



UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
ESCUELA DE INGENIERÍA DE SISTEMAS

**Desarrollo de aplicaciones SIG móviles basadas en web mediante PrimeFaces
Mobile para el manejo de información de los Bienes Inmuebles Patrimoniales de
la ciudad de Cuenca.**

**Tesis previa a la obtención del Título de:
Ingeniero de sistemas**

**Autor:
Daniel Rodríguez**

**Director
Ing. Pablo Esquivel**

**Cuenca, Ecuador
2013**

DEDICATORIA

Dedico esta tesis a mi hermano, sin el cual no tendría los conocimientos necesarios para concluirla.

A mis padres quienes estuvieron siempre a mi lado en todos los momentos de mi vida académica.

AGRADECIMIENTOS

Agradezco a mis padres por brindarme la oportunidad de lograr un título académico con su apoyo y dedicación incondicional.

A mi hermano por brindarme su tiempo y conocimientos para convertirme en lo que soy en este momento.

A la Universidad por haber permitido mi desarrollo profesional.

A todas las personas que me rodean por su paciencia y apoyo.

INDICE DE CONTENIDOS

INTRODUCCIÓN	1
Capítulo I	3
Sistemas de información geográfica en la web	3
1.1. Introducción	3
1.1.1. Tipo de información de un SIG.....	4
1.1.2. Algunas aplicaciones de los SIG.....	5
1.2. Componentes de un SIG en la web	6
1.2.1. Servidor de Mapas	6
1.2.2. Geo Base de Datos	10
1.2.3. Estándares para servicios de datos espaciales	11
1.2.4. Geoservicios.....	13
1.2.5. Aplicaciones SIG móviles.....	17
Capítulo II	19
PrimeFaces Mobile	19
2.1. Introducción	19
2.2. Arquitectura	20
2.2.1. Java Virtual Machine	20
2.2.2. Application Server	21
2.2.3. Java Server Pages.....	22
2.2.4. Java Server Faces	22
2.2.5. PrimeFaces	23
2.2.6. PrimeFaces Mobile	23
2.3. Funcionamiento.....	23
2.3.1. Java Server Faces (JSF)	24
2.3.2. Document Object Model (DOM).....	28
2.3.3. Javascript.....	29
2.3.4. Asynchronous JavaScript And XML (AJAX)	29
2.3.5. jQuery	29
2.4. Características	30
2.4.1. Ventajas.....	31
2.4.2. Desventajas	31
2.5. Requerimientos de hardware y software de PrimeFaces Mobile	32
2.6. Comparación con otras tecnologías	32

Capítulo III.....	35
Análisis y diseño de la aplicación.....	35
3.1. Análisis del sistema.....	35
3.1.1. Recolección de requerimientos	35
3.1.2. Análisis de la base de datos espacial.....	37
3.1.3. Análisis del servidor de mapas.....	43
3.2. Diseño del sistema	48
3.2.1. Diagramas de Casos de Uso.....	48
3.2.2. Diagramas de flujo de datos.....	50
3.2.3. Diagrama de clases.....	53
3.2.4. Diagrama de base de datos.....	54
3.2.5. Diccionario de datos.....	55
3.2.6. Diseño de interfaz	63
Capítulo IV.....	67
Requerimientos Técnicos.....	67
4.1. Instalación y creación de las bases de datos alfanumérica y geoespacial	67
4.1.1. Requerimientos de software.....	67
4.1.2. Requerimientos de hardware.....	67
4.1.3. Pasos para la instalación	68
4.1.4. Pasos para la creación de la base de datos con un script previamente desarrollado.....	76
4.1.5. Pasos para la creación de la base de datos desde cero	78
4.2. Instalación y configuración del Servidor de mapas	80
4.2.1. Requerimientos de software.....	80
4.2.2. Requerimientos de hardware.....	80
4.2.3. Pasos para la instalación	80
4.3. Instalación y configuración de PrimeFaces Mobile.....	82
4.3.1. Requerimientos de software.....	83
4.3.2. Requerimientos de hardware.....	83
4.3.3. JDK 1.6 update 45.....	83
4.3.4. Eclipse.....	85
4.3.5. JBoss Tools	85
4.3.6. JBoss (Application Server)	85
4.3.7. Jboss Tools 3.3.2.....	86
4.3.8. PrimeFaces y PrimeFaces Mobile.....	86

Capítulo V	88
Manual de desarrollo.....	88
5.1. Creación de un nuevo proyecto en Jboss Seam 2.3	88
5.2. Configuración de la seguridad en Jboss Seam 2.3	89
5.3. Parametrización de una búsqueda utilizando MapServer	92
5.4. Recuperación de metadatos de una Geometría	95
5.5. Agregar una geometría de puntos a una tabla en PostGIS	96
5.6. Importar archivos shape hacia PostGIS	97
CONCLUSIONES	98
RECOMENDACIONES	100
BIBLIOGRAFÍA	101
ANEXOS	103

RESUMEN

Hoy en día el avance de la tecnología pone al alcance de cualquier persona dispositivos móviles con grandes capacidades tales como posicionamiento vía GPS, toma de fotos, conexión a internet entre otros.

Las aplicaciones SIG permiten la administración y mantenimiento de información geoespacial con la desventaja de que dicha información debe primeramente ser levantada en el campo para luego ser ingresada, consumiendo una gran cantidad de tiempo y recursos de la empresa.

En el presente caso de estudio se pretende crear una aplicación SIG que permita aprovechar las capacidades de portabilidad, posicionamiento y conexión a internet que tienen los dispositivos móviles para crear así una Aplicación SIG móvil para el manejo de un Inventario de Bienes Inmuebles de la ciudad de Cuenca, todo esto utilizando la librería de componentes PrimeFaces Mobile.

ABSTRACT

Nowadays technological advances allow people to have access to mobile devices with great capacity such as GPS positioning, photograph tools, and internet connection among others.

IMS applications allow the management and storage of geospatial information. The disadvantage of this application is that the information needs to be obtained from the field first and then entered, which consumes a great amount of the company's time and energy.

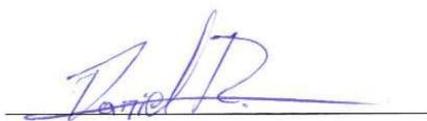
The purpose of the present study case is to develop an IMS application that allows taking advantage of the portability, positioning, and internet connection capacities of mobile devices in order to develop a Mobile IMS Application for the management of an inventory of the movable assets in the city of Cuenca, through the use of Prime Faces Mobile library components.



Translated by,
Diana Lee Rodas

RESPONSABILIDAD

El autor se responsabiliza de los criterios y conceptos vertidos en esta monografía.

A handwritten signature in blue ink, appearing to read 'Daniel R.', is written over a horizontal line.

Daniel Alberto Rodríguez López

INTRODUCCIÓN

La información geoespacial ha ido tomando vital importancia para un sin número de empresas. La toma de decisiones se facilita enormemente al tener dicha información a disposición del usuario de una forma constante, precisa y sobre todo de bajo costo. Por esta razón se han desarrollado los Sistemas de Información Geográfica (SIG).

La integración organizada de hardware, software y datos georreferenciados diseñada para capturar, almacenar, manipular, analizar y presentar información geográficamente referenciada con el fin de ayudar a la toma de decisiones dependiendo de las necesidades de su propietario se le ha denominado como SIG. Los datos geográficos que se utilizarán en la presente tesis están en el sistema de coordenadas geográficas WGS84 (World Geodetic System 1984) UTM (Universal Transverse Mercator) 32717¹(Spatial Reference 1).

El almacenamiento y la administración del SIG en muchos de los casos se realizan en equipos propios de la empresa que no permiten la portabilidad de esta actividad, es decir que no se brinda la posibilidad de actualizar, consultar o dar mantenimiento a la información directamente en el campo, que es de donde se obtienen físicamente los datos. Este es el modo en el mayoría de empresas maneja la alimentación y administración de los datos del SIG. Existen otras opciones para dar la opción de portabilidad al usuario en cuanto a almacenamiento y administración, reduciendo así la inversión en cuanto a costos y recursos.

El avance de las telecomunicaciones ha permitido publicar la información mediante internet o una intranet de forma rápida y sencilla. Además se ha llegado a tener aplicaciones capaces de almacenar y gestionar la información que contiene un SIG en tiempo real y de una manera muy segura dependiendo de la tecnología que se vaya a implementar. Para esta actividad se hace uso de recursos como servidores, bases de datos, estándares, servicios, herramientas y software de gestión de los datos.

Existen aplicaciones de que ayudan a gestionar un SIG desde un dispositivo móvil. Estas son llamadas Aplicaciones SIG móviles, las cuales pueden ser desarrolladas para ser corridas ya sea en la web o nativamente en el dispositivo. Únicamente se necesita de una conexión a red donde se encuentra el físicamente el SIG.

La popularidad de los dispositivos móviles ha crecido enormemente, y más aún cuando se tiene la posibilidad del acceso a internet desde los mismos. Además se tiene a

¹WGS84 UTM 32717: Sistema de coordenadas de la zona 17 del sur correspondiente al Ecuador bajo el estándar WGS84 UTM.

disposición del público lenguajes de programación, Frameworks², librerías de componentes, entre otras herramientas que permiten desarrollar aplicaciones que corran en dichos dispositivos móviles aprovechando todas sus bondades como son GPS³, cámara, portabilidad, etc. Un ejemplo de una librería de componentes que permite construir aplicaciones web compatibles con dispositivos móviles es PrimeFaces Mobile⁴.

Un SIG almacenado en la web y gestionado desde una aplicación compatible con dispositivos móviles puede tener un enorme potencial para la empresa pública o privada, ya que pone a disposición del usuario la capacidad de acceso a la información del SIG en cualquier momento y lugar en tiempo real, aprovechar las bondades del dispositivo móvil y muchas otras ventajas que significan un incremento muy alto de la productividad de la empresa.

Para ejemplificar el poder de una Aplicación SIG Móvil gestionada desde un dispositivo móvil se desarrollará un inventario georreferenciado de los bienes inmuebles patrimoniales de la ciudad de Cuenca, el cual permitirá la recopilación de la información de una mejor manera y directo en el campo mediante el uso de las capacidades de un dispositivo móvil. Para este desarrollo se utilizará la librería PrimeFaces Mobile.

²FrameWork: Marco de trabajo que permite el desarrollo de software sobre un lenguaje de programación.

³GPS: Sistema de Posicionamiento Global que brinda al usuario su posición geográfica.

⁴ Más información disponible en: <http://www.primefaces.org/>

Capítulo I

Sistemas de información geográfica en la web

1.1. Introducción

Para iniciar con el estudio de una aplicación SIG móvil, es necesario primero entender lo que es un SIG y sus principales características. En primera instancia de este ámbito se puede tomar en cuenta la definición que brinda la Environmental Systems Research Institute Inc (ESRI)⁵, ya que es la mayor empresa proveedora de herramientas para la gestión de Sistemas de Información Geográfica:

“Un SIG es una colección integrada de software de computadora y datos utilizados para visualizar y gestionar la información sobre lugares geográficos, analizar las relaciones espaciales, y modelar los procesos espaciales. Un SIG proporciona un marco de trabajo para la recopilación y organización de datos espaciales e información relacionada de modo que se pueda mostrar y analizar.” (ESRI, SIG 3)

Su principal característica es la capacidad de integrar datos espaciales y alfanuméricos, permitiendo así describir y proporcionar mayor información a más de la que ya se dispone de forma georreferenciada. De esta y otras características se derivan un sinnúmero de capacidades, entre las cuales se tiene como las principales las siguientes:

- Localizar objetos geográficamente y presentarlos al usuario gráficamente de modo que se disponga de su ubicación de una forma fácil, rápida y precisa.
- Agregar información descriptiva a los objetos geográficamente referenciados. Dicha información puede llegar a ser vital en la toma de decisiones. Es posible tener un claro ejemplo de esta práctica al asignar información sobre el estado de un conjunto de bienes inmuebles geográficamente referenciados, para así dividirlos por zonas y destinar los recursos necesarios para la restauración de los mismos.
- Mantener y gestionar la información que se encuentra almacenada en el SIG mediante un sinnúmero de herramientas como aplicaciones web, dispositivos destinados, aplicaciones de escritorio o compatibles con dispositivos móviles, etc.
- Creación de escenarios mediante el establecimiento de condiciones. Dichos escenarios permiten proyecciones del estado del terreno al estar sometido a ciertas combinaciones de condiciones.
- Generación y gestión de rutas para mejorar el rendimiento de los sistemas de transporte o circulación de cualquier objeto con la capacidad de moverse en el espacio.

⁵Más información disponible en: <http://www.esri.com/>

- Generación de coberturas⁶ utilizando imágenes satelitales para determinar las condiciones de la superficie terrestre. Se tiene un ejemplo de esta capacidad en el monitoreo del nivel de vegetación sana en un lugar determinado mediante herramientas de procesamiento de imágenes.
- Mostrar la información al usuario de manera gráfica brindando herramientas que permiten navegar por todas las capas temáticas⁷ del SIG de forma dinámica.
- Integración con varias tecnologías y otros SIG.

1.1.1. Tipo de información de un SIG

Un SIG puede realizar varias operaciones dependiendo del tipo de información que se haya almacenado en el mismo y de las capacidades que tenga la aplicación que lo gestiona. La información puede ser de dos tipos:

1.1.1.1. Raster

El tipo de datos raster se refiere a imágenes representadas en mallas comúnmente obtenidas mediante fotografías satelitales. Su principal objetivo es el procesamiento de las propiedades del terreno más que de una precisión en las ubicaciones de los objetos. El detalle de estas propiedades depende de la de las dimensiones de las celdas que contiene la malla. A mayor dimensión de celdas, menor precisión de la información.

Un tipo de datos raster está compuesto de filas y columnas de celdas, las mismas que almacenan un único valor. Este valor puede ser representado mediante colores (píxeles), formando así las imágenes. Mediante el procesamiento de las celdas se puede llegar a determinar valores como uso del suelo, humedad, temperatura, vegetación, contaminación atmosférica, etc.

En comparación con los datos vectorial, se requiere mayor cantidad de memoria para almacenar este tipo de datos con alta calidad y el procesamiento de los mismos es más complicado en cuanto a los procedimientos, además de que la precisión de la localización de los objetos es mucho menor que la de los datos vectoriales. Por otro lado, maneja óptimamente la alta variabilidad de los datos y ofrece mayores capacidades de procesamiento de la información. Presenta características que es imposible obtener en los tipos de datos vectoriales.

⁶ Coberturas: Analogía digital de la hoja de un mapa que forma la unidad básica de almacenamiento de datos.

⁷ Capas Temáticas: Agrupación de elementos geográficos con similares características.

1.1.1.2. Vectorial

Los datos vectoriales modelan figuras en el espacio mediante la utilización de puntos, líneas y polígonos. El principal interés de este tipo de datos es la precisión de la localización geográfica de los elementos en el espacio. Todos estos datos están relacionados con información alfanumérica para poder describir sus características.

La estructura de los datos vectoriales es más compleja que de los datos raster y presenta poca eficiencia cuando se presenta alta variabilidad, además de que es más laborioso el mantener actualizada la información. En contraste a estas deficiencias, los datos vectoriales necesitan menor cantidad de espacio de almacenamiento debido a su estructura compacta, tiene una mayor compatibilidad con las bases de datos relacionales, no pierden definición al momento de mostrarlos al usuario y en algunos aspectos ofrece mayor capacidad de procesamiento de la información, sobre todo en el procesamiento de rutas. Este tipo de datos es idóneo en cuanto a elementos del espacio que necesitan alta precisión en su localización y tener una buena descripción de sus propiedades, por ejemplo una base de datos geográfica de los predios de una ciudad.

1.1.2. Algunas aplicaciones de los SIG

- Inventario de recursos (bienes inmuebles, agrarios, forestales, hídricos, cinegéticos, humanos, etc.)
- Gestión de datos catastrales (propiedad, atributos, productividad, etc.)
- Planificación rural de construcciones.
- Gestión de equipamientos e instalaciones (electricidad, agua, teléfono, etc.)
- Gestión de infraestructuras colectivas (regadíos, caminos, etc.)
- Gestión y estudio de mercados (marketing)
- Simulación de procesos ambientales (erosión, incendios, contaminación, etc.)
- Determinación de parámetros de diseño (recursos hídricos, avenidas, altura de torres de vigilancia, etc.)
- Determinación de impactos ambientales (estudios de visibilidad) y simulación de medidas correctoras.
- Estudio de la evolución de escenarios ambientales.

1.2. Componentes de un SIG en la web

Un sistema de información geográfica no puede funcionar ni ser gestionado estando solamente de forma física en un servidor y publicado en la web. Por esta razón, es necesario un conjunto herramientas, servicios, estándares y aplicaciones que permitan al usuario interactuar con los datos del SIG y gestionarlos de la mejor manera. Dichas aplicaciones serán,este caso, ejecutadas desde un dispositivo móvil. Dentro de los principales componentes se tienen los siguientes:

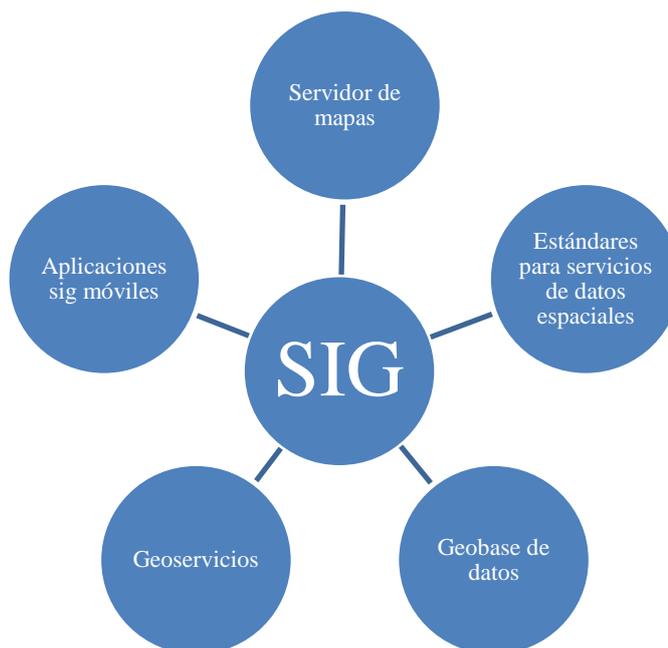


Figura 1.1: Componentes de un SIG

1.2.1. Servidor de Mapas

Un servidor de mapas es un componente de software que permite la interacción con la información geográfica, la misma que puede estar físicamente dentro de una base de datos geográfica o en archivos en el disco. Mediante peticiones al servidor el usuario tiene acceso a un sinfín de funcionalidades básicas o avanzadas dependiendo de las necesidades del mismo.

Los servidores de mapas aprovechan la arquitectura cliente-servidor para que el usuario sea capaz de consumir servicios mediante peticiones enviadas desde el lado del cliente, que en este caso es el navegador de internet. Los resultados se presentan mediante la creación de imágenes, las mismas que son enviadas al cliente para la construcción de los mapas. Los mapas son visualizados en una aplicación web mediante el uso de alguna tecnología que permita explotar los servicios del servidor de mapas.

El servidor de mapas ofrece varias funcionalidades que ayudan enormemente al desarrollo de aplicaciones SIG destinadas tanto a navegadores de equipos de escritorio o portátiles como a navegadores de dispositivos móviles. Estas funcionalidades pueden ser agrupadas principalmente en cuatro categorías:

Visualización:

Esta es la más importante de las funcionalidades debido a que es la visualización donde el usuario recupera el resultado de todas las peticiones enviadas al servidor, además de que tiene acceso a posibilidades como: selección capas cartográficas a mostrar, mapas dinámicos en los cuales se realizan cambios en la visualización mediante la interacción con el usuario, funcionalidades básicas de zoom o selección de límites de visualización, muestra de información de elementos puntuales dentro del mapa, selección de elementos para su procesamiento, entre otras.

Consulta:

El filtrado de los datos es vital para el usuario, por lo que se da la posibilidad de dos tipos de consulta dependiendo del tipo de la información que se va a filtrar:

- Alfanuméricas, donde se toman en cuenta las características descriptivas de los elementos del mapa, por ejemplo código, tipos, nombre, direcciones, estados o una fusión de varios de campos dependiendo de las necesidades del usuario.
- Geográficas, para las cuales se dispone de sentencias que facilitan el uso de funciones como búsqueda mediante proximidad, triangulaciones, entre otras.

Análisis:

El servidor de mapas, en combinación con la geo base de datos, brinda la posibilidad de implementar en la aplicación muchas de las funcionalidades de análisis que tienen herramientas de manejo de SIGs como por ejemplo Quantum GIS, gvSIG, Kosmo, etc. Algunas de estas funcionalidades pueden ser:

- Combinación de capas.
- Generación de zonas de influencia.
- Generación de rutas para la optimización de transportes.
- Visualización personalizada y filtrada de las capas y sus geometrías.
- Identificación de los atributos alfanuméricos de los elementos cartográficos.
- Integración entre bases de datos geográficas para la combinación de la información.

Mantenimiento de la información:

El mantenimiento de la información implica creación, modificación o eliminación de la misma. Esta es una de las funcionalidades más importantes desde el punto de vista de las aplicaciones SIG móviles, ya que permite la administración de los datos a más de mostrar, analizar y consultar. Este tipo de funcionalidades se implementa únicamente con datos vectoriales, ya que estos pueden ser tomados en cuenta individualmente como puntos, líneas o polígonos.

Existe otro tipo de funcionalidades adicionales que podemos obtener de un servidor de mapas como por ejemplo: impresión de los mapas según escalas seleccionadas, conexión tanto a bases de datos geográficas como a archivos de mapas, seguridades, integración con otros sistemas de información geográfica, etc.

