & Silva de la Cruz.***Una aproximación MDA para la conversión entre servicios web SOAP y RESTful*. [en línea]. Universidad Complutense De Madrid.[consulta 17 de febrero de 2014], [disponible en: http://eprints.ucm.es/23165/]

**Snell, J., Tidwell, D., & Kulchenko,.***Programming Web Services with SOAP*. O’Reilly Media, Inc. [en línea].

**Steven Davelaar,**. *Performance Study – REST vs SOAP for Mobile Applications.*[en línea]. Oracle [Consulta:15 de febrero de 2014] [disponible en: http://www.ateam-oracle.com/performance-study-rest-vs-soap-for-mobile-applications/]

**Vohra, D.** *Java 7 Jax-Ws Web Services*. Packt Publishing Ltd. [disponible en: http://www.amazon.es/Java-7-JAX-WS-Web-Services-ebook/dp/B00850BHFY]

**W3C, W.** *Web Services @ W3C*. [en línea]. [Consulta: 17 de febrero de 2014] [disponible en: https://www.w3.org/2002/ws/]

**Wikitude,** G. (2015). Wikitude - *The World’s leading Augmented Reality SDK.*[en línea].[consulta: 15 de febrero de 2014, [disponible en: http://www.wikitude.com/].

**ANEXOS**

**Anexo A: Tabla valores resultados**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Resultados tiempos de respuesta RestFull (milisegundos)** | | | | **Resultados tiempos de respuesta SOAP (milisegundos)** | | | |
| 6490 | 6757 | 7286 | 6995 | 6535 | 6703 | 8920 | 8476 |
| 6417 | 6471 | 7135 | 6786 | 6835 | 7385 | 7343 | 6742 |
| 5942 | 6742 | 7529 | 8426 | 6444 | 8376 | 8208 | 6916 |
| 4764 | 6629 | 7289 | 6590 | 5707 | 8232 | 7090 | 6911 |
| 5413 | 6738 | 6536 | 6635 | 5808 | 6631 | 7366 | 6847 |
| 5855 | 6722 | 7193 | 6580 | 6873 | 7321 | 8347 | 7532 |
| 5981 | 6678 | 6717 | 6838 | 6221 | 7128 | 7351 | 7894 |
| 6121 | 6646 | 7066 | 6991 | 6186 | 6919 | 7527 | 6626 |
| 6005 | 7760 | 7833 | 6779 | 6560 | 7187 | 7433 | 6614 |
| 6649 | 6269 | 6648 | 6745 | 5777 | 6715 | 7609 | 6876 |
| 6477 | 5684 | 7560 | 7022 | 7634 | 6804 | 7053 | 8423 |
| 6297 | 6231 | 7006 | 6911 | 6675 | 6519 | 8040 | 6498 |
| 7158 | 5882 | 6758 | 8037 | 6160 | 7572 | 6970 | 6280 |
| 7304 | 6466 | 8328 | 5774 | 7323 | 7427 | 6744 | 7533 |
| 6690 | 5962 | 6547 | 6068 | 7552 | 6503 | 8005 | 7135 |
| 7898 | 6409 | 7739 | 7368 | 9435 | 7055 | 7283 | 7236 |
| 3788 | 7014 | 6788 | 7388 | 4967 | 6599 | 7436 | 6045 |
| 6346 | 5439 | 6367 | 7542 | 6888 | 7552 | 7665 | 6629 |
| 6999 | 6826 | 6376 | 6763 | 8823 | 6312 | 7159 | 7123 |
| 7942 | 6703 | 8087 | 6661 | 7174 | 7240 | 5846 | 7089 |
| 8461 | 5625 | 7425 | 7410 | 7453 | 6088 | 7433 | 8356 |
| 8259 | 7243 | 7682 | 6257 | 6948 | 7403 | 6600 | 6005 |
| 7618 | 6538 | 6350 | 7109 | 7766 | 6894 | 7132 | 6878 |
| 7629 | 7010 | 7165 | 6487 | 7863 | 6289 | 7199 | 7486 |
| 7711 | 5962 | 7850 | 6083 | 7990 | 6492 | 7817 | 7424 |
| 5367 | 6440 | 6353 | 6786 | 8562 | 7550 | 6776 | 7913 |
| 7237 | 6911 | 6103 | 6893 | 7759 | 6430 | 6644 | 6927 |
| 4235 | 7824 | 7622 | 7689 | 4987 | 6869 | 7850 | 6042 |
| 5541 | 6387 | 6832 | 7140 | 6426 | 6671 | 8070 | 6462 |
| 7605 | 6280 | 7919 | 5945 | 7740 | 7483 | 6711 | 6071 |
| 5754 | 7390 | 6559 | 6639 | 8541 | 8277 | 8083 | 6889 |
| 6952 | 6801 | 7147 | 5954 | 7405 | 9510 | 6735 | 7214 |
| 8081 | 7564 | 9928 | 6448 | 7171 | 7930 | 7909 | 7034 |
| 7077 | 8758 | 6362 | 5946 | 7419 | 6888 | 7003 | 6600 |
| 6763 | 8682 | 6280 | 5624 | 7480 | 7146 | 7198 | 7347 |
| 7202 | 7720 | 6025 | 6147 | 8031 | 7833 | 6976 | 6536 |
| 7187 | 7536 | 7432 | 6665 | 9228 | 6983 | 6685 | 7256 |
| 6754 | 7779 | 7092 | 6046 | 6985 | 8158 | 7854 | 6355 |
| 7461 | 7455 | 6934 | 9746 | 6215 | 7552 | 8450 | 6791 |
| 6805 | 7004 | 6870 | 5711 | 7730 | 8070 | 7212 | 6883 |
| 6545 | 8993 | 7552 | 6364 | 7534 | 7954 | 7127 | 6381 |
| 6701 | 7878 | 6705 | 6009 | 7762 | 7909 | 7853 | 6562 |
| 7547 | 7566 | 7191 | 5733 | 7078 | 8191 | 6399 | 6519 |
| 8427 | 7766 | 7044 | 5789 | 7273 | 7858 | 7712 | 7110 |
| 7513 | 8061 | 6488 | 7092 | 6891 | 8829 | 6785 | 5478 |
| 6471 | 7143 | 7322 | 6912 | 7731 | 8824 | 7001 | 6452 |
| 7115 | 7311 | 7178 | 6798 | 7305 | 7777 | 7364 | 5855 |
| 7428 | 8500 | 6125 | 6098 | 7973 | 7951 | 8090 | 8414 |
| 7474 | 7801 | 6071 | 6524 | 7043 | 8258 | 7327 | 6577 |
| 8151 | 6652 | 6782 | 7354 | 7650 | 7397 | 6968 | 7412 |
| 8358 | 6864 | 7287 | 6881 | 7124 | 7418 | 7434 | 7719 |
| 7046 | 6919 | 6537 | 6367 | 7041 | 8889 | 6855 | 7651 |
| 6517 | 7641 | 5992 | 7192 | 6735 | 7166 | 7475 | 8631 |
| 6368 | 7019 | 6664 | 6984 | 7962 | 7834 | 8168 | 7325 |
| 7155 | 7191 | 6514 | 7846 | 7213 | 6956 | 7666 | 6302 |
| 7119 | 7438 | 6807 | 7980 | 6574 | 8042 | 6290 | 6680 |
| 6772 | 7362 | 6534 | 6748 | 7326 | 6828 | 8670 | 7050 |
| 6296 | 6548 | 7020 | 6206 | 6988 | 8412 | 7004 | 7329 |