1.2.1.1. Arquitectura de un servidor de mapas

Los servidores de mapas adoptan la arquitectura cliente-servidor, donde las peticiones al servidor son enviadas desde el lado del cliente y respondidas según sus políticas de forma una ordenada. Esta arquitectura consta de los siguientes elementos:

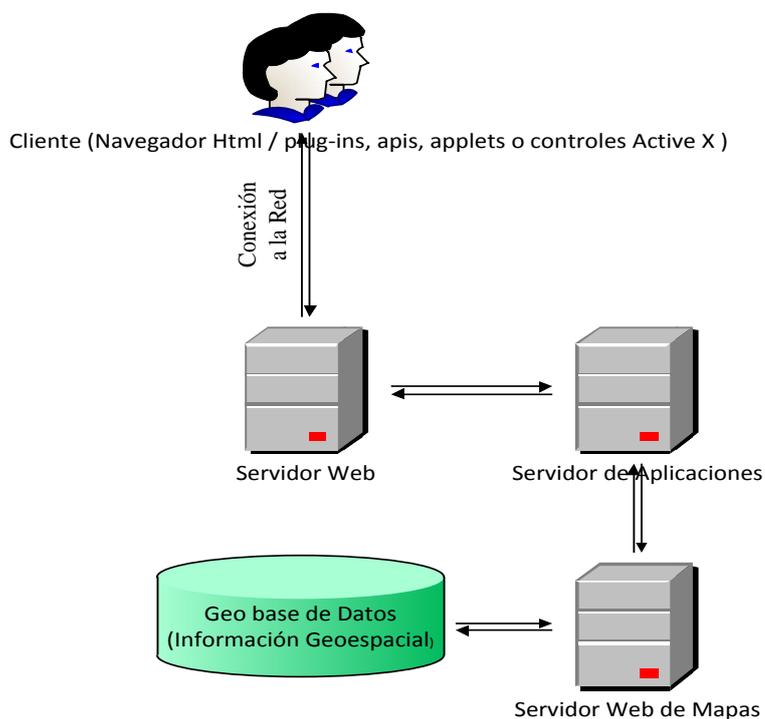


Figura 1.2: Arquitectura de un servidor de mapas

- Cliente

El cliente puede ser un computador o un dispositivo móvil que disponga de la capacidad de conectarse a la red donde se encuentra el servidor. Existen varios tipos de clientes dependiendo del punto de vista del dispositivo, del tipo de carga que tiene el mismo sobre el servidor o del formato de información que recibe.

Según el dispositivo tenemos clientes que se conectan desde un dispositivo móvil ya sea mediante el navegador web o una aplicación nativa que corra sobre el sistema operativo y clientes que se conectan desde un computador de escritorio o portátiles que se conectan desde un navegador o un programa que corra sobre el sistema operativo.

En cuanto a la carga que se provoca sobre el servidor, se tiene clientes ligeros que únicamente consumen servicios y toda la carga se refleja en el lado del servidor y clientes pesados que ejecutan varios procedimientos dentro de la aplicación que están ejecutando, necesitando así mayores recursos.

Desde el punto de vista del formato de la información que maneja el cliente tenemos el tipo imagen raster (JPEG, PNG GIF, o cualquier otro formato genérico de imagen) donde el cliente simplemente debe disponer de un navegador HTML. Pero si la información es de tipo vectorial encriptado, se debe disponer de herramientas para interpretación de esta información tales como APIs, plug-ins, applets o controles Active X.

- Una conexión a la red de internet o intranet

Este es el medio por el cual las peticiones del cliente llegan al servidor. Aparte de tener la capacidad de un acceso local en una intranet, se puede llegar a una red más amplia como es el internet.

- Un servidor web

Este es el servidor que enviará las respuestas al cliente según sea necesario, ya sea mediante páginas HTML o archivos dependiendo de la petición.

- Un servidor de aplicaciones

El servidor de aplicaciones realiza conexión entre el software de la aplicación SIG y el servidor web. Este tipo de servidor es llamado Middleware⁸.

- Un servidor web de mapas

El servidor de mapas es una puerta de enlace hacia la información geoespacial. Es aquí donde se generan los mapas que serán enviados a petición del cliente. Se deben tomar en cuenta los estándares que define el formato de las peticiones de servicios realizadas al servidor.

⁸Middleware: Software que permite la interacción entre aplicaciones, en este caso en la web.

- Información geoespacial

Es aquí donde se almacena la información de forma física. Muchos de los casos se tienen una base de datos geográfica con un Sistema Gestor de Base de Datos que brinda la facilidad de administración de los datos.

1.2.2. Geo Base de Datos

Es la geo base de datos donde residen los datos físicamente. El Sistema Gestor de Base de Datos (SGBD) permite el almacenamiento y gestión de la información de una forma segura y sencilla. Según la ESRI, se la puede definir como:

“Una base de datos o estructura de archivos usada principalmente para almacenar, consultar y manipular los datos espaciales. La Geo Base de Datos almacena geometrías, un sistema de referencia espacial, atributos y reglas de comportamiento para los datos. Varios tipos de conjuntos de datos geográficos pueden ser recogidos dentro de una Geo Base de Datos incluyendo características de las clases, tablas de atributos, conjuntos de datos raster, conjuntos de datos de red, topologías y muchos otros”. (ESRI, GeoDatabase 10)

Dentro de la Geo Base de Datos, la información espacial es tomada como un nuevo tipo de datos. Por lo general se dispone de información geométrica o topológica, donde se brinda la posibilidad de geo referenciar objetos en el espacio. Además se dispone del modelo geo-relacional, el mismo que fusiona los datos geo espaciales con los alfanuméricos para ofrecer así al usuario todas las capacidades de ambos tipos de datos.

La principal ventaja del manejo de una geo base de datos es que se dispone de un SGBD. Las bondades de dicho gestor son enormes en cuanto a seguridades, posibilidades de consulta de la información geo espacial mediante SQL⁹, facilidad del manejo de concurrencia hacia los datos y muchas más.

Las peticiones enviadas al servidor de mapas son procesadas en la geo base de datos. Por ejemplo cuando se realiza una petición para un ingreso de un elemento en una capa de puntos, el SGBD es quien realiza internamente un ingreso en la base de datos según los datos enviados por el servidor de mapas.

Al ser tomada a la información geo espacial como un tipo de datos dentro de las tablas, las aplicaciones SIG móviles pueden interactuar directamente con el SGBD para realizar consultas y acciones de creación, modificación y eliminación. Todas estas peticiones

⁹SQL: Lenguaje de consulta estructurado que permite obtener datos desde la base mediante sentencias.

se realizan mediante SQL dependiendo de la conexión que la aplicación tenga con la base de datos.

1.2.3. Estándares para servicios de datos espaciales

Para garantizar la integración entre sistemas al momento de manejar datos espaciales, han surgido organizaciones que proveen estándares para el manejo de geoservicios. El manejo de dichos estándares asegura la compatibilidad e interoperabilidad necesaria para que los datos y servicios puedan ser utilizados, combinados y compartidos entre distintos sistemas sin estar sujetos a productos o licencias determinadas, reduciendo así el costo de compartir información geográfica a través de medios digitales.

1.2.3.1. Organizaciones proveedoras de estándares

Existe un sinnúmero de organizaciones que han desarrollado estándares para el manejo de datos espaciales. Las mismas que están soportadas por las principales geo bases de datos y geoservicios del mercado. Entre las principales y más utilizadas están dos encabezando la lista: OGC (Open Geospatial Consortium) e ISO (International Organization for Standardization).

1.2.3.1.1. OGC (Open Geospatial Consortium)



OGC es un consorcio creado en 1994 que agrupa 372 organizaciones comerciales, gubernamentales, sin fines de lucro y de investigación en todo el mundo. El consorcio fue fundado a partir del software libre GRASS perteneciente a la OGF (Open GIS Foundation) fundada en 1992. Su misión es la de fomentar el desarrollo y la implementación de estándares abiertos para el procesamiento e intercambio de datos geoespaciales y servicios.

Existen a la fecha varias especificaciones entre las cuales las principales son:

- WMS (Web Map Service)
- WFS (Web Feature Service)
- WCS (Web Coverage Service)
- WPS (Web Processing Service)
- WFS-T (Web Feature Service Transactional)

1.2.3.1.2. ISO (International Organización for Standarization)



La ISO u Organización Internacional para la Estandarización inicia sus operaciones de manera oficial el 23 de febrero de 1947. Actualmente es una red de institutos de normas nacionales de 157 países cuya misión es la de garantizar que los productos y servicios sean seguros, fiables y de buena calidad. Todo esto mediante la adopción de normar y estándares.

Las principales normas ISO que hacen referencia al manejo de información geográfica, metadatos, procesamiento datos espaciales y servicios web son las siguientes:

- ISO 19115:2003 información geográfica – Metadatos.
- ISO19110:2005 información geográfica – Metodología para la catalogación de objetos.
- ISO 19106:2004 Información geográfica – Perfiles.
- ISO 19119:2005 Información geográfica – Servicios.
- ISO19111:2007 Información geográfica – Referencia espacial por coordenadas.
- ISO 19112:2003 Información geográfica – Referencia espacial.
- ISO 19113:2002 Información geográfica – Principios de calidad.
- ISO 19131:2007 Información geográfica – Especificaciones de productos de datos.
- ISO 19135:2005 Información geográfica – Procedimientos para el registro de elementos.
- ISO/TS 19127:2005 Información geográfica – Códigos geodésicos y parámetros.
- ISO/TS 19138:2006 Información geográfica – Medidas de calidad de datos.
- ISO/TS 19139:2007 Información geográfica – Metadatos – La aplicación del esquema XML.

Estas normas ISO hacen énfasis en la calidad y mejores prácticas al momento de realizar la captura coherente y el mejor uso de los metadatos geoespaciales. Proporciona una forma de descripción y representación de los datos estandarizada para garantizar así la interoperabilidad y comunicación entre sistemas.

En Ecuador, según el Perfil Ecuatoriano de Metadatos (PEM), se ha acogido la norma ISO 19115:2003 información geográfica para la normalización en cuanto al levantamiento de metadatos. El principal objetivo de dicha norma es:

“Establecer las especificaciones técnicas mínimas que deben cumplir las instituciones públicas o privadas que generan información espacial, para la construcción, edición y revisión de metadatos, de tal manera que puedan otorgar funcionalidad con la búsqueda, localización, acceso, uso, distribución y transferencia de datos”(PEM 13)

1.2.4. Geoservicios

Para el presente caso de estudio se ha tomado en cuenta a OCG como proveedor de estándares para geoservicios. OCG ha definido a OWS (Open Geospatial Consortium Web Services) como estándar de comunicación de datos y servicios geoespaciales en la web. Entre los principales servicios que provee este estándar están los siguientes:

1.2.4.1. WMS (Web Map Service)

Brinda la capacidad de interactuar gráficamente con el cliente mediante la producción de mapas a partir de datos georreferenciados, los mismos que pueden ser obtenidos desde una geo base de datos. Presenta la información por medio de imágenes digitales en formatos como PNG, GIF, JPG o como gráficos vectoriales en formato SVG (Scalable Vector Graphics) o Web CGM (Web Computer Graphics Metafile). Mediante este servicio no se acceden a los datos mismos, sino a una representación gráfica de ellos.

Al acceder a las diferentes capas del SIG se tiene la opción de mostrarlos sobrepuestos uno sobre otro y crear una vista personalizada de la información geográfica. El desarrollador puede construir dicha vista estableciendo la forma en la que la información será visualizada en el lado del cliente a partir de un archivo de configuración o de un asistente de configuración de configuración dependiendo del servidor de mapas.

Dentro de este estándar se tiene a disposición del desarrollador tres operaciones distintas:

1. Desarrollar metadatos del nivel servicio.
2. Desarrollar un mapa cuyos parámetros geográficos y dimensionales han sido bien definidos.
3. Desarrollar información de características particulares mostradas en el mapa (opcionales).

Generalmente las operaciones WMS son invocadas en el navegador mediante peticiones enviadas a través de la URL (Uniform Resource Locator). Mediante la URL es posible indicar la información que será mostrada en el mapa a petición del usuario. Se puede establecer parámetros como la porción del mapa que será mostrada, el sistema de coordenadas a utilizar, el ancho y el alto de la imagen a mostrar, entre otros. Existe la opción de solapamiento de mapas de iguales características, además de poder obtener dichos mapas de distintos servidores.

Al permitir la obtención de mapas de diversos servidores, es posible la creación de una red distribuida de servidores, a partir de la cual el cliente puede generar sus mapas a medida únicamente el envío de peticiones en la URL.

Cuando se implementa este servicio se tiene a disposición varias operaciones dependiendo de las capacidades del servidor de mapas que se esté utilizando. Entre las principales operaciones tenemos las siguientes:

- Exceptions: Si una excepción ha ocurrido.
- GetCapabilities: Recupera los metadatos del servicio, incluyendo las operaciones y parámetros compatibles, y una lista de las capas disponibles.
- GetMap: Recupera una imagen de mapa de una área y contenido especificados.
- GetFeatureInfo: Recupera los datos subyacentes, incluyendo valores de geometría y atributos para una ubicación de un píxel en el mapa.
- DescribeLayer: Indica el WCS o WFS para recuperar información adicional acerca de la capa.
- GetLegendGraphic: Recupera una leyenda generada para el mapa.

1.2.4.2. WFS(Web FeatureService)

Ofrece una interfaz de comunicación que brinda la posibilidad de superponer capas de distintos servidores con la ventaja de acceder directamente a información misma y no solo a imágenes generadas por el servidor como en WMS. Además permite interactuar con los mapas obtenidos mediante el estándar WMS, como por ejemplo, editar la imagen o analizar la imagen siguiendo criterios geográficos. WFS recupera y analiza datos codificados en GML¹⁰ (Geography Markup Language) que deriva de XML¹¹ (eXtensible Markup Language), el mismo que es el lenguaje estándar para el envío de órdenes WFS.

¹⁰GML: Es una herramienta XML de la OGC para la descripción de datos espaciales de forma vectorial. No brinda ningún tipo de información acerca de la presentación de los datos al cliente.

¹¹XML: Lenguaje de marcas extensible. Mediante este lenguaje se realiza el intercambio de datos entre aplicaciones, ya que es un lenguaje estándar.

Al ser un servicio no transaccional, únicamente permite hacer consultas y recuperar elementos geográficos, mas no así realiza acciones de creación, eliminación o actualización de los elementos gráficos del mapa.

1.2.4.3. WCS(Web CoverageService)

Este servicio permite el intercambio de información en forma de coberturas en la web. Dichas coberturas pueden ser imágenes raster, satelitales o una matriz digital de elevación del terreno. WCS devuelve datos en un formato que se puede utilizar como entrada para análisis y modelado a diferencia de WMS que únicamente devuelve una imagen de los datos.

Para obtener una cobertura, es necesario conocer el punto de ingreso de un servidor de coberturas, por ejemplo: <http://mapas.topografia.upm.es/cgi-bin/v46/WCS>. Luego de conocer dicho punto, se tiene acceso a las siguientes operaciones:

- GetCapabilities: Obtener los metadatos del servicio y de las coberturas que ofrece.
- DescribeCoverage: Describir de forma detallada una o varias coberturas.
- GetCoverage: Obtener una cobertura o parte de ella.

1.2.4.4. WPS(Web ProcessingService)

Dentro de este servicio se establecen normas para la estandarización de las entradas y salidas (solicitudes y respuestas) para servicios geoespaciales de procesamiento. Un ejemplo de este tipo de servicios es el de superposición de polígonos. Además se brinda la capacidad de definir como un cliente puede solicitar la ejecución de un proceso, y la forma en la que se manejará la salida de dicho proceso.

Los estándares definen una interfaz que facilita la publicación de los procedimientos geoespaciales y el descubrimiento de los clientes y de la unión de los procesos. Todos los datos requeridos por este servicio pueden ser distribuidos a través de una red o tan solo estar disponibles en el servidor.

En sí el servicio describe como invocar procesamientos distribuidos mediante el protocolo HTTP, adoptando un comportamiento similar al de un Web Service¹² con la única particularidad de que está orientado a procedimientos geoespaciales.

¹² Web Service: Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

1.2.4.5. WFS-T(Web Feature Service Transactional)

Es una extensión del servicio WFS que a más de permitir la consulta de información geoespacial, brinda la capacidad de realizar transacciones de creación, modificación y eliminación de elementos geográficos directamente en el mapa mediante el uso de un cliente GIS. Debido a su lógica de Web Service, permite la interoperabilidad entre servicios y aplicaciones.

Al ser un servicio que permite transaccionalidad, brinda la capacidad de control de concurrencia, es decir que varios usuarios pueden realizar transacciones al mismo tiempo. Gracias a su acceso compartido y estandarizado a la geo base de datos, permite el mantenimiento y visualización en conjunto.

Este servicio brinda la posibilidad de acceso a los datos a través de un entorno SIG, permitiendo así al usuario la consulta y edición de la información de la geo base de datos de una forma fácil y segura, que es lo más importante. La interoperabilidad es otra de las capacidades que se consigue con este estándar, ya que las transacciones se pueden realizar directamente o traducirla al lenguaje del repositorio de datos con el que se conecta, para posteriormente realizar la transacción dentro de la geo base de datos.

Las transacciones posibles dentro de este geoservicio, a más de la de consulta, son las siguientes:

Insert

Al utilizar la lógica de un Web Service, se usa el lenguaje GML para especificar las propiedades de la geometría¹³. La sentencia insert permite el ingreso de múltiples elementos en la base de datos en una sola transacción.

Update

Dentro de la modificación de las geometrías, es necesario especificar la propiedad que se pretende modificar, ya que a más de las propiedades alfanuméricas, se puede modificar las propiedades geográficas de los elementos. Por último se debe especificar el nuevo valor de la propiedad, todo esto en lenguaje GML.

¹³ Geometría: Elemento del plano almacenado en una geo base de datos o archivos de un SIG.

Delete

Se deben especificar el elemento conjunto de elementos a ser eliminados. Se puede realizar una consulta de estos elementos para poder seleccionarlos y eliminarlos en base a un filtrado utilizando el comando Filter. También es posible bloquear la eliminación de ciertos elementos.

1.2.5. Aplicaciones SIG móviles

Las empresas están acostumbradas a realizar un doble trabajo en cuanto a la alimentación de la información de un sistema de información geográfica, ya que muchas de las veces recolectan la información en el campo y posteriormente la ingresan a su SIG. Esta información en muchos de los casos requiere ser verificada directamente en el terreno debido a la evasión de muchos detalles importantes, por lo que provoca un triple trabajo a más del ya realizado. Además al transportar la información de un lugar a otro para cargarla en el sistema, esta corre riesgos como distorsión al ingresarla, pérdidas al momento de transportarla, demasiados recursos utilizados en la recolección y muchos más.

El crecimiento de la popularidad de los dispositivos móviles y sus capacidades ha sido enorme en los últimos tiempos. Su tecnología permite funcionalidades como conexión a internet o intranet ya sea mediante acceso Wi-Fi¹⁴ o 3g¹⁵, además de capacidades como GPS (para el geo posicionamiento instantáneo), cámara, etc. Este tipo de funcionalidades brinda al usuario la facilidad de realizar tareas directamente en el terreno y mejor aún interactuar directamente con SIG desde cualquier lugar donde requiera hacerlo el usuario, resolviendo así todos los problemas de trabajo innecesario y dificultad del manejo de información que tiene una empresa. Para esta interacción han surgido las “Aplicaciones SIG móviles”.

Según el tipo de interacción que tenga la aplicación SIG móvil con la información, existen tres tipos:

- Sistemas de visualización y navegación de mapa: soluciones de baja precisión y bajo coste que permiten a la aplicación móvil utilizar el mapa en el campo. Esto a menudo incluye la capacidad de capturar o bosquejar datos de mapa no estructurados (información alfanumérica).
- Sistemas de recopilación y mantenimiento de datos: soluciones destinadas a la recopilación y mantenimiento de datos del campo que garantizan la precisión de las capas de datos SIG.

¹⁴Wi-Fi: Hardware que permite la conexión inalámbrica de dispositivos.

¹⁵Red 3g: Red telefónica de tercera generación para la transferencia de voz y datos.

- Sistemas topográficos: soluciones más complejas para la recopilación de datos topográficos de campo. La topografía¹⁶ tradicionalmente no se ha considerado una parte común de los flujos de trabajo SIG; sin embargo, esto cambia rápidamente.

Según la forma en la que se corre en el dispositivo, una aplicación SIG móvil puede ser de dos tipos:

- Aplicación nativa, que es instalada directamente en el sistema operativo del dispositivo y se corre sobre el mismo.
- Aplicación web, que se corre sobre el navegador y realiza una renderización de la página de tal manera que sea compatible con el tamaño de la pantalla del dispositivo. Para su desarrollo existen herramientas como PrimeFaces Mobile, una librería de componentes que se implementa sobre Java Server Faces 2.0 (JSF 2,0)¹⁷ y permite la compatibilidad de la aplicación web con el dispositivo móvil.

¹⁶ Topografía: Representación gráfica de la superficie terrestre.

¹⁷JSF: Implementación que permite el desarrollo de páginas web en lenguaje JAVA.

Capítulo II

PrimeFaces Mobile

2.1. Introducción

PrimeFaces es una librería de componentes de código abierto que se implementa sobre JSF (Java Server Faces), por lo cual permite la integración con varias tecnologías como RichFaces, IceFaces y muchas más. Fue creada por la empresa turca Prime Technology. Se encuentra bajo la licencia Apache License V2.

Existen varias versiones de PrimeFaces, las mismas que son compatibles con las diferentes versiones de JSF. Esta es una capacidad muy importante, ya que las versiones de JSF anteriores no permiten ciertas mejoras en funcionalidad, performance y facilidad de uso desde el punto de vista del desarrollador. La compatibilidad es una de las principales ventajas de esta tecnología.

PrimeFaces Mobile es una extensión de la librería PrimeFaces con un kit de componentes destinados para el desarrollo de aplicaciones web renderizadas¹⁸ y compatibles con la interfaz del navegador de un dispositivo móvil, para lo cual utiliza JQuery internamente. Además permite la implementación de AJAX¹⁹ (Asynchronous JavaScript And XML) dentro de la aplicación web para aumentar la velocidad y optimizar los recursos al momento de la interacción con el usuario.

Gracias a la integración entre PrimeFaces Mobile y JSF, se tiene acceso a un sinnúmero de capacidades y tecnologías como Frameworks de persistencias, compatibilidad con una gran variedad APIs, conexiones a bases de datos alfanuméricas y geográficas, todas las seguridades que ofrece el MVC (Modelo Vista Controlador), entre otras.

¹⁸ Renderizar: Recargar una página web y mostrarla según la una resolución de pantalla seleccionada o por defecto.

¹⁹ AJAX: Técnica de desarrollo que permite renderizar pequeños elementos o partes de una página web, ahorrando así recursos de procesamiento.

2.2. Arquitectura

PrimeFaces Mobile tiene la misma arquitectura genérica que de cualquier librería que corre sobre JSF y Java. Dicha arquitectura consta de las siguientes partes:

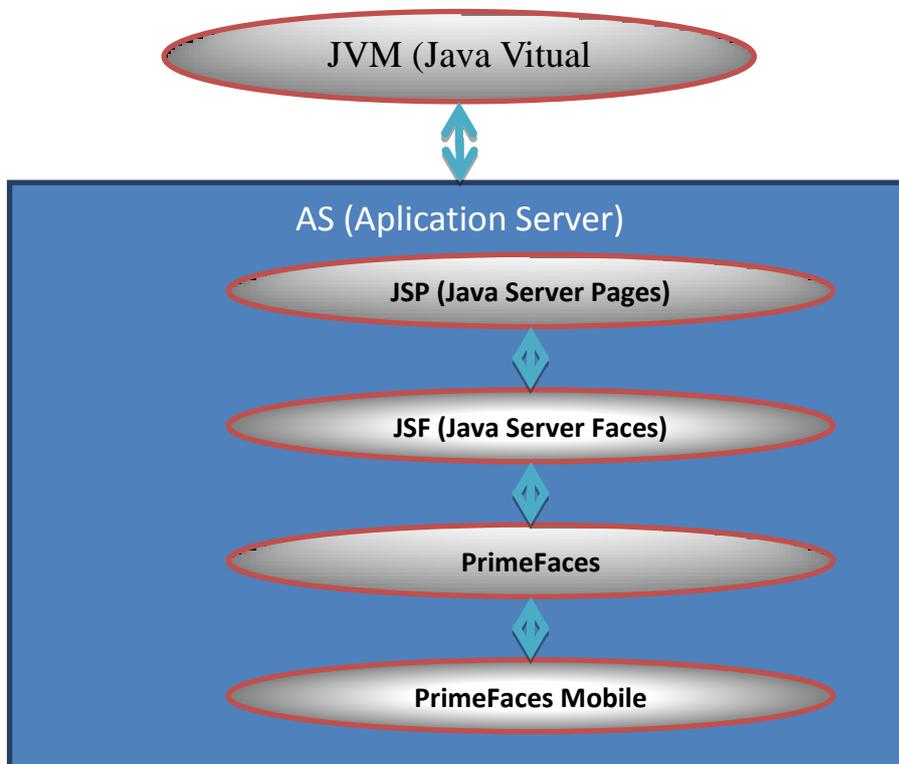
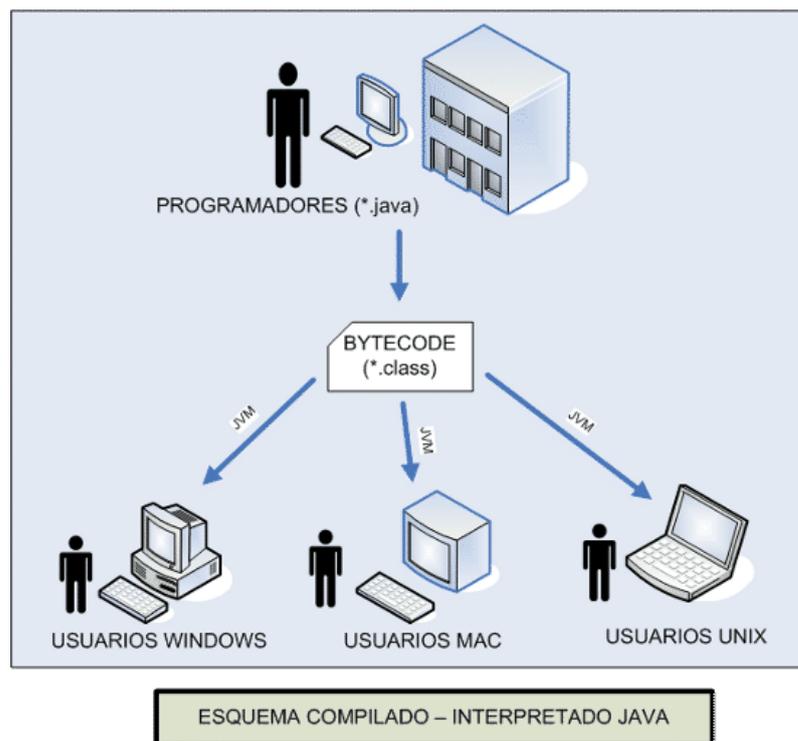


Figura 2.3: Arquitectura de PrimeFaces Mobile

2.2.1. Java Virtual Machine

Es una máquina virtual que se ejecuta directamente sobre una plataforma específica y tiene la capacidad de interpretar y ejecutar instrucciones Java generadas por un compilador. Actúa como intermediario entre la plataforma y las instrucciones, logrando así que una aplicación desarrollada con el lenguaje Java pueda ser ejecutada en cualquier sistema operativo, tan solo es necesario disponer de la máquina virtual adecuada. De ahí es de donde viene la frase “escríbelo una vez, ejecútalo en cualquier parte”.

Las instrucciones de java vienen en un código llamado Bytecode, que es el intermedio entre el código con el que el desarrollador realiza el programa (código fuente) y el lenguaje de máquina. Java toma el código fuente (.java) y lo transforma en Bytecode (.class). Posteriormente JVM traduce este código intermedio a lenguaje de máquina para su ejecución.



Fuente: (Tutoriales Aprende a Programar 29)

Figura 2.4: Esquema compilado – interpretado JAVA

2.2.2. Application Server

Un servidor de aplicaciones es un software que permite el despliegue y ejecución sobre sí mismo de aplicaciones que serán accedidas desde el lado del cliente. Entre los servidores de aplicaciones libres más populares se encuentran JOnAS del consorcio ObjectWeb, JBoss AS de JBoss (división de Red Hat), Geronimo de Apache, TomEE de Apache, Resin Java Application Server de Caucho Technology, Blazix de Desiderata Software, Enhydra Server de Enhydra.org y GlassFish de Oracle.

El servidor de aplicaciones sigue un conjunto de estándares brindando la capacidad de funcionar como contenedor de los componentes que conforman las aplicaciones y permitiendo así la integración entre las mismas. Dichos componentes se conocen con el nombre de Servlets²⁰, Java Server Pages (JSPs)²¹, controladores y Enterprise JavaBeans (EJB)²².

²⁰ Servlets: Pequeños programas java que se ejecutan dentro del contexto del navegador.

²¹ JSP: Tecnología que permite el desarrollo de páginas web basadas en HTML, XML, etc. Utiliza java como lenguaje de programación.

²² EJB: Permiten al desarrollador modelar la lógica del negocio, generando así componentes flexibles y reutilizables en el lado del servidor.

2.2.3. Java Server Pages

Es una tecnología que utiliza el lenguaje java para desarrollo de páginas web dinámicas basadas en HTTP, XML, etc. JSP implementa servlets para poder ser desplegado, por lo que es necesario un servidor de aplicaciones compatible con contenedores servlets. A diferencia de los servlets que utilizan únicamente código java, JSP utiliza HTML y java combinados.

Al estar en lenguaje java, JSP es compilado dinámicamente por el JVM a lenguaje de máquina mejorando el desempeño y eficiencia de la aplicación en comparación con aplicaciones puramente interpretadas. Además al ser compilado por el JVM tiene la ventaja de portabilidad y funcionalidades del lenguaje java.

Su sintaxis agrega acciones JSP, las mismas que son etiquetas XML adicionales que permiten invocar funcionalidades incorporadas. Permite la creación de bibliotecas de etiquetas para ser implementadas sobre la página web a más de las ya existentes en HTML y XML. Existen varias de estas bibliotecas ya desarrolladas disponibles de forma gratuita.

2.2.4. Java Server Faces

Aparte de ser una tecnología también es un framework en lenguaje java para el desarrollo de aplicaciones web. Su principal característica es que simplifica el desarrollo de interfaces de usuario tratando de crear un entorno de desarrollo similar al que se tiene en la creación de una aplicación de escritorio en lenguaje java utilizando sus librerías de componentes.

Inicialmente las aplicaciones web basadas en lenguaje java eran desarrolladas mediante JSP. JSF está implementado sobre JSP, lo cual, a más de un entorno de desarrollo de interfaz poderoso, le brinda todas las capacidades del lenguaje java como Frameworks de persistencias, lógica de negocio, conexiones a bases de datos, seguridades y muchas más.

JSF traduce las acciones realizadas desde el lado del cliente (página HTML) en eventos enviados al servidor, brindando así la capacidad de regenerar la página web de forma dinámica según los requerimientos que se hayan definido para la aplicación web. Esto permite un desarrollo más apegado a las necesidades del usuario.

Más adelante en el presente capítulo se profundizará más sobre JSF en cuanto a su funcionamiento, características, arquitectura y más importante el Modelo Vista Controlador (MVC), que es la lógica que tiene JSF al momento de su implementación.

2.2.5. PrimeFaces

Dentro de JSF se implementa PrimeFaces con una extensa librería de componentes de muy fácil utilización. Esta librería corre sobre JSF extendiendo las etiquetas que vienen por defecto. Para utilizarla dentro de una página JSF se debe únicamente agregar dentro del XML Name Space la siguiente URL `xmlns:p="http://primefaces.org/ui"` y todos los componentes serán automáticamente accesibles.

Todos los componentes de PrimeFaces son utilizados como si fueran etiquetas de JSF, su única diferencia es que generalmente tienen el prefijo `<p:` seguido del nombre del componente. Al igual que JSF, PrimeFaces se conectan con los beans²³ y puede utilizar todo su potencial al momento de conectarse con la lógica del negocio.

2.2.6. PrimeFaces Mobile

PrimeFaces Mobile es una extensión de la librería PrimeFaces con la capacidad de crear páginas web que son renderizadas para la resolución de la pantalla de un dispositivo móvil. Estas páginas web son accedidas por el navegador del dispositivo como si fuera cualquier otra página. Esta extensión utiliza JQuery²⁴ para dar al usuario la sensación de que en el navegar se tiene una aplicación propia de un dispositivo móvil.

Al igual que PrimeFaces, también se tiene que agregar una URL al XML Name Space una, la misma que es `xmlns:pm="http://primefaces.org/mobile"` y generalmente el prefijo `<pm:` seguido del nombre del componente.

2.3. Funcionamiento

Al estudiar la arquitectura de PrimeFaces Mobile, se tiene una idea de su funcionamiento como una librería de componentes, pero existen cinco tecnologías principales y necesarias para su funcionamiento: JSF, que es donde se implementa la librería; Document Object Model DOM, que guarda un árbol con todos los componentes de la página web actualmente cargada; Javascript, un lenguaje de programación para páginas HTML; Ajax para la renderización de ciertas zonas de la página web para mejorar la eficiencia de la aplicación; JQuery, que permite la visualización y diseño de interfaz.

²³ Bean: Objeto java capaz de ser accedido y utilizado por una página web que contiene una identificación, propiedades y un conjunto de procedimientos. Es la conexión entre la vista y la lógica de negocio.

²⁴ JQuery: Es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web.

2.3.1. Java Server Faces (JSF)

JSF es una tecnología y framework que permite el desarrollo de aplicaciones web basadas en lenguaje java. Su principal objetivo es el simplificar el diseño de interfaz. Se basa en la forma en la que se desarrollan las aplicaciones java que corren nativamente en un sistema operativo para crear una experiencia de diseño mucho más amigable y dinámica dependiendo de los requerimiento planteados para la aplicación que se quiera desarrollar.

Contiene un sinnúmero de componentes prediseñados como botones, controles Ajax, contenedores, tablas, bucles que permiten la generación dinámica de una página web en tiempo real, entre otros. Además permite la adición de librerías que extienden aún más la lista de componentes disponible para el usuario.

Otra funcionalidad importante de JSF es que permite la captura de eventos de una manera muy fácil y ejecución de procedimientos sobre la aplicación. Permite también la ejecución de procedimientos java mediante la conexión con los Beans, disponiendo así de todo el poder de este lenguaje y del Modelo Vista Controlador (MVC), que es el paradigma de programación sobre el que está basado JSF.

Al desarrollar los controles JSF se pensó en que cada uno tuviera su propia forma de funcionamiento, características y forma de interacción con la información, para así tener una gama ilimitada de posibilidades de desarrollo. Por esta razón ha sido posible desarrollar extensas librerías de componentes muy potentes como las de PrimeFaces, RichFaces, IceFaces y muchas más.

2.3.1.1. Modelo Vista Controlador (MVC)

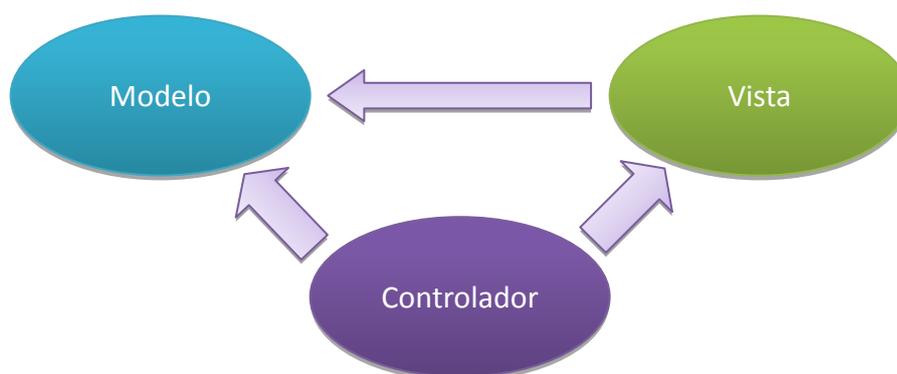


Figura 2.5: MVC

Es un paradigma de programación que propone una arquitectura de software que separa la lógica de control, la lógica de negocio y la interfaz de usuario, permitiendo la gestión y comunicación entre todas las partes de la aplicación. Para lograrlo, se proponen tres componentes distintos que son el modelo, la vista y el controlador.

Existe un sinnúmero de ventajas al utilizar MVC, como por ejemplo la organización de la estructura de la aplicación, la reutilización de código, componentes y módulos realizados, facilidad de programación y modularización, integración con tecnologías similares, facilidad de mantenimiento, estandarizar el desarrollo y muchas más.

2.3.1.1.1. Modelo

El modelo es el objeto que representa, gestiona y controla los datos de la aplicación, los mismos que son generalmente obtenidos desde una base de datos. Tiene total independencia de la vista y el controlador sin tener ni siquiera una referencia hacia ellos. El sistema en conjunto es el encargado del acceso hacia el modelo y reflejar cambios sobre el mismo dependiendo de los requerimientos de la aplicación.

Desde el punto de vista de flujo de datos hacia el usuario, el modelo se comporta como una entrada de datos de la aplicación, ya que las demás partes únicamente hacen peticiones al modelo, el mismo que hace un acceso a los datos y los recupera y modifica de forma segura, y al tener independencia con las demás partes, los datos son mucho más dinámicos y flexibles.

2.3.1.1.2. Vista

Es la forma en la que se presentan al usuario los datos obtenidos desde el modelo. La vista es una página web dinámica que puede ser accedida desde un navegador, donde el usuario puede interactuar con la misma. La forma en la que interacciona con el modelo es mediante referencias al mismo, por ejemplo: `<h:outputText value="#{usr.primerNombre}"`.

2.3.1.1.3. Controlador

En el controlador es donde se gestiona la interacción entre el usuario y la aplicación, además de la lógica de negocio. Cuando el usuario dispara un evento sobre la vista, el controlador es quién gestiona las acciones que se van a realizar sobre el modelo. Lógica de negocio es donde se dice a la aplicación que hacer frente a las acciones del usuario comunicándose con la vista para modificarla de forma dinámica, por lo que el controlador hace de intermediario entre las acciones de la vista y el modelo.

2.3.1.2. Arquitectura JSF según MVC

La arquitectura de una aplicación JSF está basada en el paradigma Modelo Vista Controlador, por esta razón, es mejor agrupar los componentes de dicha aplicación dentro de este paradigma. Además, al estar en lenguaje java corre sobre un servidor de aplicaciones y se vale de la JVM para su ejecución. La arquitectura es la siguiente:

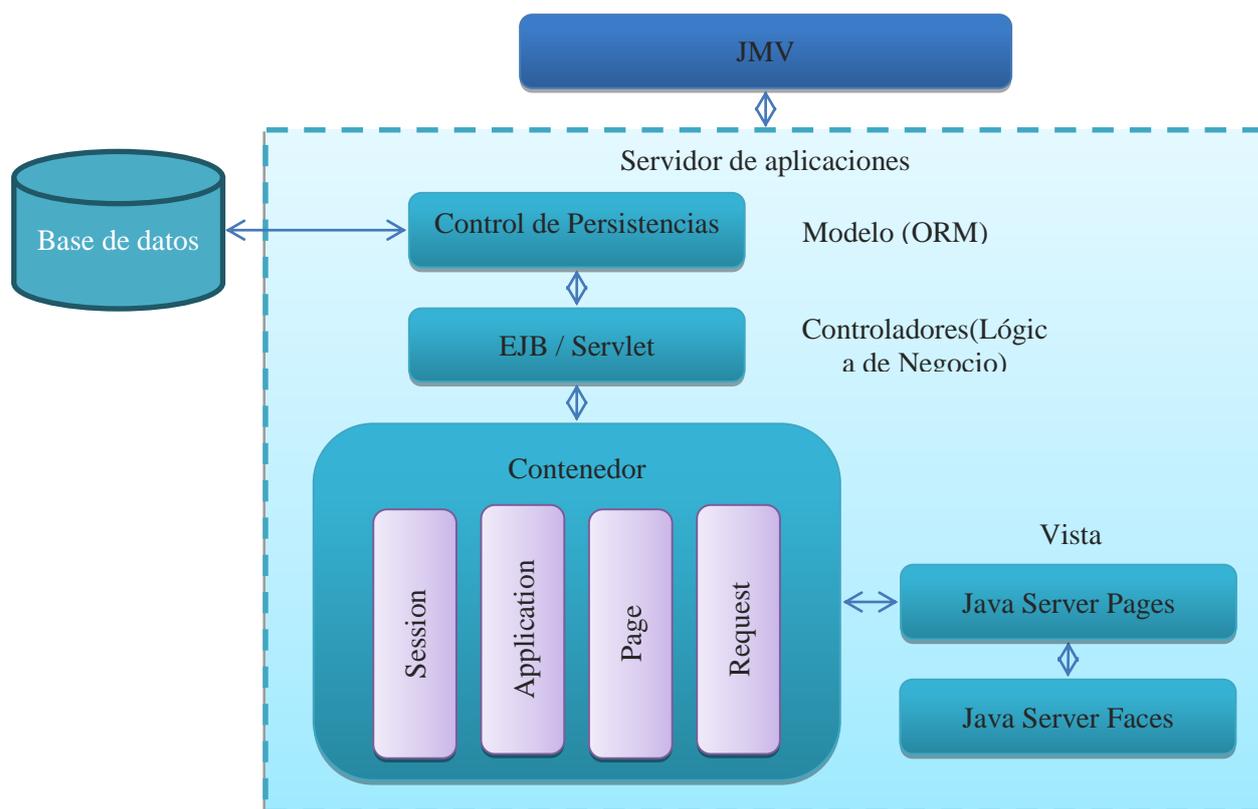


Figura 2.6: Arquitectura JSF

Desde el punto de vista del flujo de los datos en la aplicación, cada uno de los componentes de una aplicación JSF cumple una función específica. En primer lugar para el acceso a los datos en la capa de modelos, se tiene lo que se conoce como un ORM (Object Relational Mapping), el mismo que genera una abstracción de la base de datos en objetos y estos a su vez son manejados por el control de persistencias. El ORM más utilizado en conjunto con JSF es Hibernate.

En la parte del controlador se tiene los EJB (Enterprise Java Beans), que conforman la lógica de negocio y todos los procedimientos y acciones que serán realizados sobre el modelo. Además en el controlador están los servlets, los mismos que detectan las interacciones del usuario con la vista a modo de eventos. Todos estos elementos se encapsulan dentro de los diferentes tipos de contenedores, los mismos que tienen un alcance definido. Entre los contenedores más genéricos tenemos los siguientes:

Contenedor	Alcance o vigencia del contenedor
Session	Mientras una sesión esté activa
Application	Mientras el servidor de aplicación esté corriendo
Page	Mientras una página permanezca abierta
Request	Mientras dure una solicitud de un usuario

Tabla 2.1: Contenedores JAVA

En cuanto a la vista, las páginas JSF en interacción con JSP muestran al usuario los datos obtenidos por el modelo mediante referencias al mismo y son manipulados dinámicamente según la lógica de negocio.

2.3.1.3. Ciclo de vida de una aplicación JSF

1. Renderizar la vista

Se localiza la vista con la URL ingresada. Esta acción la realiza el Faces Servlet. Si es la primera vez que va a ser generada, se genera el árbol de componentes (UIViewRoot) para dicha vista a partir de su fichero de generación, caso contrario se carga el árbol correspondiente que debió haber sido generado previamente.