| 6770 | 7227 | 6552 | 5725 | 8128 | 6542 | 7639 | 8285 |
| 6742 | 8230 | 6624 | 6663 | 6630 | 7808 | 7785 | 7615 |
| 7225 | 6727 | 7298 | 8324 | 7548 | 7049 | 7574 | 6257 |
| 6376 | 7049 | 7336 | 7388 | 6730 | 7612 | 6721 | 6692 |
| 6036 | 7287 | 6722 | 6000 | 6604 | 7769 | 7643 | 5802 |
| 7023 | 7179 | 7869 | 8329 | 6625 | 7281 | 7169 | 7750 |
| 7526 | 6969 | 7003 | 5084 | 6890 | 7340 | 9110 | 9291 |
| 6408 | 7919 | 6578 | 7385 | 7160 | 6839 | 6324 | 6400 |
| 6627 | 7222 | 7934 | 6092 | 5942 | 7625 | 7644 | 8497 |
| 6339 | 6899 | 6994 | 7354 | 7047 | 7452 | 8554 | 6555 |
| 7060 | 6656 | 6771 | 7055 | 6632 | 8323 | 7138 | 5701 |
| 8444 | 7405 | 6901 | 6825 | 7243 | 7797 | 8653 | 7323 |
| 5919 | 7404 | 7096 | 5897 | 7287 | 6971 | 7753 | 8156 |
| 5809 | 6392 | 7473 | 7434 | 7223 | 7627 | 7791 | 6170 |
| 6721 | 7553 | 7340 | 6735 | 8010 | 8158 | 8386 | 11820 |
| 6493 | 6081 | 6882 | 6718 | 7273 | 6765 | 6911 | 6877 |
| 7619 | 7188 | 7584 | 6540 | 7979 | 7038 | 7775 | 6960 |
| 7440 | 7135 | 6867 | 6505 | 7374 | 6465 | 7424 | 8162 |
| 8046 | 5972 | 7556 | 5906 | 7400 | 6976 | 7661 | 6511 |
| 6545 | 6852 | 6710 | 7648 | 7025 | 6885 | 6589 | 6577 |
| 6903 | 6402 | 5450 | 6346 | 8372 | 8239 | 7125 | 6970 |
| 7000 | 6199 | 7360 | 6299 | 6803 | 6303 | 7234 | 6910 |
| 9107 | 7780 | 6224 | 6499 | 6716 | 7213 | 7206 | 7462 |
| 6452 | 6002 | 7414 | 6412 | 8037 | 6880 | 7750 | 6666 |
| 6814 | 7301 | 7181 | 7189 | 8253 | 7385 | 6085 | 7158 |
| 6601 | 5934 | 6943 | 6227 | 8446 | 9141 | 7158 | 6167 |
| 7182 | 6559 | 6947 | 6429 | 7911 | 7754 | 6581 | 7204 |
| 6574 | 7985 | 5879 | 7009 | 6516 | 6953 | 7768 | 6270 |
| 8418 | 7423 | 6518 | 7080 | 7244 | 8496 | 7288 | 6886 |
| 7364 | 7543 | 6061 | 5745 | 6102 | 7242 | 8380 | 6663 |
| 6816 | 7222 | 6670 | 5975 | 7160 | 6809 | 6935 | 7032 |
| 7161 | 7457 | 6592 | 5704 | 7406 | 6771 | 6479 | 7194 |
| 6005 | 6765 | 6656 | 6829 | 6645 | 7894 | 6661 | 6456 |
| 7938 | 7068 | 6034 | 6599 | 6490 | 7735 | 7962 | 7260 |
| 5648 | 7549 | 7341 | 6591 | 7780 | 7329 | 7618 | 7169 |
| 7018 | 6718 | 7048 | 5877 | 6883 | 7991 | 7334 | 7127 |
| 6520 | 7183 | 6358 | 6046 | 7840 | 7450 | 7677 | 6777 |
| 6286 | 7028 | 6745 | 7149 | 6696 | 8142 | 7284 | 7660 |

**Anexo B**

Manual Técnico

Manual de Usuario

Manual de Instalación