2. Actualizar valores de la petición HTTP²⁵

Una petición HTTP puede llevar consigo valores de entrada como campos, parámetros, cookies, campos de cabecera, etc. En este paso se actualizan los nodos del árbol de componentes a partir de dichos valores. Únicamente se actualiza la vista, en cambio la parte de la lógica de negocio vinculada a ella permanece intacta. Se realizan las siguientes acciones:

- Los beans permanecen sin cambios, mientras que la vista actualiza el estado de sus componentes.
- Actúan los Converters para realizar los cambios entre String y Object necesarios.
- Al haber errores en el renderizado de la página, se detiene el proceso y dependiendo del error se muestra un mensaje en la misma página en controles destinados para esta acción o se carga una página sobre la actual especificando los detalles del error.

3. Procesar las validaciones

Al realizar un recorrido recursivo del árbol de componentes de la vista actual, JSF verifica la validez del contenido de los mismos. Gracias a esto, es posible realizar una validación en la vista, en la parte de lógica de negocio y también en la parte del modelo mediante la implantación de un sinnúmero de métodos como controles de la vista, anotaciones dentro del modelo y los beans, procedimientos en la lógica de negocio, expresiones regulares, etc.

Además JSF automáticamente puede asociar cada control con un componente message para especificar el lugar y el motivo del error de validación, además de la

²⁵ Petición HTTP: Es la petición que se realiza al ingresar una URL.

posibilidad de usar los mismos componentes message pero de forma general para la especificación de errores en procedimientos de los beans asociados a la vista.

4. Actualizar valores de los beans

Los componentes de la vista están vinculados a los atributos y procedimientos de los beans mediante EL (Expresion Language) como por ejemplo `value="#{usr.primerNombre}"`. Esto hace que los cambios realizados en la vista sean reflejados sobre el bean de forma directa. Al actualizar los valores en las vistas se disparan los validadores antes de reflejar los cambios en el bean para evitar errores.

5. Invocar la lógica de aplicación

JSF brinda la posibilidad de manejar eventos ya sea mediante atributos de los controles en la vista o también mediante manejadores de eventos dentro del bean. Esto permite generar un entorno de desarrollo similar al que tiene java nativo en un sistema operativo en ventanas. Con la ocurrencia de los eventos, se hace el llamado a los diferentes procedimientos de la lógica de negocio. Es en este punto donde se fusiona el poder del lenguaje java con la flexibilidad de diseño que tiene una página web.

6. Renderizar respuesta

Luego de realizar los cambios sobre la vista y haber generado el árbol de componentes con todas sus validaciones y vínculos de atributos y procedimientos hacia el bean, se envía la respuesta hacia el usuario al renderizar la vista correspondiente. Luego de esto el contenido del árbol se almacena en memoria para que posteriormente se lo pueda cargar mucho más rápidamente.

2.3.2. Document Object Model (DOM)

Su traducción al español es Modelo de Objetos del Documento. Es un API que provee un estándar al momento de presentar los componentes de una página HTML o XML. Dentro de este estándar se especifica como los componentes se combinan, la interfaz estándar para poder ser accedidos y manejados, sus características, etc. El DOM brinda la posibilidad de modificar, acceder y reestructurar el contenido de las páginas HTML y XML permitiendo que estas sean dinámicas.

El DOM se comporta como un árbol donde los componentes se distribuyen de forma jerárquica, siendo la etiqueta `<body>` la principal. Al tener acceso a este árbol, se tiene un completo control sobre la estructura de la página, además de que se la puede alterar completamente mediante el uso de lenguajes como Javascript, Ajax o JQuery.

2.3.3. Javascript

Es un lenguaje de programación interpretado que permite el desarrollo de páginas web dinámicas. Se ejecuta desde el lado del cliente cuando la aplicación web es desplegada en el navegador permitiendo la interacción más avanzada con el usuario y con ciertos datos del navegador como el rastreo de la posición según la IP del dispositivo, detalles del sistema operativo, etc. También es usado en aplicaciones de escritorio como widgets, documentos PDF, etc.

La sintaxis de este lenguaje de programación es similar a la de C pero con la adopción de nombres y convenciones del lenguaje Java, aunque cabe recalcar que no está relacionado de ninguna forma con Java, y su propósito, modo de ejecución y semánticas son completamente diferentes.

2.3.4. Asynchronous JavaScript And XML (AJAX)

Técnica de desarrollo que permite la creación de aplicaciones web de manera mucho más interactiva que el desarrollo normal. AJAX mantiene una comunicación asíncrona en segundo plano con el servidor mientras se corre la aplicación en el lado del cliente, permitiendo realizar cambios sobre la página web sin necesidad de renderizarla por completo, es decir que permite recargar solo ciertas zonas de la página agregando ventajas como interactividad, eficiencia, velocidad, usabilidad, etc.

El modo en el que opera esta tecnología es asíncrono, debido a que realiza peticiones y obtiene datos del al servidor en un segundo plano sin afectar el rendimiento de la página. Para realizar esta acción utiliza el lenguaje Javascript y cada acceso al servidor se realiza mediante objetos XML Http Request, el mismo que permite la realización de peticiones HTTP y HTTPS al servidor.

2.3.5. JQuery

Es una librería de Javascript diseñada para un diseño de más alto nivel de páginas web. Brinda la posibilidad de modificar de una manera mucho más fácil una página HTML, manipular el árbol DOM a conveniencia del desarrollador, creación animaciones, implementar Skins para el diseño de interfaz, agregar funciones AJAX sobre la página y muchas más.

Existe un sinnúmero de extensiones y componentes prediseñados implementados sobre esta librería, lo cual hace que se requiera de menos recursos al momento del desarrollo e implementación de funcionalidades bastante grandes en una página normal HTML. Por esta razón es utilizado por frameworks y librerías de componentes para extender sus capacidades.

2.4. Características

PrimeFaces Mobile es una extensión de la librería PrimeFaces, por lo que hereda todas sus características, además de la mayoría de sus componentes. A continuación se enumeran unas de las principales características que hacen de estas librerías de componentes una potente herramienta para el desarrollo de aplicaciones móviles:

- Contiene un extenso conjunto de componentes ricos en funcionalidades, manejo de eventos, soporte Ajax, etc. Ejemplos de dichos componentes pueden ser: vistas renderizadas para navegadores de dispositivos móviles, contenedores de mapas enlazados a google maps, controles capaces de ejecutar funciones JQuery o Javascript, etc.
- Construcción de páginas web dinámicas basadas en Ajax y regidas bajo el estándar JSF 2.0 Ajax API. Esto permite la compatibilidad con varias tecnologías como RichFaces, IceFaces, el Core de JSF, etc.
- Dependiendo del tipo de tecnología o empaquetador que se esté utilizando, la librería puede ser implementada en la aplicación mediante dependencias o simplemente un archivo .jar internamente.
- Brinda lo posibilidad de implementar la tecnología Ajax Push, que mediante el uso de websockets, crea un canal bidireccional que permite enviar información desde el servidor hacia el cliente únicamente cuando es solicitada, a diferencia de la tecnología Ajax Poll que realiza un testeo constante de estado cliente para verificar cuando debe enviar información.
- Un Mobile UI Kit bajado en JQuery para el desarrollo de aplicaciones móviles personalizadas y dinámicas.
- Proporciona al desarrollador un Skinning Framework con más de 35 temas prediseñados que permiten acomodarse a casi cualquier diseño. Además es posible, mediante el uso de CSS²⁶, modificar el diseño visual de los componentes para acomodarlo al de la aplicación.
- Extensa documentación tanto en la página oficial de PrimeFaces (<http://www.primefaces.org/documentation.html>) como en blocks, foros y páginas anexas. Esto se debe a que la comunidad de desarrolladores de PrimeFaces es bastante extensa y está en continuo crecimiento.
- Procesos de validación parcial de los componentes y verificación de ciertas zonas de una página web.

²⁶ CSS: Hojas de estilo en cascada, permiten alterar el diseño de los componentes del árbol DOM

- Aplicación de la tecnología Ajax para realizar un submit únicamente en ciertas zonas de la página web, permitiendo el multiprocesamiento.
- Manejadores de eventos implementados no únicamente sobre la vista sino también en los Beans.
- Manejadores de eventos implementados directamente sobre la lógica de negocio (Beans).

2.4.1. Ventajas

- La implementación de Ajax permite la optimización al momento de la renderización y multiprocesamiento de la aplicación web.
- Contiene mayor variedad de componentes que otras librerías.
- Mayores capacidades de diseño de interfaz. Para el desarrollo de aplicaciones móviles existe la posibilidad de creación de Skins mediante JQuery Theme de forma gratuita.
- Capacidad de integración con otras librerías que corren sobre JSF.
- Gran cantidad de documentación.
- No requiere configuraciones adicionales para su implementación.
- No tiene dependencias con otras librerías.
- Facilidad de uso.
- Mayores posibilidades en cuanto al manejo de eventos tanto desde la vista como desde la lógica de negocios.
- Se especializa en el diseño de las aplicaciones.

2.4.2. Desventajas

- Al utilizar Ajax, en ciertos escenarios se pierde la capacidad de almacenar en el historial los estados que va teniendo una página web.
- Al utilizar ciertas versiones de JQuery, surgen errores de compatibilidad con ciertas tecnologías como por ejemplo GoogleCharts que implementa una versión distinta de JQuery que la de PrimeFaces.
- Tiene ciertas diferencias con los atributos de JSF nativo, por ejemplo para renderizar un control en JSF nativo se utiliza la función reRender, mientras que en PrimeFaces se utiliza la función update. Esto crea problemas de compatibilidad con librerías y controles que son implementados sobre JSF.
- Algunos de sus componentes de su versión mobile, no son compatibles con ciertos dispositivos móviles.

- En su versión mobile, PrimeFaces no tiene una documentación muy extensa.

2.5. Requerimientos de hardware y software de PrimeFaces Mobile

Se debe tomar en cuenta que se tiene dos partes necesarias para el funcionamiento de una aplicación móvil desarrollada con PrimeFaces Mobile: la una es el servidor (donde se ejecuta la aplicación) y la otra es la del cliente (donde se despliega el código HTML genera en el navegador, que en este caso es el de un dispositivo móvil).

Requerimientos de para el servidor sobre el que correrá PrimeFaces Mobile	
Hardware	Procesador DualCore de 3GHz o superior
	Mínimo 2Gb de memoria RAM
Software	JDK 1.6 actualización 45 o superior
	JSF 2.0
	PrimeFaces 3.4
	Servidor de Aplicaciones (Jboss AS, Apache Tomcat, Glassfish o cualquiera que soporte JSF 2.0)

Tabla 2.2: Requerimientos del servidor de PrimeFaces

Requerimientos de para el funcionamiento de PrimeFaces Mobile en un dispositivo móvil	
Hardware	Procesador 512Mhz
	Mínimo 128Mb de memoria RAM
	Pantalla táctil
Software	Navegador compatible con Ajax y HTML 5

Tabla 2.3: Requerimientos de funcionamiento dePrimeFaces

2.6. Comparación con otras tecnologías

		PrimeFaces	RichFaces	IceFaces
Funcionales	Componentes disponibles			
	Componentes de ingreso de datos	Si	Si	Si
	Paneles contenedores	Si	Si	Si
	Menús	Si	Si	Si
	Cuadros de dialogo	Si	Si	Si
	Gráficos estadísticos	Si	No	Si
	Componentes drag and drop	Si	Si	Si
	Multimedia	Si	Si	Si
	Efectos de visualización	Si	Si	Si
	Soporte Ajax			
	Eventos Ajax	Si	Si	Si
	Multiprocesamiento	Si	Si	Si
	Renderización parcial	Si	Si	Si
	Comunicación en background con capa de negocio	Si	Si	Si
	Ejecución procesos javascript	Si	Si	Si

	Ajax Listeners	Personalizable	Estándar	Estándar
	Renderización de pantalla	De acuerdo al navegador	Fijo	Fijo
	Validaciones en componentes			
	Expresiones regulares	Si	No	No
	Validaciones sobre capa de negocio	Si	Si	Si
	Validaciones sobre la vista	Si	Si	Si
	Skinning			
	Número de skins proporcionados	37	8	26
	Asistentes desarrollo skins	JQuery	CSS	CSS
	Tecnología Poll	Si	Si	Si
	Tecnología Push	Si	Si	Si
	Licenciamiento	Apache 2.0	LGPL v2.1	Apache 2.0
Usabilidad	Soporte	Fabricante, Comunidad Desarrollo	Fabricante, Comunidad Desarrollo	Fabricante, Comunidad Desarrollo
	Manuales	Avanzado	Avanzado	Medio
	Videos Demostrativos	Si	Si	Si
	Actividad Comunidad	Alta	Media	Baja
	Documentos de instalación	Si	Si	Si
	FAQs	Si	Si	Si
	Foros de Usuario	Avanzado	Avanzado	Medio
	Libros/Tutoriales	Disponible	Disponible	Parcialmente Disponible
	Showcase	Si	Si	Si
	Demostración en línea	Si	Si	Si
	Código demostrativo	Disponible	Disponible	Parcialmente Disponible
	Requerimientos de software			
	JDK mínimo	1.6	1.6	1.5
	JSF	2.0	2.0	2.0
Mantenimiento	Registro de errores del fabricante	No disponible	64%	93%
	Estabilidad de desarrollo			
	Numero de versiones estables	12	22	8
Portabilidad	Adaptabilidad			
	Sistemas operativos	Linux/Unix, Mac OS, Windows, Solaris, AIX	Linux/Unix, Mac OS, Windows, Solaris, AIX	Linux/Unix, Mac OS, Windows, Solaris, AIX

	Servidores Web	Tomcat, Wass, Appserver, WebLogic, Jboss AS, Websphere, Glashfish	Tomcat, Wass, Appserver, WebLogic, Jboss AS, Websphere, Glashfish	Tomcat, Wass, Appserver, WebLogic, Jboss AS, Websphere, Glashfish
	Navegadores soportados	Navegadores con soporte HTML 5	Navegadores con soporte HTML 5	Navegadores con soporte HTML 5
Instalación				
	Asistente de creación de proyectos	No	No	No
	Disponibilidad de librerías	Alta	Alta	Alta
	Disponibilidad de ayuda	Alta	Alta	Media
Coexistencia				
	Frameworks de persistencias	Hibernate, JDO, Oracle Toplink, iBatis, Jpa, EclipseLink, OpenJpa	Hibernate, JDO, Oracle Toplink, iBatis, Jpa, EclipseLink, OpenJpa	Hibernate, JDO, Oracle Toplink, iBatis, Jpa, EclipseLink, OpenJpa
	Otras librerías de componentes	Media	Alta	Alta
	Herramientas de reportaría web	Si	Si	Si
	Conectores JDBC	Si	Si	Si
	Frameworks web compatibles	Cualquier framework JSF	Cualquier framework JSF	Cualquier framework JSF

Tabla 2.4: Comparación entre PrimeFaces, IceFaces y RichFaces.

Capítulo III

Análisis y diseño de la aplicación

3.1. Análisis del sistema

3.1.1. Recolección de requerimientos

La orientación de este sistema es open source, por lo que su licenciamiento será gratuito. Para por realizar esto, todos sus componentes deberán tener también licenciamiento gratuito o ser de código abierto tanto en el manejo de la información alfanumérica y geoespacial como en el diseño y funcionamiento de la aplicación web.

Los requerimientos del sistema están basados en información brindada por el INPC (Instituto Nacional del Patrimonio Cultural) ubicado en la calle Benigno Malo 6-40 entre Presidente Córdova y Juan Jaramillo en la ciudad de Cuenca. Dicho instituto ha proporcionado un formulario de llenado de la información de un bien inmueble que consta como patrimonial.

De toda la información contenida en el formulario de bienes inmuebles patrimoniales, se tomarán en cuenta únicamente datos descriptivos como fue especificado en los objetivos del presente caso de estudio. Para lo cual se debe brindar al usuario la capacidad de seleccionar la información que desea ingresar previamente al sistema, por esta razón se deberá desarrollar una interfaz dinámica lo permita.

Dentro del sistema se tendrán módulos para el control tanto de la Aplicación SIG Móvil para el ingreso de la información geoespacial como para la administración propia del sistema y la consulta de la información ingresada mediante un computador de escritorio. Los módulos del sistema serán los siguientes:

- Administración del sistema

En cuanto a la administración del sistema se podrá seleccionar de forma dinámica la información que se quiere ingresar en cuanto al bien inmueble. Por ejemplo si en algún momento se quiere tener un registro de la fecha en la que fue vendido el predio, se tiene que únicamente seleccionar la categoría de información a la que pertenece y el tipo de dato, el mismo que puede ser:

- Listas de selección única (radio button)
- Listas de selección múltiple (check box)
- Listas desplegadas (combo box)
- Valores de ingreso, los mismo que pueden ser: Números, números y letras, fechas o de cualquier tipo sin restricción.

- Aplicación SIG móvil

Brinda la posibilidad de ingreso de la información desde el campo mediante un dispositivo móvil compatible con PrimeFaces Mobile. Debido a las limitaciones de usabilidad de los dispositivos móviles, se podrá ingresar únicamente los siguientes campos desde el mismo:

- Clave catastral del predio.
- Denominación o nombre con el que se conoce al bien inmueble.
- Descripción o una breve reseña.
- Propietario actual.
- Una referencia sobre la dirección o el sector donde se encuentra.
- Posición geoespacial tomada ya sea manualmente o leyendo la posición del GPS del dispositivo móvil.
- Fotografías almacenadas en el dispositivo móvil o tomadas directamente con la cámara del mismo si es que la tiene.

- Administración de usuarios

Los usuarios deben tener un nombre que los identifique y una contraseña, la misma que deberá ser cifrada. Además debe implementarse una tecnología para la seguridad al momento de login y restricciones que tienen los usuarios en el los módulos del sistema según los siguientes roles:

- Administrador (Acceso total)
- Operador (Restringido el control de usuarios y administración del sistema)
- Básico (Sólo consulta e ingreso de información desde la aplicación SIG móvil)

- Administración del Bienes Inmuebles

Luego del ingreso desde la aplicación SIG móvil, es necesario mantener la información y extenderla, debido a que la información ingresada desde el dispositivo móvil no está completa. Es necesario desarrollar una interfaz que lo permita.

- Consulta de bienes inmuebles patrimoniales de la ciudad de cuenca

La interfaz de consulta estará abierta al público sin necesidad de login y permitirá la visualización y filtrado en el mapa de los bienes inmuebles. Además brindará la posibilidad de ver toda la información completa de un bien inmueble seleccionado.

3.1.2. Análisis de la base de datos espacial

Dentro de una aplicación SIG móvil lo principal es la información geoespacial. El lugar donde se almacena dicha información es la geo base de datos. Por esta razón es indispensable hacer un análisis para determinar la que mejor se acople a los requerimientos del sistema. Se tomarán en cuenta únicamente las que dispongan de una versión gratuita, entre la cuales tenemos como principales en el mercado a PostGIS y MySQL Spatial Extension.

3.1.2.1. Desarrollo de los requerimientos

El sistema a desarrollar como ejemplo en el presente caso de estudio no tiene mayor cantidad de requerimientos, por lo que los requerimientos mínimos de la geo base de datos deben ser los siguientes:

- Licencia gratuita.
- Soporte para la plataforma GNU/Linux.
- Soporte de los estándares de OGC.
- Consulta de datos geoespaciales.
- Consulta de datos alfanuméricos.
- Integridad de datos.
- APIS o JDBC para el desarrollo de aplicaciones en el lenguaje de programación Java.
- Soporte para el manejo de datos vectoriales.
- Capacidad de importar información a la geo base de datos a partir archivos Shape.
- Soporte para geometrías de puntos, líneas, polígonos y multipolígonos (datos vectoriales).
- Capacidad de realizar transacciones sobre SQL para el ingreso, modificación, eliminación y consulta de la información geoespacial.
- Facilidad de integración con la base de datos alfanumérica y el servidor de mapas en la plataforma GNU/Linux.
- Eficiencia, eficacia y facilidad de uso.
- Documentación disponible.
- Generación de respaldos de la geo base de datos.

3.1.2.2. Tecnologías disponibles

3.1.2.2.1. PostGIS

PostGIS es una extensión de la base de datos relacional PostgreSQL. Es mantenido y desarrollado por la compañía Refrations Research. Su código fuente está distribuido bajo la licencia GNU (General Public License). Debido a la popularidad de esta extensión, es posible encontrar gran cantidad de documentación y ejemplos en la web.

PostGIS permite el ingreso de información geoespacial, a más de la información alfanumérica ya ingresada en PostgreSQL. Únicamente realiza la creación de base de datos según el tipo de geometrías que el usuario requiera. Incluye soporte para índices GiST basados en R-Tre, mejorando así la capacidad de realización de búsquedas. Incluye también funciones para el análisis y procesamiento de información geoespacial.

Este módulo de PostgreSQL brinda la posibilidad de implementación de funcionalidades topológicas para la construcción de Sistemas de Información Geográfica. Gracias a su soporte para los estándares OGC, tiene capacidades de almacenamiento y recuperación de datos según la especificación estándar SFS (Simple Feature Specification). La topología también se rige a dicha especificación estándar.

Características principales

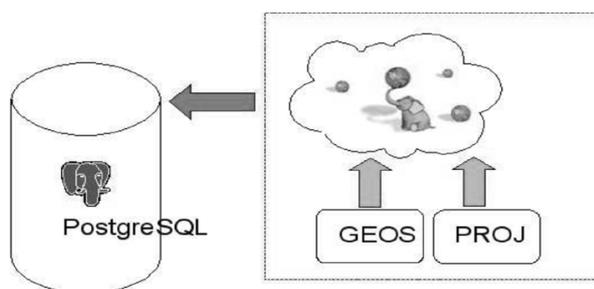
Una de las principales características de PostGIS es la eficiencia que ha presentado en cada una de sus versiones, y a su facilidad en cuanto a implementación y utilización. Es capaz de manejar un gran volumen de información, para lo cual utiliza índices GiST²⁷ (Generalized Search Trees). Entre las principales características de PostGIS se encuentran:

- Utilización de representaciones muy pequeñas de geometría y estructura de índices para mejorar el rendimiento de la base de datos espacial.
- Debido a que es una extensión de PostgreSQL, maneja integridad referencial tanto en la base de datos alfanumérica como en la espacial. Además usa un bloqueo a nivel de fila, permitiendo la concurrencia de procesos y garantizando aún más la integridad referencial.
- Soporte de estándares OGC mediante la especificación SFS (Simple Feature Specification), además de la implementación de las normas ISO SQL-Multimedia para el manejo de funcionalidades espaciales.

²⁷GiST: Índices de bases de datos basados en R-Tree que permiten consultas de grandes volúmenes de datos espaciales de una manera altamente óptima y rápida sin exigir alta cantidad de recursos de procesamiento.

- Agrega una amplia gama de funcionalidades para el análisis espacial como joins, buffers, intersections, polygon, line, linear referencing y muchas más.
- Al momento de consulta de datos geo espacial se dispone varias operaciones posibles, de las cuales las más importantes son: distance, containment, intersection, y varias matrices de relación.
- Soporte para varios tipos de datos como points, linestrings, polygons, multipoints, multilinestrings, multipolygons y geometrycollections.
- Manejo de índices GiST basados en R-Tree para aumentar la eficiencia de las consultas
- Conexión con lenguajes de programación como Phytom, Java, PHP, Ruby, etc.
- Posibilidad de realización de consultas híbridas entre información alfanumérica y espacial.

Arquitectura



Fuente: (Rodriguez 40)

Figura 3.7: Arquitectura de PostGIS

PostGIS utiliza dos librerías PROJ para dar soporte a la proyección dinámica de coordenadas en tiempo de ejecución (on-the-fly) y GEOS para la realización de pruebas y operaciones sobre los datos espaciales.

Principales funcionalidades

Importar y Exportar datos

La herramienta Shape Loader/Dumper brinda la capacidad de importar información geoespacial a partir de archivos ESRI Shape, que son los más populares del mercado debido a que su generación se realiza mediante ArcGIS²⁸.

²⁸ ArcGIS: Conjunto de productos de software desarrollados por la ESRI para la generación de archivos con información georreferenciada en un mapa (Sistemas de Información Geográfica)

Geometrías y funciones soportadas

De acuerdo a los estándares de OGC, todas las geometrías y funciones soportadas se rigen a la especificación Simple Features for SQL y extiende el estándar con soporte de coordenadas para 3DZ, 3DM y 4D. Entre las geometrías soportadas se dispone de point, linestring, polygon, multipoint, multilinestring, multipolygon y geometrycollection, y entre las funciones soportadas se dispone de disjoint, intersects, touches, crosses, within y contains.

PostGIS EWKB, EWKT y formas canónicas

Los estándares de OGC soportan geometrías en 2D únicamente, pero PostGIS ha extendido los formatos de OGC para llegar a soportar WKB/WKT en EWKB/EWKT para el poder llegar a manejar datos 3DM, 3DZ, 4D.

PostGIS WKT Raster (Well Known Text)

Existen funciones para el manejo simultáneo tanto de información raster como de información vectorial. PostGIS WKT Raster (Well Known Text) brinda la capacidad de ingreso de información raster como un nuevo tipo de datos dentro de PostgreSQL. Introduce además un set de funciones para trabajar con este nuevo tipo de datos.

3.1.2.2.2. MySQLSpatialExtension

MySQL Spatial Extensions aumenta las capacidades de MySQL de tal manera que permite la generación, almacenamiento y análisis de información geoespacial. Dicha información geoespacial está disponible para los siguientes motores internos de MySQL: MyISAM, InnoDB, NDB y tablas archive.

Para la representación de la información geoespacial se introduce un nuevo tipo de datos propuesto por los estándares de OGC bajo la especificación Simple Features Specifications For SQL, soportando así los formatos OGC Well-Known-Text (WKT) y OGC Well-Known-Binary (WKB). Para la el manejo de este nuevo tipo de datos geográficos se dispone de un set de funciones para el análisis y creación de valores.

Esta extensión tiene la capacidad de soportar una alta gama de geometrías para representar datos geoespaciales, entre los cuales se encuentran:

- Geometry
- Point
- Linestring

- Polygon
- Geometrycollection
- Multipoint
- Multilinestring
- Multipolygon

Para el análisis de la información geoespacial MySQL Spatial Extensions brinda una serie de funcionalidades tales como:

- Conversión de geometrías en varios formatos
- Propiedades cualitativas y cuantitativas
- Relaciones entre geometrías
- Creación de geometrías
- Consulta y análisis de geometrías

Características principales

- Soporte de estándares OCG mediante la especificación SFS (Simple Feature Specification).
- Soporta los formatos de datos espaciales Well-Known Text (WKT) y Well-Known Binary (WKB).
- Capacidades de implementar funciones como: conversión entre los tipos de datos de geoespaciales, consultas geoespaciales, creación y clonación de geometrías a partir de las ya existentes, comprobación de la relación entre objetos geométricos y geometrías, operaciones geoespaciales, relaciones entre rectángulos MBR (Minimal Bounding Rectangles), etc.
- Optimización del análisis espacial mediante la creación de índices espaciales.
- Jerarquización de los tipos de datos geométricos para especificar herencias de características y procedimientos de las mismas.
- Conexión con lenguajes de programación como Python, Java, PHP, Ruby, etc.
- Capacidad de realización de consultas híbridas entre información alfanumérica y geoespacial.

3.1.2.3. Comparación

Para la comparación de las bases de datos geográficas, se tomaran en cuenta los requerimientos presentados inicialmente.

Requerimientos	PostGIS	MySQL Spatial Extensions
Licencia gratuita	Si	Si
Soporte para la plataforma GNU/Linux	Si	Si
Soporte de los estándares de OCG	Si	Si
Consulta de datos geoespaciales	Si	Si
Consulta de datos alfanuméricos	Si	Si
Integridad de datos	Si	Si
APIS o JDBC para el desarrollo de aplicaciones en el lenguaje de programación Java	Si	Si
Soporte para el manejo de datos vectoriales	Si	Si
Capacidad de importar información a la geo base de datos a partir archivos Shape	Si	Con herramientas externas
Soporte para geometrías de puntos, líneas, polígonos y multipolígonos (datos vectoriales)	Si	Si
Capacidad de realizar transacciones sobre SQL para el ingreso, modificación, eliminación y consulta de la información geoespacial	Si	Si
Facilidad de integración con la base de datos alfanumérica y el servidor de mapas en la plataforma GNU/Linux	Si	Si
Eficiencia, eficacia y facilidad de uso	Si	Si
Documentación disponible	Alta	Baja
Generación de respaldos de la geo base de datos	Si	Si

Tabla 3.5: Comparación entre PostGIS y MySQL Spatial Extensions

3.1.2.4. Análisis y conclusión

Ambas herramientas cumplen con todos los requerimientos planteados inicialmente, pero MySQL Spatial Extentions tiene algunas falencias en cuanto la importación de información hacia la base de datos desde un archivo shape y también en cuanto a la disponibilidad de documentación para el desarrollo.

La importación de información desde un archivo shape es vital, ya que es el tipo de archivo que resulta de la utilización de la herramienta ArcGIS. Esta herramienta es la más popular del mercado y de donde se tomará previamente la información geoespacial.

MySQL Spatial Extentions no tiene una herramienta o plugin nativo que permita la importación de archivos shape, lo cual puede en ciertos casos producir problemas incompatibilidad y producir errores al momento importar la información. Mientras que PostGIS si tiene una herramienta nativa para realizar esta tarea: PostGIS shapefile Import/Export Manager.

La disponibilidad de la documentación interna de la herramienta y de la necesaria para el desarrollo es vital. En cuanto PostGIS, por ser una de las herramientas más populares del mercado, tiene una amplia documentación y ejemplos disponibles en la web. En cambio MySQL Spatial Extentions tiene una documentación mucho más reducida.

Tomando en cuenta lo anterior, ambas herramientas cumplen con los requerimientos, pero PostGIS los cumple de una mejor manera, además el motor de base de datos PostgreSQL sobre el cual se implementa PostGIS tiene una mejor capacidad de procesamiento, eficacia, eficiencia y posibilidades de desarrollo. Por estas razones se utilizará a PostGIS para desarrollar el sistema.

3.1.3. Análisis del servidor de mapas

El encargado de la interacción entre el cliente que accede a la aplicación y los datos geoespaciales es el servidor de mapas. Dependiendo de los requerimientos del sistema, este puede tener un sinnúmero de características. Debido a que se va a implementar únicamente un servicio WMS, los requerimientos no son muy altos.

3.1.3.1. Desarrollo de los requerimientos

De acuerdo al análisis de sistema previamente realizado, los requerimientos en cuanto a un servidor de mapas son los siguientes:

- Licencia gratuita
- Funcione sobre la plataforma GNU/Linux
- Tenga soporte para desarrollo sobre java
- Soporte la base de datos seleccionada en el análisis de la geo base de datos
- Permita el manejo de datos vectoriales
- Soporte el Geoservicio WMS
- Disponibilidad de documentación y ejemplos que hagan posible el desarrollo
- Bajos requerimiento de procesamiento
- Funcione bajo un servidor web compatible con la aplicación realizada
- Funcione bajo un estándar OGC

3.1.3.2. Tecnologías disponibles

3.1.3.2.1. MapServer

MapServer es una plataforma de código abierto que facilita la publicación de información geoespacial y aplicaciones SIG en la web. Fue creada a mediados de los 90 en la universidad de Minesota y es distribuida bajo una licencia gratuita similar a la de MIT. En la actualidad PSC (Project Steering Committee) es el encargado de la administración y manejo de esta plataforma. Al ser de código abierto, MapServer es mantenido, mejorado y soportado por desarrolladores de todo el mundo, por los que su documentación es muy extensa.

Esta plataforma fue desarrollada bajo el lenguaje C. Se concentra en pasar la información mediante la URL y la configuración para la visualización de las capas de SIG utilizando el servicio WMS se realiza mediante un archivo de configuración llamado MapFile. Este archivo de configuración es capaz de mostrar, a más de la imagen propiamente del mapa, información anexa a esta tal como leyenda, sombreados, barras de escala, mapas de diferencia, entre otros.

Existen herramientas como Quantum GIS Project que brindan un entorno gráfico para facilitar la creación del MapFile. Además existe una extensa documentación sobre el lenguaje utilizado para la codificación del MapFile, para mayor información se puede acudir a la página oficial de MapServer (<http://www.mapserver.org/mapfile/>).

Características principales:

- Funciona sobre la plataforma Linux (Apache) y Windows (MS4W)
- Soporte de datos tanto raster como vectoriales
- Gráficas de alta calidad
- Visualización personalizable mediante MapFile
- Soporta los siguientes formatos vectoriales: ESRI shapefiles, PostGIS, ESRI ArcSDE, GML y otros muchos vía OGR.
- Los formatos raster soportados son: JPG, PNG, GIF, TIFF/GeoTIFF, EPPL7 y otros vía GDAL.
- Fuentes TrueType
- Se configura en tiempo de ejecución mediante el envío de parámetros por la URL
- Para los desarrolladores se dispone de un API que soporta los lenguajes de programación PHP, Java, Perl, Python, Ruby o C#.
- Cumple con las especificaciones del OGC (Open Geospatial Consortium)

Ventajas:

- Licencia gratuita
- Código abierto
- Extensa documentación y soporte brindado por varias comunidades de desarrolladores
- Ciclo de desarrollo corto y rápido
- Soporte multiplataforma
- Personalizable en tiempo de ejecución
- Soporte para el desarrollo en varios lenguajes
- Tiene su propio estándar de personalización
- Fácil manejo del servicio WMS
- Tiene una alta madurez, ya que se viene desarrollando y perfeccionando desde 1996.
- Provee vectores dinámicos de alta calidad

Desventajas:

- Demasiada personalización en cuanto a la visualización mediante un MapFile
- No soporta servicios WFS-T
- No muchas compañías trabajan sobre el lenguaje C, que es en el que está desarrollado esta plataforma.
- Genera paquetes separados
- Tiene problemas de compatibilidad con software comercial

3.1.3.2.2. GeoServer

GeoServer es un servidor de mapas de código abierto que trabaja sobre la licencia GPL (General Public License). Es dirigido por la comunidad GeoServer y construido sobre la librería GeoTools. Permite la publicación y edición de información geoespacial. Está basado y desarrollado en el lenguaje Java. Utiliza estándares abiertos, por lo que su interoperabilidad es muy alta. Brinda también la posibilidad de conectarse con la información existente en globos virtuales como Google Earth y NASA World Wind.

Características principales:

- Soporte de estándares OGC
- Fácil de administración mediante herramientas vía web
- Extenso soporte de formatos de entrada como PostGIS, shapefile, ArcSDE, Oracle VFP, MySQL, MapInfo, WFS, etc.

- Certificación de OGC en los estándares WMS, WCS, WFS y WFS-T, además de la capacidad de implementación de los mismos
- La librería OpenLayers para el uso del servicio WMS viene integrada al servidor
- Puede ser integrado a cualquier contenedor de servlets o cualquier arquitectura ya sea de 32 o 64 bits.
- Incluye seguridad integrada y una herramienta de administración vía web
- Se integra con Mapping APIs, brindando así la capacidad de mostrar datos en aplicaciones como Google Maps, Google Earth, Yahoo Maps y Microsoft Virtual Earth.
- Soporte de formatos de salida tales como JPEG, GIF, PNG, SVG y GML
- Capacidad de edición de información geoespacial mediante los servicios WFS y WFS-T
- Dispone de funciones avanzadas para output de Google Earth incluyendo plantillas para pop-ups personalizados, visualizado de altitud y longitud, y "super-overlays".
- Es compatible con arquitecturas SIG tradicionales como ESRI ArcGIS

Ventajas:

- Licencia gratuita
- Integración con varias tecnologías
- Fácil administración
- Posibilidades de personalización de mensajes al servidor en los formatos http, XML, soap
- Acceso de manera uniforme hacia los datos
- Módulos integrados para el manejo de información y transacciones
- Estandarizado y certificado por OGC
- Control de concurrencia con aplicación de sistema de bloqueos (locking)
- Alta gama de tipos de datos soportados en entrada y salida
- Alto nivel de documentación disponible
- Posibilidades de edición de la información geoespacial

Desventajas:

- Complicado al momento de la compilación y configuración
- Se limita la configuración a través de interfaz gráfica
- Consume una alta cantidad de recursos de procesamiento y memoria RAM

3.1.3.3. Comparación

Requerimientos	MapServer	GeoServer
Licencia gratuita	Si	Si
Funcione sobre la plataforma GNU/Linux	Si	Si
Tenga soporte para desarrollo sobre java	Si	Si
Soporte la base de datos seleccionada en el análisis de la geo base de datos	Si	Si
Permita el manejo de datos vectoriales	Si	Si
Soporte el Geoservicios WMS	Si	Si
Disponibilidad de documentación y ejemplos que hagan posible el desarrollo	Alta	Alta
Requerimiento de procesamiento	Bajos	Altos
Funcione bajo los estándares OCG	Si	Si
Funcione bajo un servidor web compatible con la aplicación realizada	Si	Si

Tabla 3.5: Comparación entre MapServer y GeoServer

3.1.3.4. Análisis y conclusión

Tanto MapServer como GeoServer cumplen con los requerimientos planteados inicialmente. Su única diferencia está los requerimientos de procesamiento. GeoServer tiene mayor potencia y capacidades que MapServer, por los que necesita mayor cantidad de recursos de procesamiento.

Para el sistema que se va a desarrollar se necesita que el servidor de mapas sea lo más liviano posible, debido a que el servidor de aplicaciones de java que se va a utilizar es JBoss AS 7.1.1 Final. Este servidor de aplicaciones consume gran cantidad de memoria RAM y recursos de procesamiento.

Únicamente se va a implementar un servicio WMS para presentar y filtrar gráficamente la información geoespacial hacia el lado del cliente y MapServer es capaz de realizar esta actividad sin ningún problema. MapServer es uno de los servidores de mapas más livianos del mercado en cuanto al consumo de requerimientos de procesamiento y memoria RAM, por lo que será utilizado en el desarrollo del sistema.

3.2. Diseño del sistema

3.2.1. Diagramas de Casos de Uso

Como se mencionó anteriormente en los requerimientos del sistema, la totalidad del sistema se divide en 5 módulos. Por esta razón se ha dividido el diagrama de casos de usos en 5 partes para mantenerlo ordenado y su entendimiento sea más fácil.

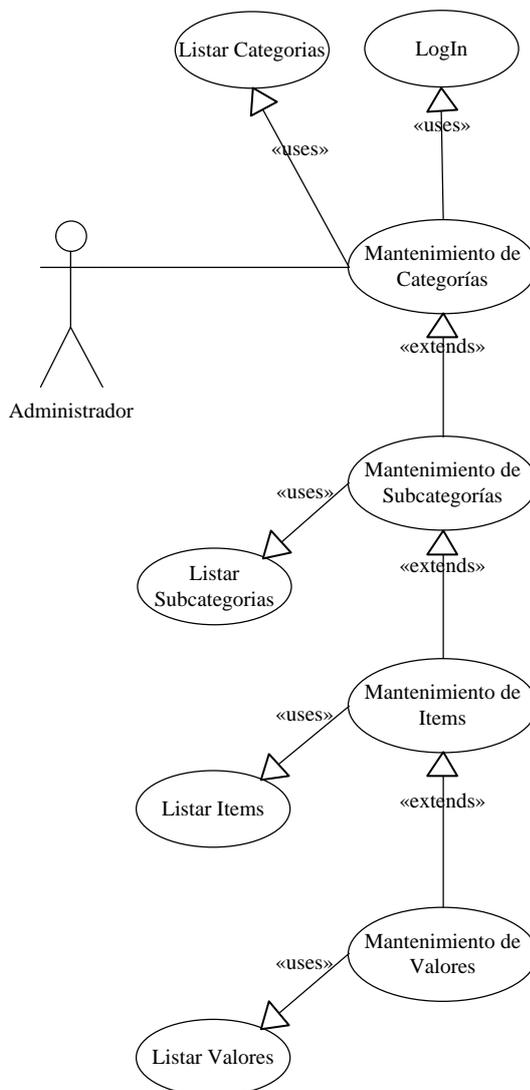


Diagrama de casos de uso de Administración del sistema

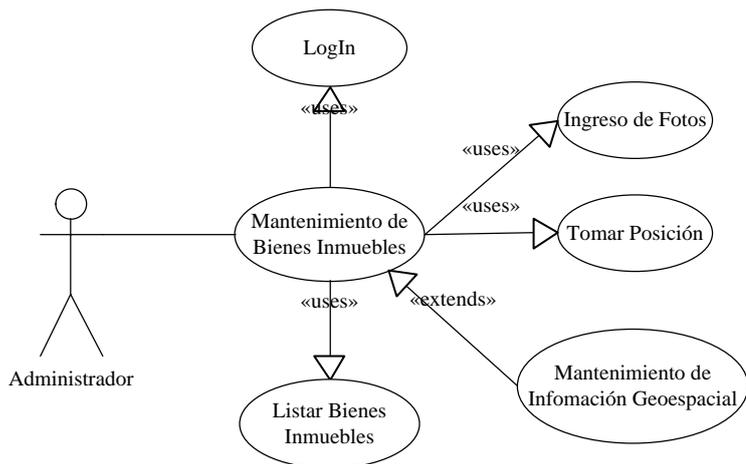


Diagrama de casos de uso de Aplicación SIG móvil

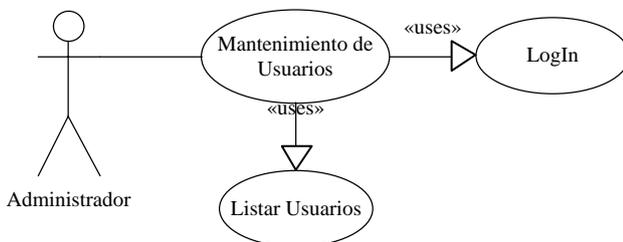


Diagrama de casos de uso de Administración de usuarios

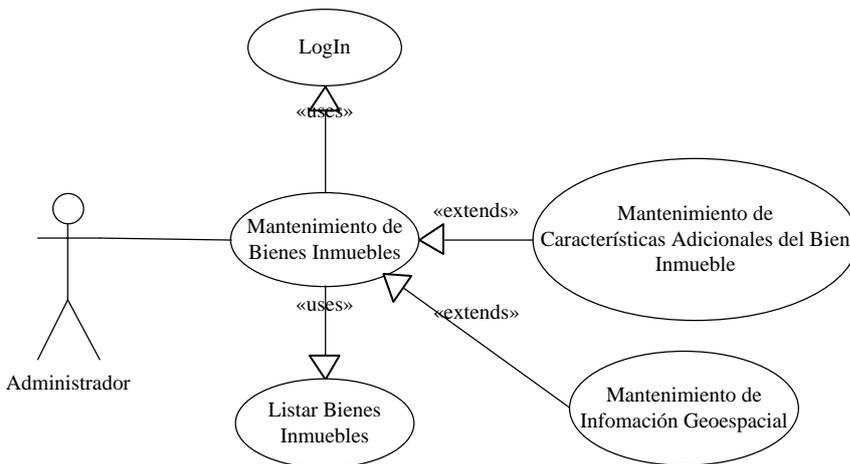


Diagrama de casos de uso de Administración del Bienes Inmuebles

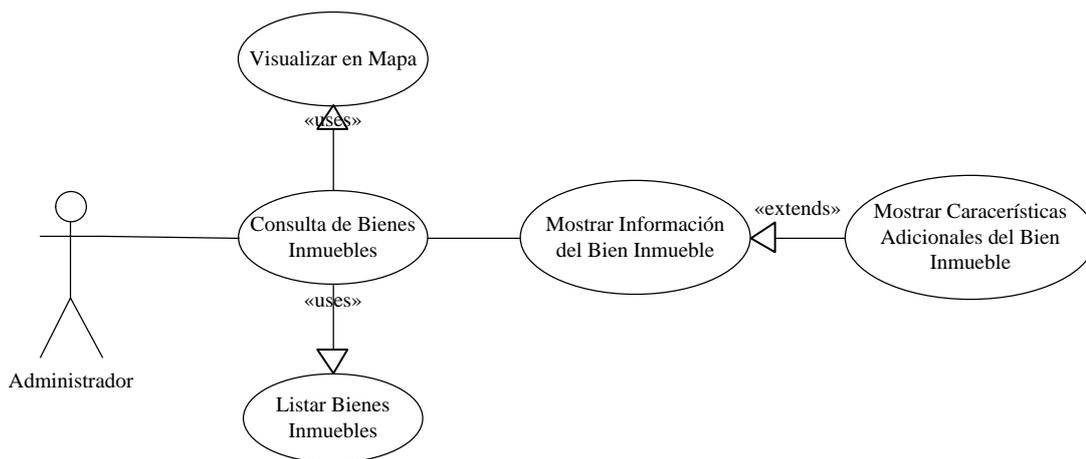


Diagrama de casos de uso de Consulta de bienes inmuebles

3.2.2. Diagramas de flujo de datos

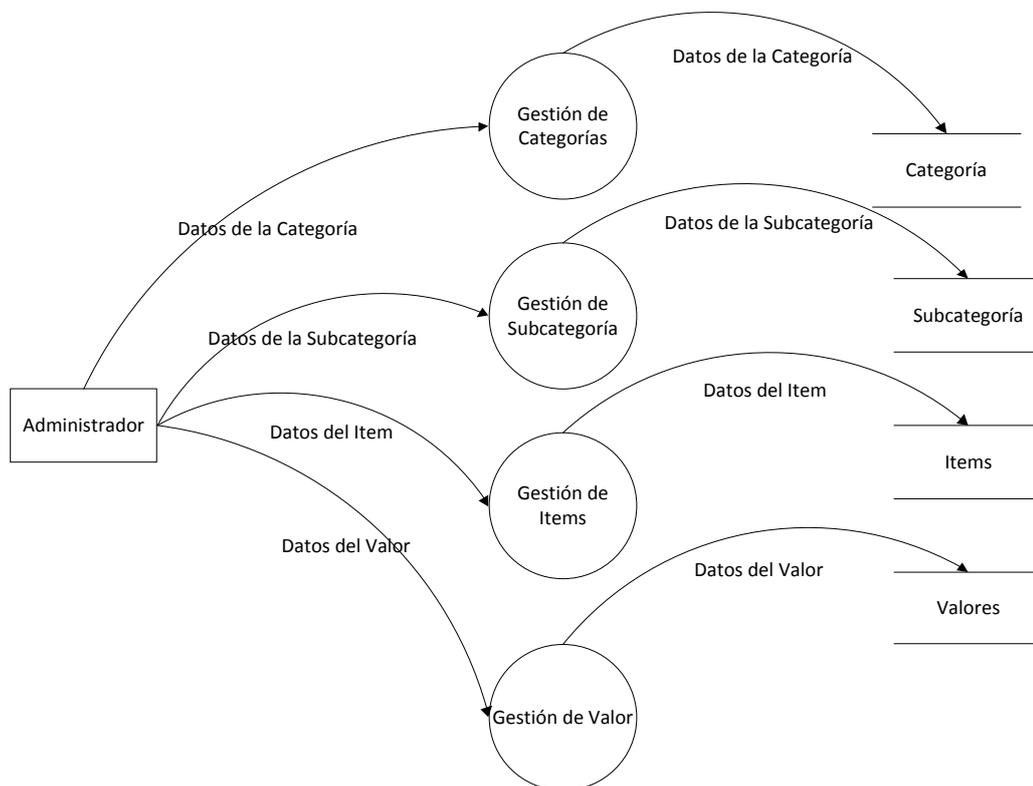


Diagrama de Administración del sistema

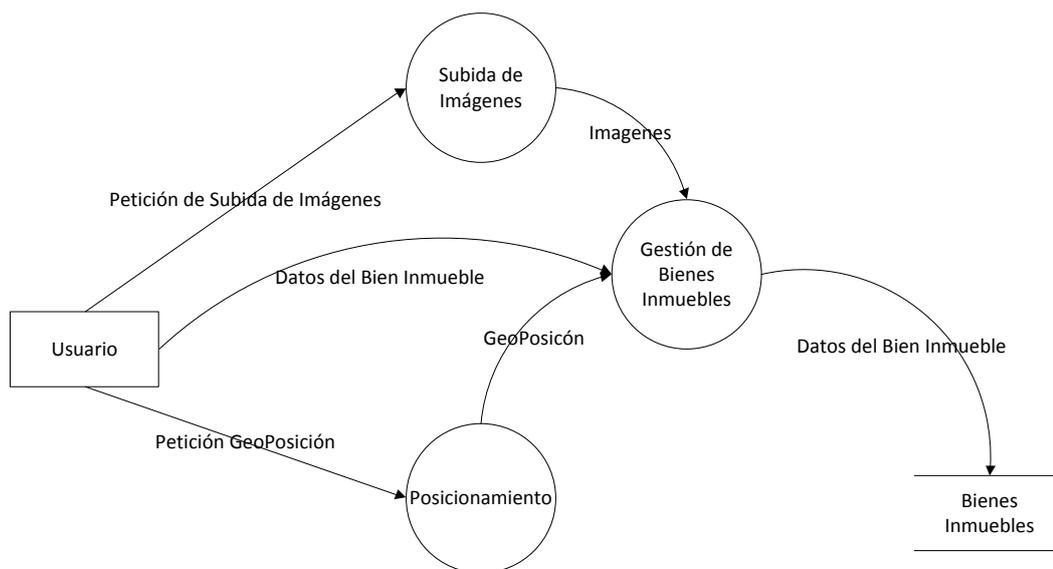


Diagrama de Aplicación SIG móvil

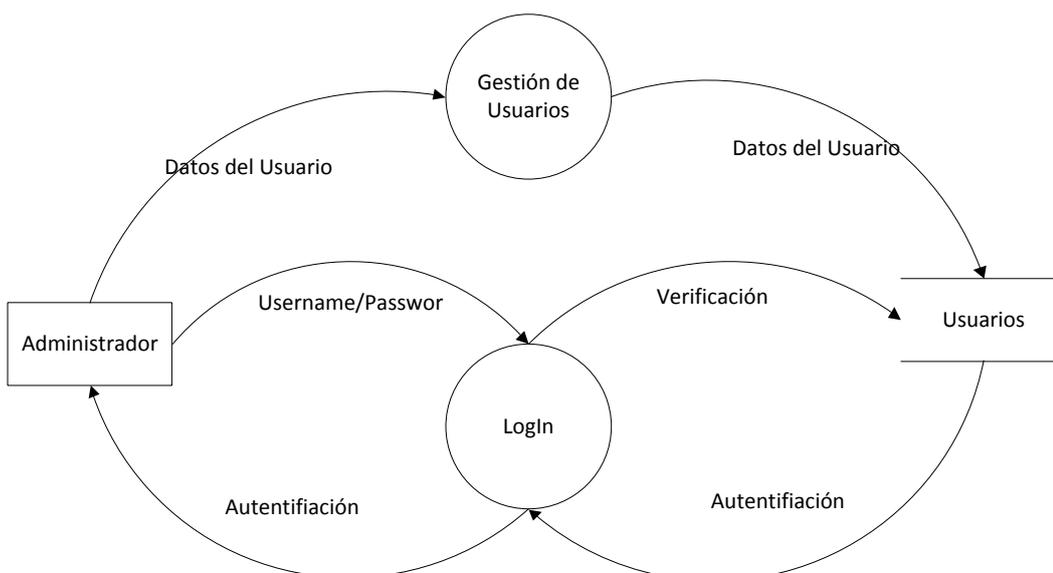


Diagrama de Administración de usuarios y autenticación (LogIn)

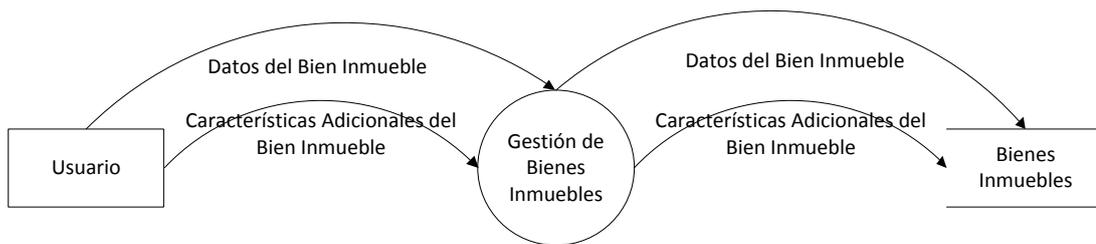


Diagrama de Administración del Bienes Inmuebles

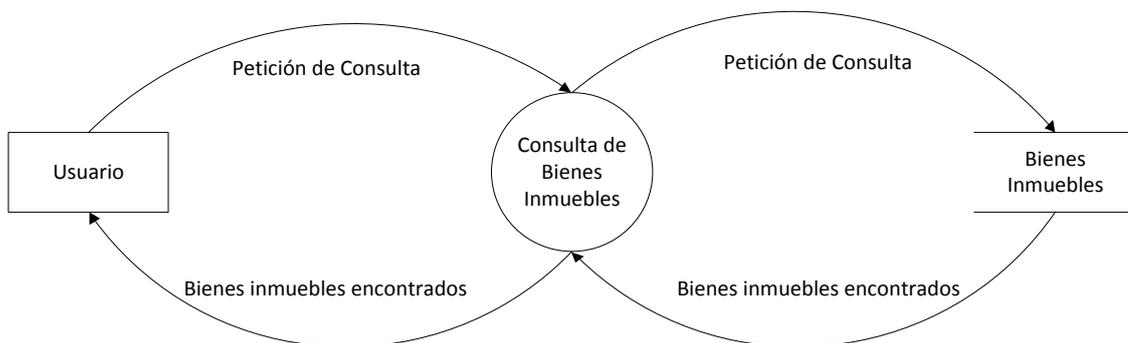
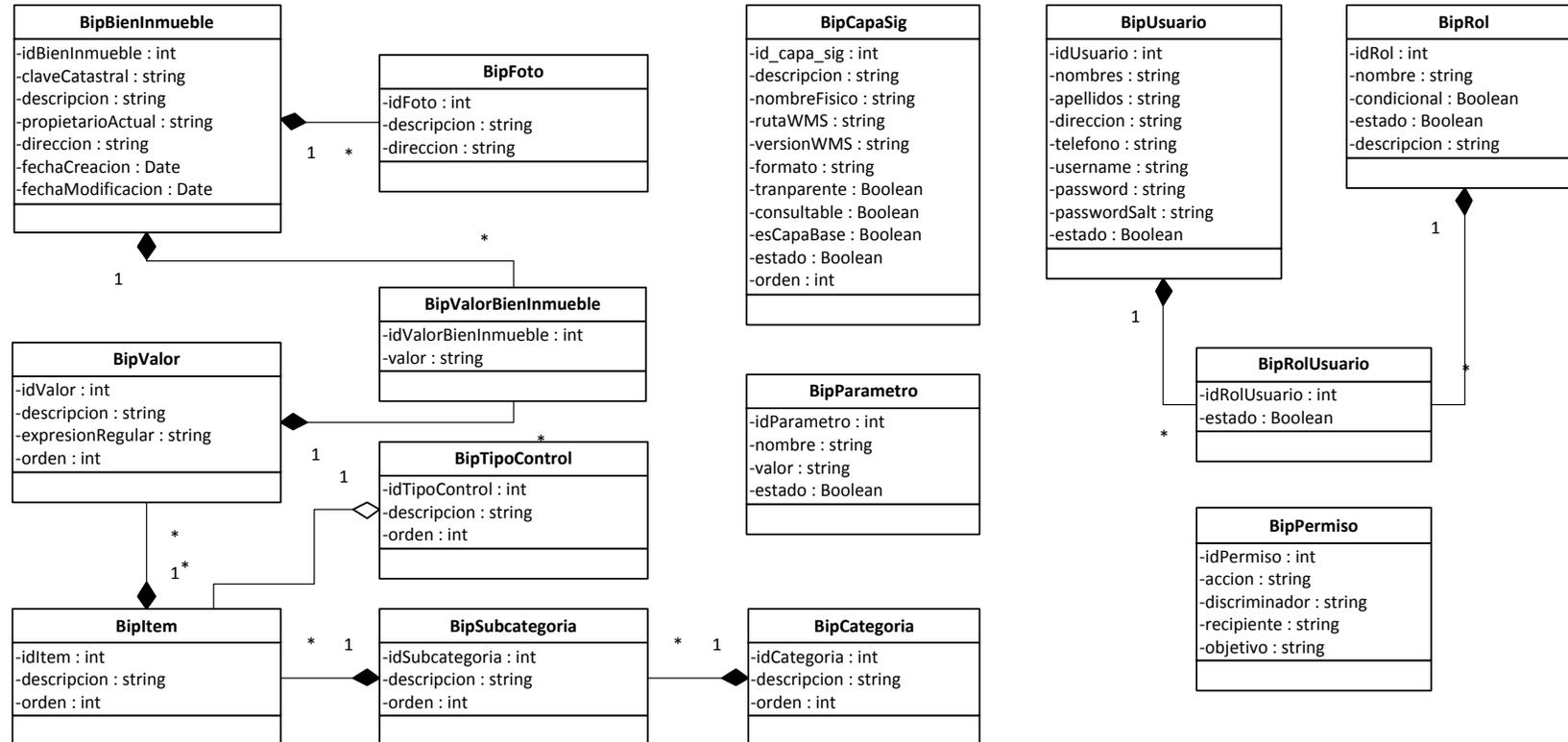
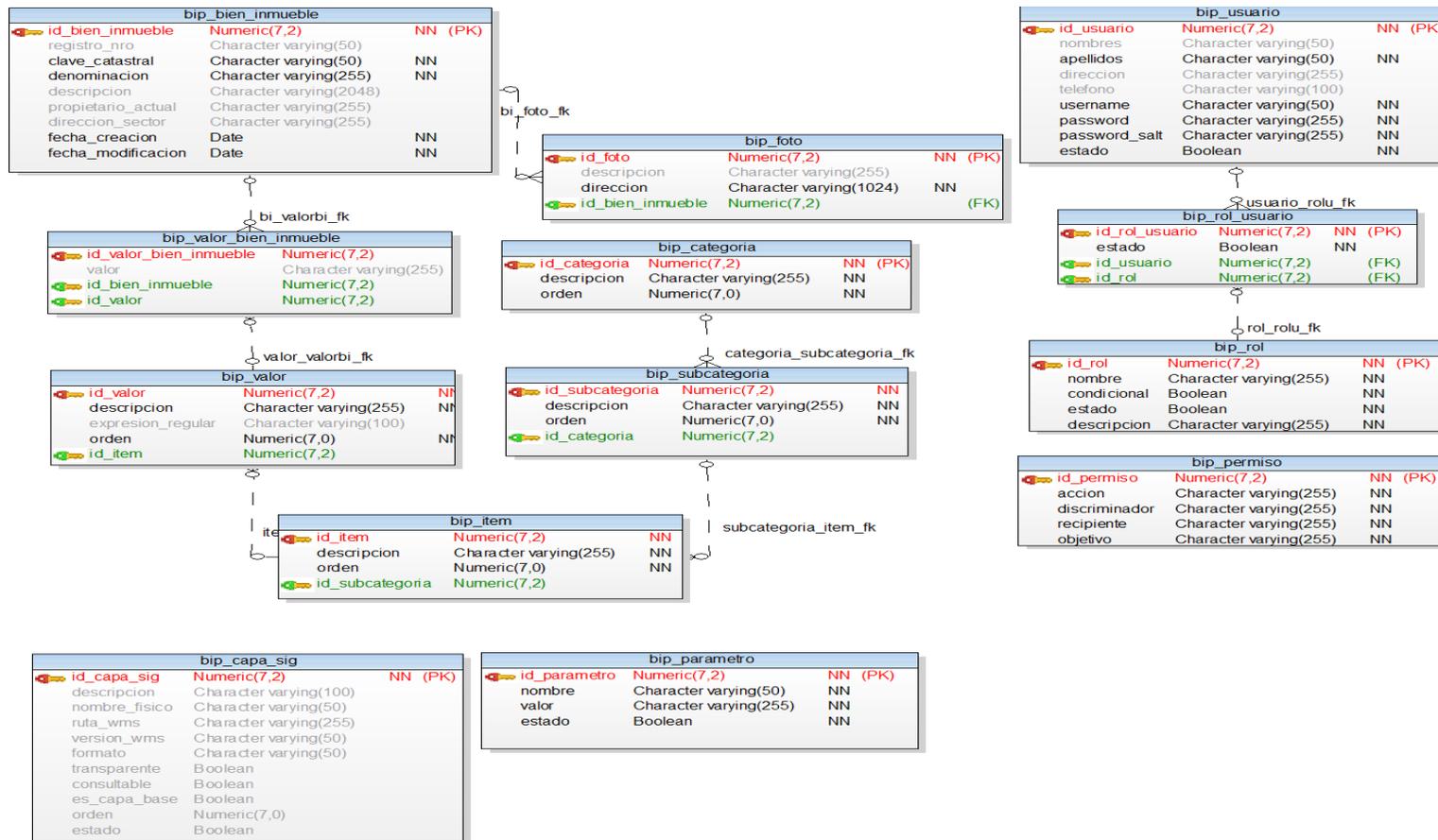


Diagrama de Consulta de bienes inmuebles

3.2.3. Diagrama de clases



3.2.4. Diagrama de base de datos



3.2.5. Diccionario de datos

A continuación se realiza una descripción detallada de todas las tablas de base de datos, además de los campos con sus tipos de datos, claves primarias, claves foráneas, tablas a las que se referencian, largo del campo, valor por defecto, permisión de nulos y comentarios.

Nombre de la Tabla: bip_bien_inmueble

Descripción de la Tabla: Almacena los datos del bien inmueble además de su geometría.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_bien_inmueble	integer		32,0		NO	Condigo del bien inmueble
			clave_catastral	character varying	50			NO	Clave catastral del bien inmueble
			denominacion	character varying	255			NO	Denominación del bien inmueble
			descripcion	character varying	2048			YES	Descripción del bien inmueble
			Geom	gemetry				YES	Capa de bienes inmuebles de la ciudad de cuenca
			propietario_actual	character varying	255			YES	Propietario actual del bien inmueble
			Dirección	character varying	255			YES	Dirección del bien inmueble
			fecha_creacion	date				NO	Fecha de creación del registro
			fecha_modificacion	date				NO	Fecha de última modificación del registro

Nombre de la Tabla: bip_foto

Descripción de la Tabla: Almacena las direcciones físicas de las fotos agregadas al bien inmueble.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_foto	numeric		7,0		NO	Código de la foto
			Descripción	character varying	255			YES	Descripción del contenido de la foto
			Dirección	character varying	1024			NO	Dirección del lugar del servidor donde se encuentra almacenada la foto
	#	bip_bien_inmueble	id_bien_inmueble	Integer		32,0		NO	Bien inmueble al que le pertenece la foto
#			id_foto	Numeric		7,0		NO	Código de la foto

Nombre de la Tabla: bip_categoria

Descripción de la Tabla: Almacena las categorías de información para ser adjuntada al bien inmueble.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_categoria	Numeric		7,0		NO	Código de la categoría
			Descripción	character varying	255			NO	Descripción de la categoría
			Orden	Numeric		7,0		YES	Orden de las categorías

Nombre de la Tabla: bip_subcategoria

Descripción de la Tabla: Almacena las subcategorías de información para ser adjuntada al bien inmueble.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_subcategoria	Numeric		7,0		NO	Código de la subcategoría
			descripcion	character varying	255			NO	Descripción de la subcategoría
			orden	Numeric		7,0		NO	Orden de la subcategoría
	#	bip_categoria	id_categoria	Numeric		7,0		YES	Categoría padre

Nombre de la Tabla: bip_item

Descripción de la Tabla: Almacena los ítems para ser adjuntados al bien inmueble.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_item	Numeric		7,0		NO	Código del ítem
			descripcion	character varying	255			NO	Descripción del ítem
	#	bip_tipo_control	id_tipo_control	Numeric		7,0		NO	Tipo de control al que pertenece el ítem
	#	bip_subcategoria	id_subcategoria	Numeric		7,0		NO	Subcategoría a la que pertenece el ítem
			orden	Numeric		7,0		YES	Orden de los ítems

Nombre de la Tabla: bip_valor

Descripción de la Tabla: Almacena los valores de los ítems para ser adjuntados al bien inmueble.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_valor	Numeric		7,0		NO	Código del valor
			descripcion	character varying	255			NO	Descripción del valor
			expresion_regular	character varying	100			YES	Expresión regular del valor para su próxima validación en el sistema
	#	bip_item	id_item	Numeric		7,0		NO	Ítem al que pertenece el valor
			orden	Numeric		7,0		YES	Orden de los valores

Nombre de la Tabla: bip_valor_bien_inmueble

Descripción de la Tabla: Almacena los valores seleccionados en el módulo de características adicionales del bien inmueble.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_valor_bien_inmueble	Numeric		7,0		NO	Código del valor del bien inmueble
			valor	character varying	255			YES	Valor que tiene el ítem en caso de ser de tipo texto
	#	bip_bien_inmueble	id_bien_inmueble	Integer		32,0		NO	Bien inmueble al que pertenece el ítem
	#	bip_valor	id_valor	Numeric		7,0		NO	Valor al que se enlazara el bien inmueble

Nombre de la Tabla: bip_tipo_control

Descripción de la Tabla: Almacena los tipos de controles que se puede tener en la generación dinámica de las características del bien inmueble.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_tipo_control	Numeric		7,0		NO	Código del tipo de control
			descripcion	character varying	50			NO	Descripción del tipo de control

Nombre de la Tabla: bip_usuario

Descripción de la Tabla: Almacena los datos de los usuarios además de su username y password.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_usuario	Numeric		7,0		NO	Código del usuario
			nombres	character varying	50			NO	Nombres del usuario
			apellidos	character varying	255			NO	Apellidos del usuario
			direccion	character varying	255			YES	Dirección del usuario
			telefono	character varying	100			YES	Teléfono del usuario
			username	character varying	50			NO	Username único para el login del usuario
			password	character varying	255			NO	Password del usuario cifrado a 128 bits
			password_salt	character varying	255			NO	Llave para desencriptar el password del empleado
			estado	Boolean			true	NO	Estado del usuario

Nombre de la Tabla: bip_rol

Descripción de la Tabla: Almacena los tipos de roles que se puede tener en la aplicación.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_rol	Numeric		7,0		NO	Código del rol
			nombre	character varying	255			NO	Nombre del rol
			condicional	Boolean				NO	Condicional del rol
			estado	Boolean				NO	Estado del rol
			descripcion	character varying	255			YES	Descripción a ser mostrada al usuario

Nombre de la Tabla: bip_rol_usuario

Descripción de la Tabla: Almacena la asignación realizada de cada usuario a su respectivo rol.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_rol_usuario	numeric		7,0		NO	Código del rol del usuario
			estado	boolean			true	NO	Estado del rol del usuario
	#	bip_usuario	id_usuario	numeric		7,0		NO	Usuario al que será agregado el rol
	#	bip_rol	id_rol	numeric		7,0		NO	Rol que será asignado al usuario

Nombre de la Tabla: bip_permiso

Descripción de la Tabla: Almacena los diferentes permisos que puede el usuario tener sobre el acceso a los datos.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_permiso	numeric		7,0		YES	Identificador del permiso
			accion	character varying	255			YES	Acción del permiso
			discriminador	character varying	255			YES	Discriminador del permiso
			recipiente	character varying	255			YES	Recipiente del permiso
			objetivo	character varying	255			YES	Objetivo del permiso

Nombre de la Tabla: bip_parametro

Descripción de la Tabla: Almacena el valor de los parámetros de la aplicación.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_parametro	numeric		7,0		NO	Código del parámetro
			nombre	character varying	50			NO	Nombre del parámetro
			valor	character varying	255			NO	Valor del parámetro
			estado	boolean				YES	Estado del parámetro

Nombre de la Tabla: bip_capa_sig

Descripción de la Tabla: Almacena la configuración de las diferentes capas del SIG.

PK	Clave Foránea	Tabla referenciada	Campo	Tipo de dato	Largo Cadena	Largo Número	Valor defecto	Permite nulos	Comentario
#			id_capa_sig	numeric		7,0		NO	Código de la capa sig
			descripcion	character varying	100			YES	Descripción de la capa sig
			nombre_fisico	character varying	50			YES	Nombre con el que está en la base de datos
			ruta_wms	character varying	255			YES	Ruta WMS para hacer el llamado desde OpenLayers
			version_wms	character varying	50			YES	Versión del Servicio WMS que se implementa
			formato	character varying	50			YES	Formato de la imagen a generar
			transparente	boolean				YES	Si es transparente
			consultable	boolean				YES	Si puede ser consultado
			es_capa_base	boolean				YES	Si es tomado como capa base
			estado	boolean				YES	Estado del registro
			orden	numeric		7,0		YES	Orden en el que se encuentra la capa

3.2.6. Diseño de interfaz

En cuanto a la plantilla sobre la cual se desarrollará toda la plataforma, se ha buscado tener colores con tonalidades envejecidas debido a que se trata de un inventario de bienes inmuebles patrimoniales.



Para la interfaz de mantenimientos se tiene que modificar los Skins que trae PrimeFaces por defecto debido a que sus colores predefinidos pueden afectar el diseño de la plantilla inicial.

Bienes Inmuebles Patrimoniales de la Ciudad de Cuenca

Conectado como: drodriguez Cerrar Sesión

Inicio Administración Consulta Acerca Contactenos

Menu

- Bienes Im (General)
- Bienes Im (Mapa)
- Bienes Im (Características)
- Usuarios
- Sistema

Links externos

- » Universidad del Azuay
- » INPC
- » UNESCO
- » Ministerio Coordinador del Patrimonio

Categorías

Nueva categoría

Descripción *

Orden *

Crear

Lista de usuarios

Descripción	Orden	Opciones
Información general de la edificación	1	X
Descripción de la edificación	2	X
Intervenciones	3	X
Valoración del bien inmueble	4	X
Levantamiento físico del inmueble	5	X
Grados de intervención requerida	6	X
Observaciones generales	7	X
Valoración del inmueble Baremo	8	X

Realizado por:

En el diseño de la interfaz de consulta de bienes inmuebles debe tomarse en cuenta la configuración de colores que se crea dentro del MapFile para que combine con los colores de la plantilla.

The screenshot displays a web-based GIS application interface. It features a central map area with a cyan background and a dark blue landmass. The map includes a scale bar (0 to 2 km) and a north arrow. To the left of the map is a sidebar with several sections: 'Mini Mapa' showing a small map of Ecuador with a red box over Cuenca; 'Buscador' with a search input field and a 'Buscar' button; and 'Capas' with a list of layers: 'Parroquias de Cuenca', 'Manzanas de Cuenca', 'Predios de Cuenca', 'Vias de Cuenca', and 'Bienes Inmuebles Patrimoniales'. Below the sidebar is a table titled 'Lista de Bienes Inmuebles' with the following data:

Clave Catastral	Denominacion	Propietario	Descripcion	Opciones
60465406a5400467	Pileta 1	Carlos Piedra	Pileta	 
1233	UDA	UDA	Universidad del Azuay	 
ghj	vbn	vghj	ghj	 

At the bottom of the table, there is a page indicator '1/1'.

Para la interfaz móvil se dejará el Skin que trae PrimeFaces Mobile, debido a que está normalizado a como se ve una aplicación desarrollada para dispositivos móviles con sistema operativo IOS.



Capítulo IV

Requerimientos Técnicos

4.1. Instalación y creación de las bases de datos alfanumérica y geoespacial

Según el análisis previamente realizado, se determinó que la base de datos alfanumérica a utilizar para el desarrollo del sistema será PostgreSQL y la base de datos geoespacial será PostGIS. Para el presente caso de estudio ambas serán instaladas sobre la plataforma Ubuntu 12.10 (Linux).

4.1.1. Requerimientos de software

- Un compilador ISO/ANSI C. GCC el compilador por defecto en Linux funciona correctamente.
- tar, gzip o bzip2 para desempaquetar las fuentes.
- GNU make (gmake)
- Biblioteca GNU Readline
- Biblioteca de compresión zlib
- Perl y python para soporte de PL/Perl y PL/Python

Nota: No hay mayor preocupación en cuanto los requerimientos de software, ya que en las versiones más actuales de Linux vienen instalados por defecto a excepción de algunas veces en las que es necesario instalar únicamente el paquete Make.

4.1.2. Requerimientos de hardware

- Un procesador Pentium IV de 2.6 GHz o superior.
- 1 GB de memoria RAM o más.
- 10 GB o más de espacio de almacenamiento en el disco duro.

4.1.3. Pasos para la instalación

1. Descargar el instalador en modo gráfico de la base de datos PostgreSQL 9.2.1.1 de su página principal: <http://www.postgresql.org/download/linux/ubuntu/>.
2. Brindar privilegios de ejecución al instalador.

```
sudo chmod a+x postgresql-9.2.1-1-linux.run
```

3. Iniciar el instalador con el siguiente comando:

```
sudo ./postgresql-9.2.1-1-linux.run
```



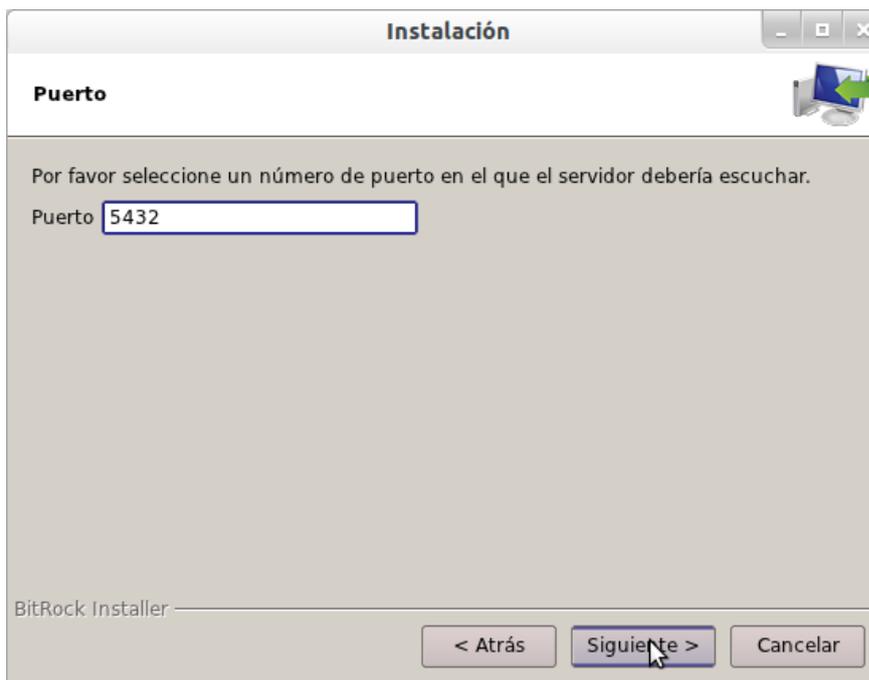
4. En la pantalla actual dar un clic en siguiente.
5. Seleccionar la carpeta donde será instalado PostgreSQL. En este caso dejará la carpeta por defecto, ya que es un estándar para de instalación en la plataforma Linux.



6. Ingresar la contraseña de la base de datos



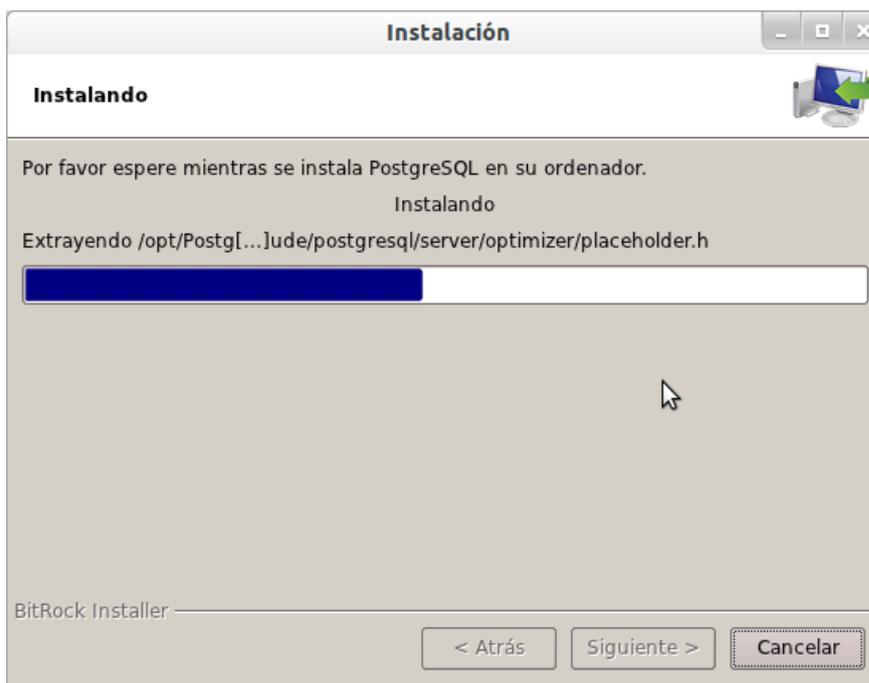
7. Seleccionar el puerto por donde el servidor de la base de datos va a escuchar las peticiones.



8. Seleccionar la configuración regional que determinará el idioma de la base de datos, que en este caso será es_EC.utf8 para ecuador.



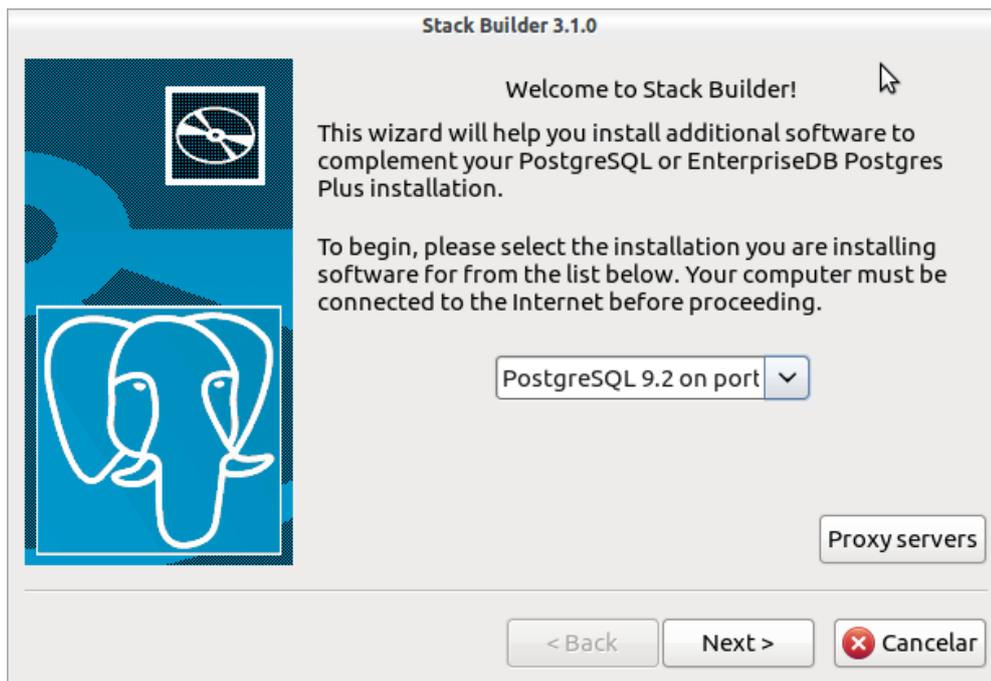
9. En la siguiente pantalla dar un clic en siguiente y la instalación comenzará.



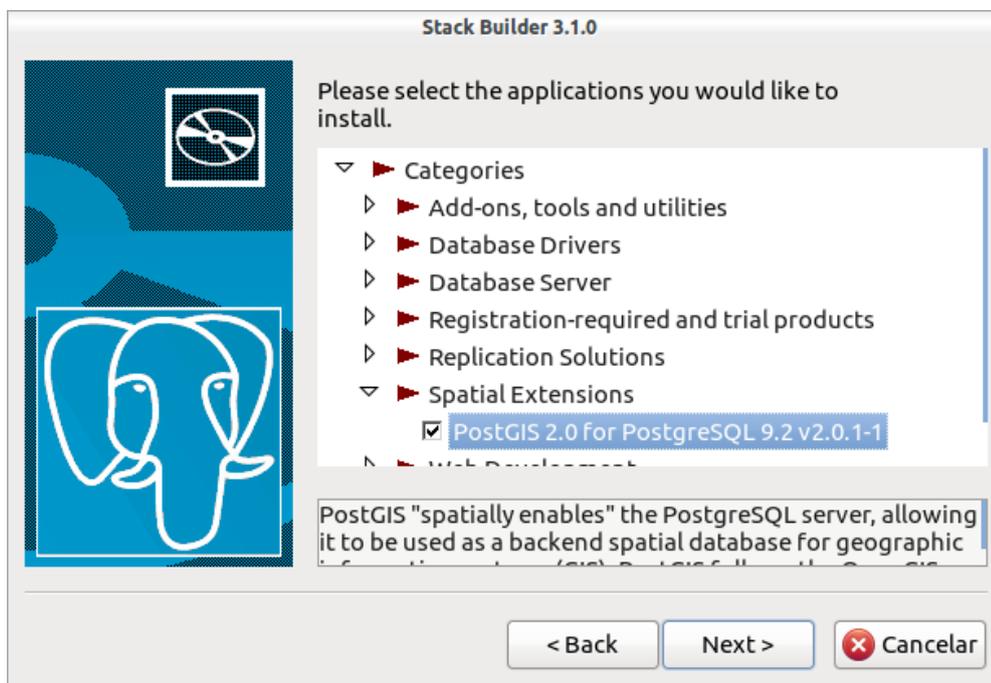
10. Una vez terminada la instalación el asistente pregunta si desea lanzar el Stack Builder, el mismo que permite descargar extensiones de PostgreSQL. Es aquí de donde se descargará PostGIS. Seleccionar lanzar Stack Builder y presionar en terminar.



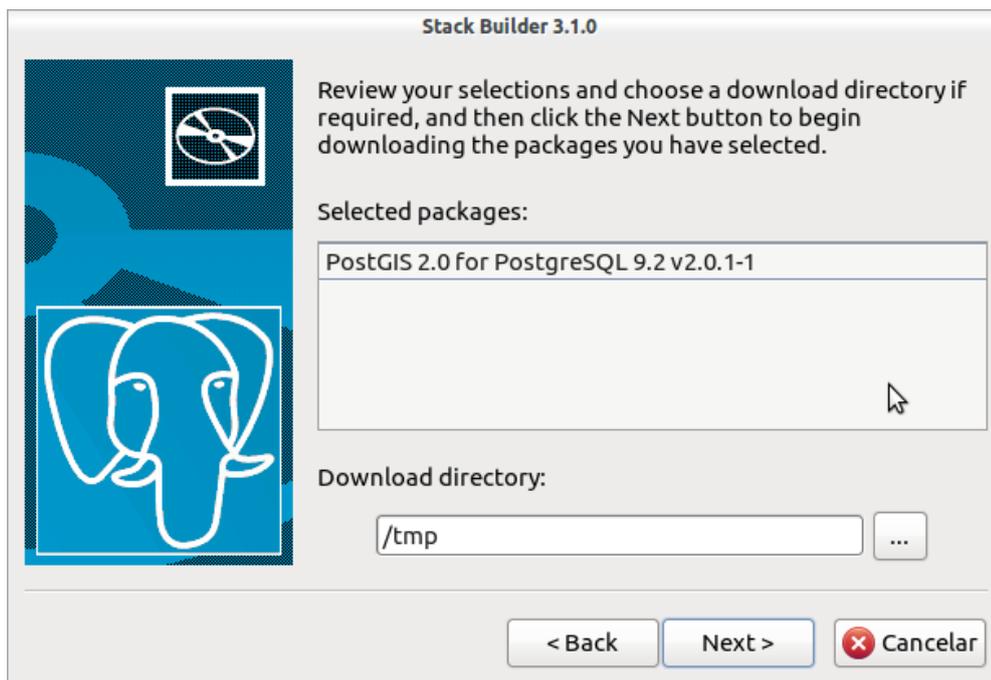
11. En la siguiente pantalla seleccionar PostgreSQL 9.2 on port 5432.



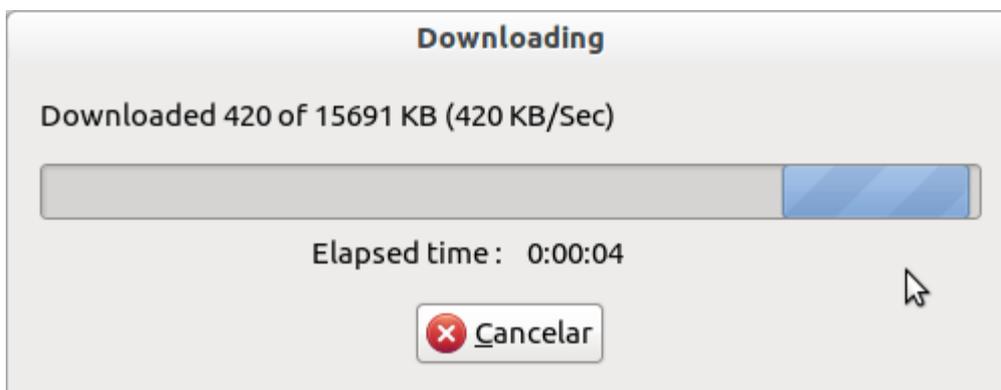
12. Stack Builder pone a disposición del usuario gran cantidad de extensiones, de las cuales se debe seleccionar PostGIS 2.0 from PostgreSQL 9.2 v2.1.1, la misma que se encuentra en la categoría Spatial Extensions.



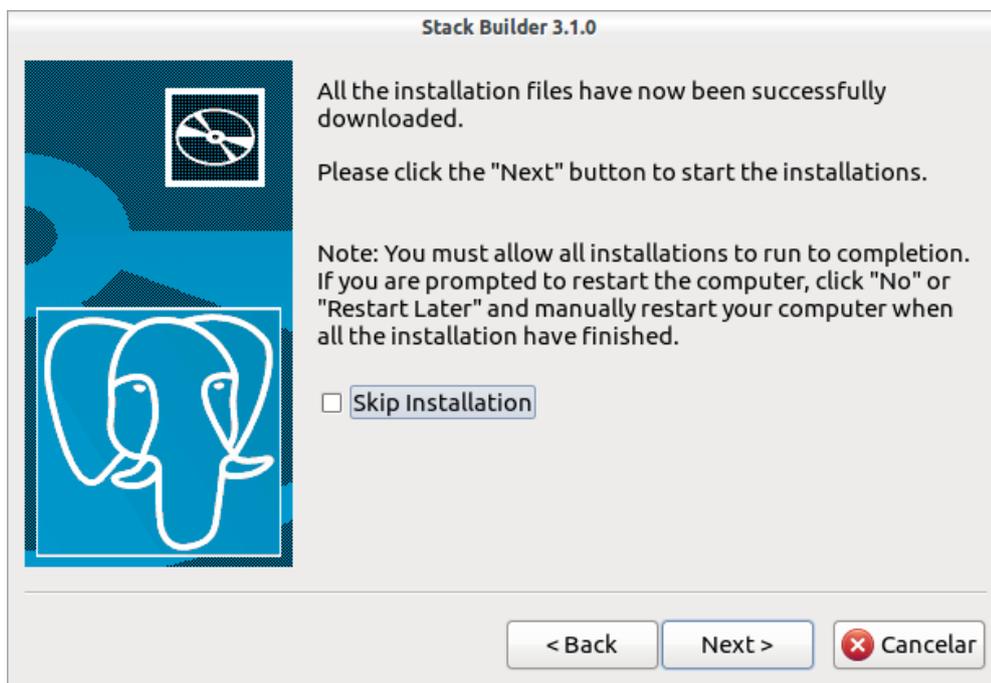
13. El asistente pedirá una dirección donde descargar temporalmente el instalador de PostGIS para su previa instalación. Dejaremos por defecto la carpeta temp.



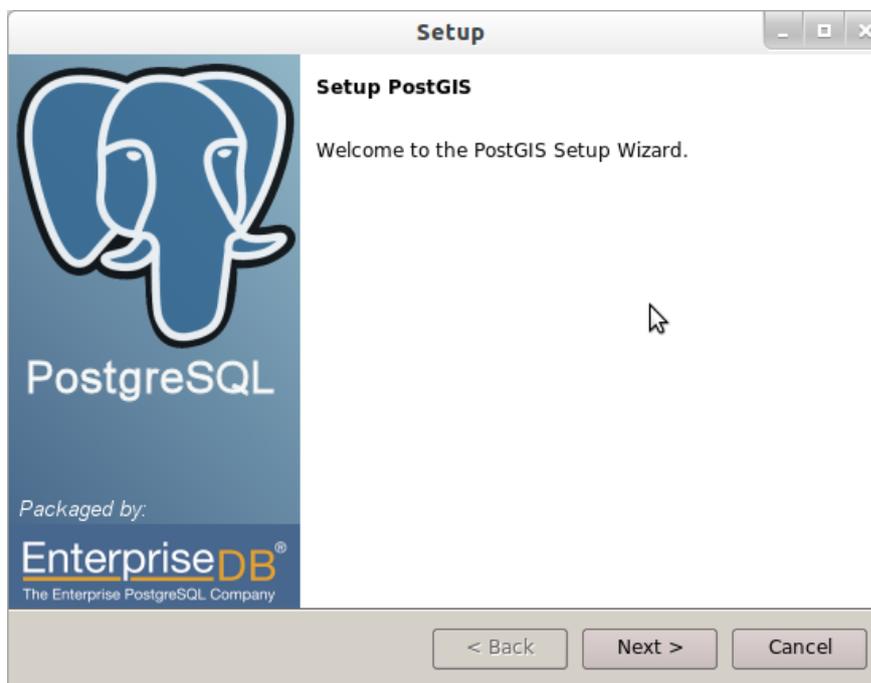
14. Una vez hecho el asistente empezará a descargar el instalador de PostGIS.



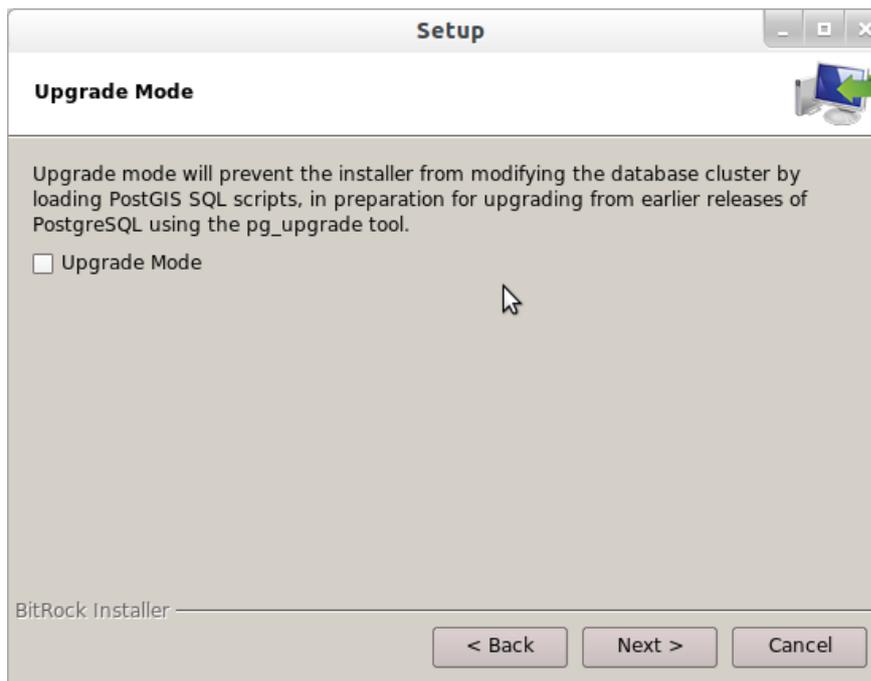
15. Una vez descargado el instalador el asistente pregunta si se desea saltar la instalación de las extensiones seleccionadas. Como se quiere instalar todas las extensiones, se desmarca esta opción y se da un clic en siguiente.



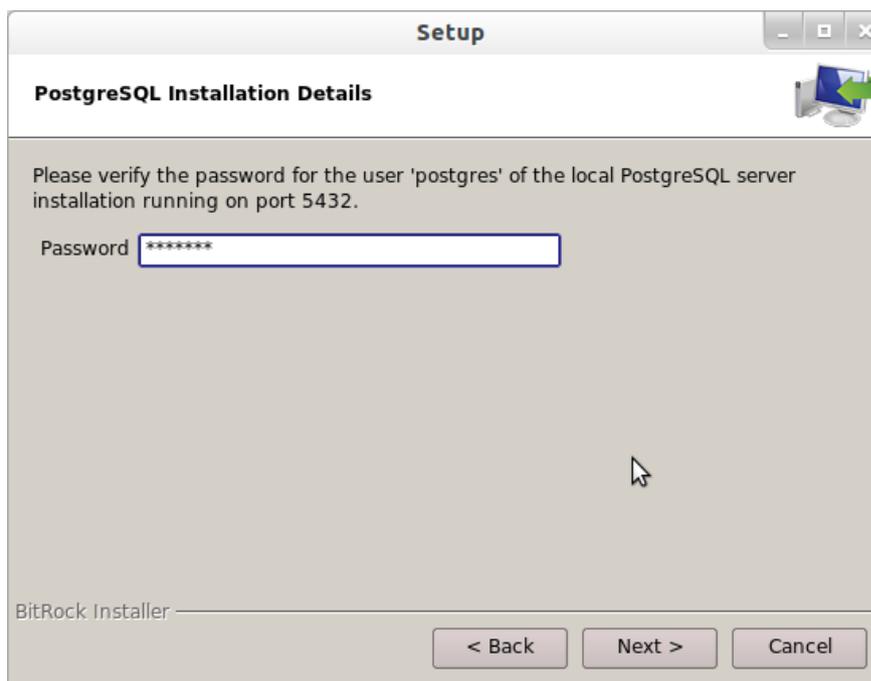
16. Inicialará la instalación de PostGIS.



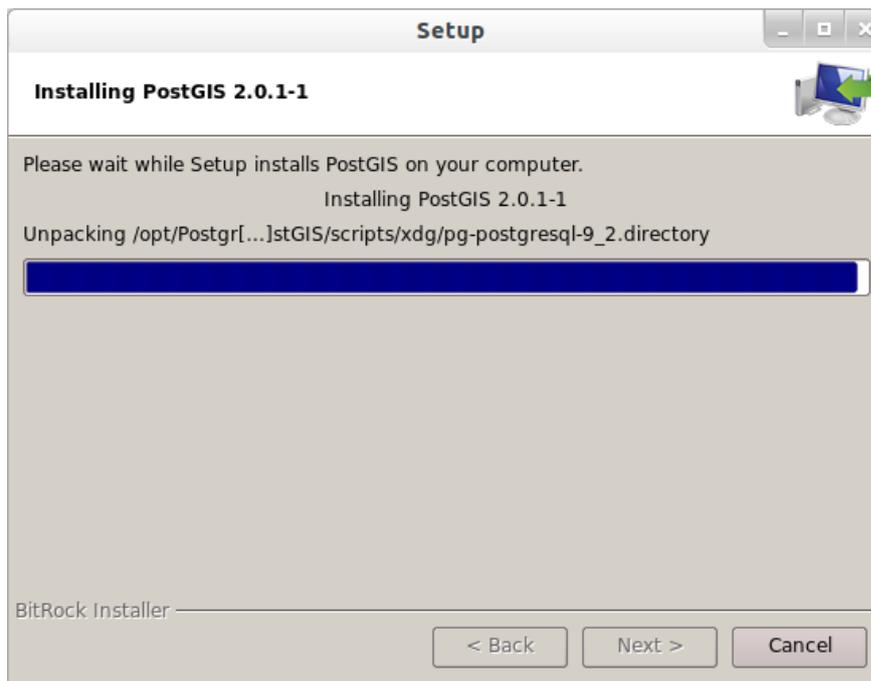
17. Deseleccionar el Upgrade Mode y dar clic en siguiente.



18. Ingresar el password previamente ingresado en PostgreSQL.

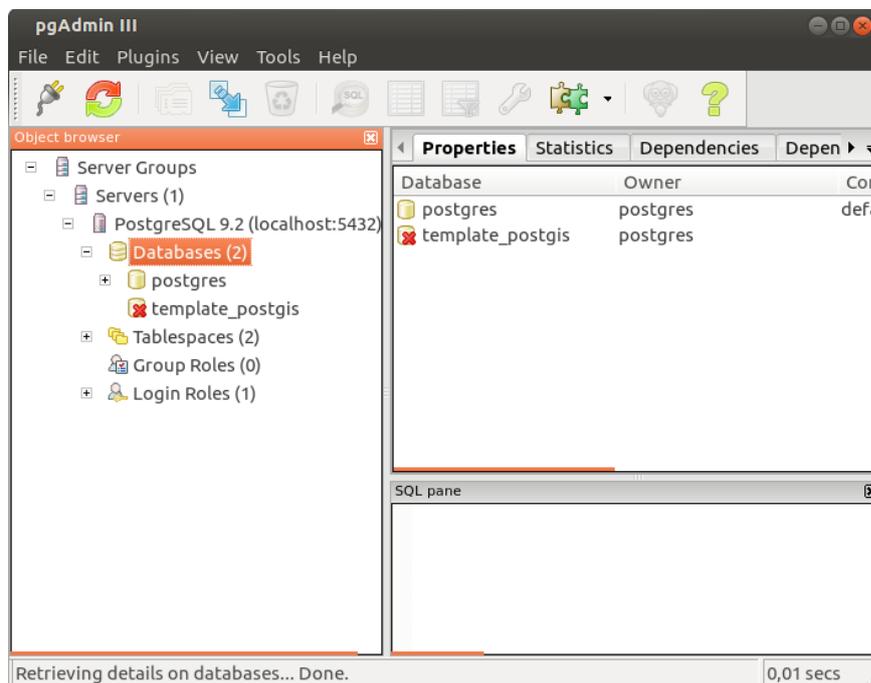


19. En las siguientes ventanas dar clic en siguiente y finalmente en terminar.



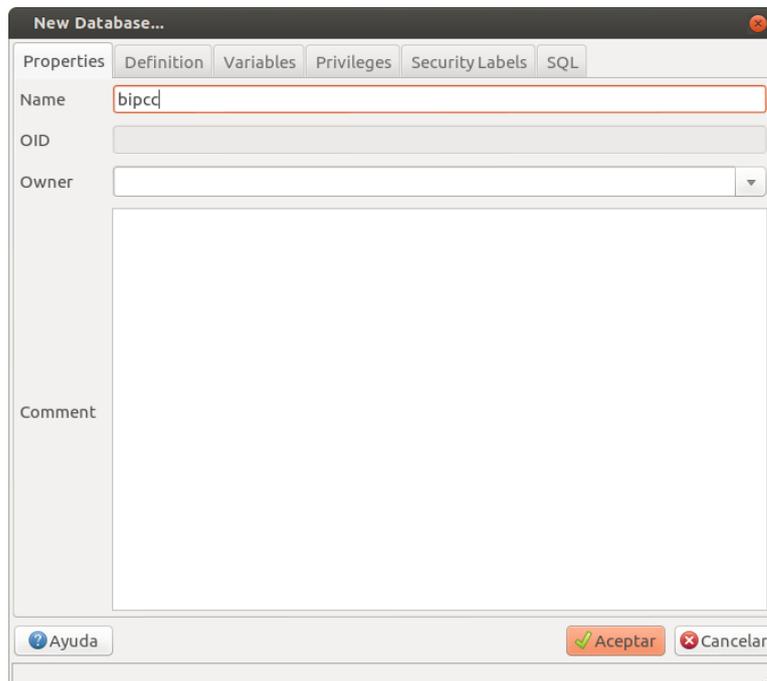
4.1.4. Pasos para la creación de la base de datos con un script previamente desarrollado

1. Ingresar en el administrador del bases de datos de PostgreSQL (PgAdmin III)



2. Dar clic derecho en Bases de Datos y seleccionar nueva New Database...

3. Ingresar el nombre de la base de datos, que en este caso es bipcc y dar clic en aceptar.

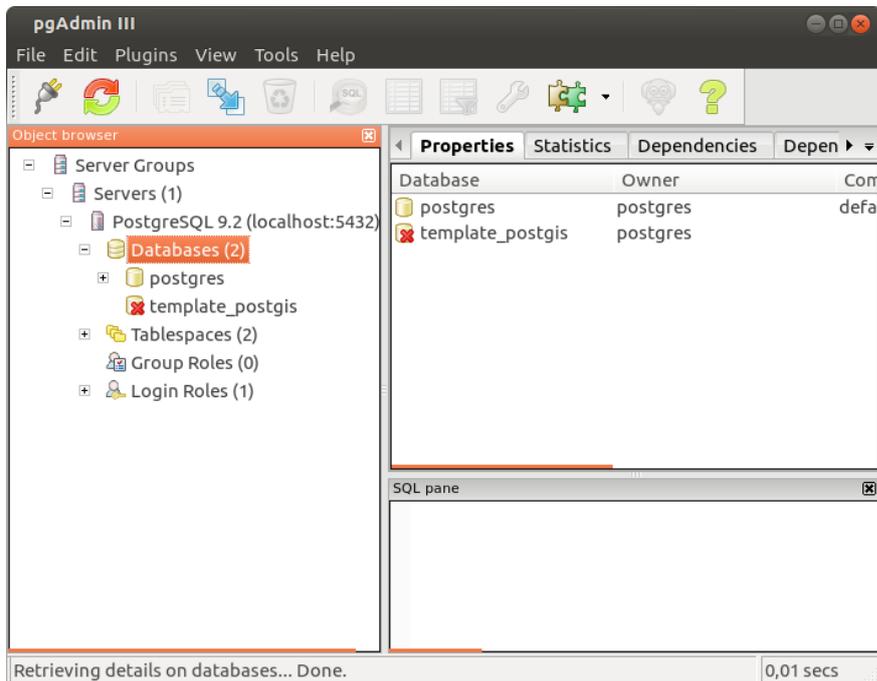


4. Dirigirse a la carpeta donde se encuentra el .sql con script de la base de datos y correr la siguiente línea:

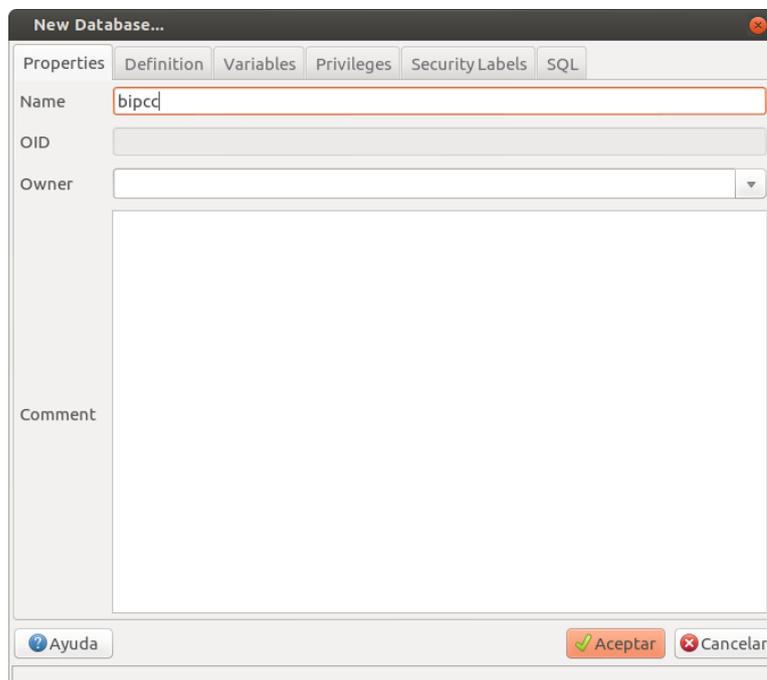
```
/opt/PostgreSQL/9.2/bin/pg_restore -i -U postgres -d bipcc -v bipcc.dump
```

4.1.5. Pasos para la creación de la base de datos desde cero

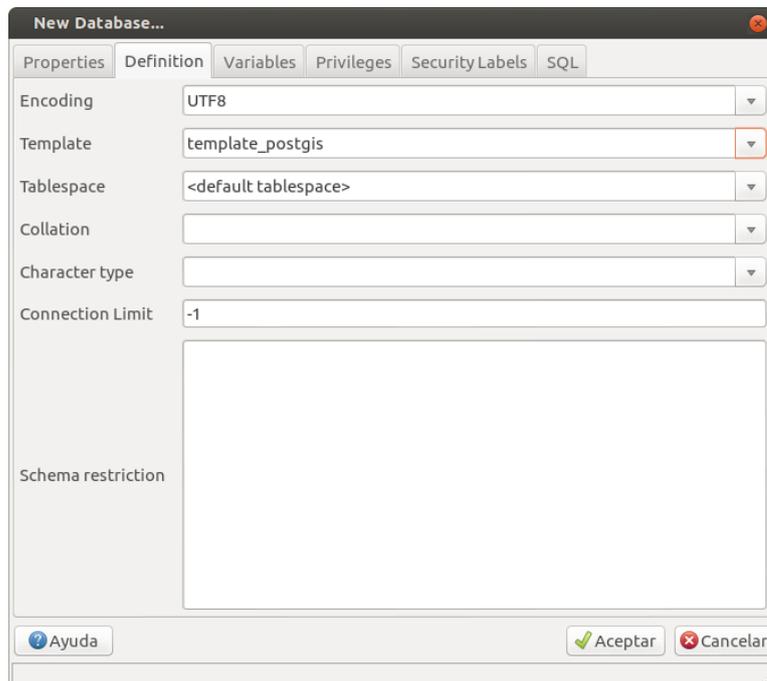
1. Ingresar en el administrador del bases de datos de PostgreSQL (PgAdmin III)



2. Dar clic derecho en Bases de Datos y seleccionar nueva New Database...
3. Ingresar el nombre de la base de datos, que en este caso es bipcc.



4. Seleccionar el Tab de definición, en la opción de template seleccionar `template_postgis`



5. Dar clic en aceptar y comenzar a crear la base de datos.

4.2. Instalación y configuración del Servidor de mapas

El servidor de mapas elegido en el análisis es MapServer 6.0.2, el mismo que será instalado sobre la plataforma Ubuntu 12.10 (Linux).

4.2.1. Requerimientos de software

- Librería GDAL. Una biblioteca de software para la lectura y escritura de formatos de datos geoespaciales.

Nota: La librería GDAL está disponible en un paquete llamado FWTools, el mismo que está disponible en <http://fwtools.maptools.org>.

4.2.2. Requerimientos de hardware

- Un procesador Core 2 Duo de 2 Ghz o superior.
- 2 GB de memoria RAM o más.
- 20 GB o más de espacio de almacenamiento en el disco duro.

4.2.3. Pasos para la instalación

1. Actualizar la cache de paquetes de Ubuntu con la siguiente línea de código:

```
sudo apt-get update
```

2. Descargar e instalar las MapServer mediante la siguiente línea de código:

```
sudo apt-get install cgi-mapserver mapserver-bin mapserver-doc php5-  
mapscript python-mapscript
```

Para probar que se haya instalado correctamente MapServer, se ejecuta el comando

```
/usr/lib/cgi-bin/mapserv -v
```

El resultado de la de la ejecución de este comando debe ser el siguiente:

```
MapServer version 5.6.5 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG  
OUTPUT=WBMP OUTPUT=SWF OUTPUT=SVG SUPPORTS=PROJ  
SUPPORTS=AGG SUPPORTS=FREETYPE SUPPORTS=ICONV  
SUPPORTS=FRIBIDI SUPPORTS=WMS_SERVER  
SUPPORTS=WMS_CLIENT SUPPORTS=WFS_SERVER  
SUPPORTS=WFS_CLIENT SUPPORTS=WCS_SERVER
```

```
SUPPORTS=SOS_SERVER SUPPORTS=FASTCGI
SUPPORTS=THREADS SUPPORTS=GEOS SUPPORTS=RGBA_PNG
INPUT=EPPL7 INPUT=POSTGIS INPUT=OGR INPUT=GDAL
INPUT=SHAPEFILE
```

3. Instalar el servidor web Apache y agregar el usuario respectivo con la siguientes líneas de comandos:

```
sudo apt-get install apache2
adduser apache
```

4. Para Configurar el directorio de publicación de MapServer se debe alterar un archivo con el siguiente comando:

```
sudo gedit /etc/apache2/sites-available/default
```

Este archivo define las reglas y permisos de acceso a los archivos de los diferentes sitios web que harán uso de MapServer.

En el caso del presente caso de estudio, cuando se hace referencia a “NOMBRE_PC” se trata del nombre que tiene el equipo en la red a la que está conectado, el archivo de configuración deberá quedar como a continuación:

```
<VirtualHost *:80>
    ServerName NOMBRE_PC
    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
```

```

<Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
# Possible values include: debug, info, notice, warn,
# error, crit, alert, emerg.
LogLevel warn
CustomLog ${APACHE_LOG_DIR}/access.log combined
Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
</VirtualHost>

```

5. Resetear el servidor apache con el comando:

```
sudo /etc/init.d/apache2 restart
```

6. Para que el proyecto funcione se debe copiar los archivos de configuración como MapFile, imágenes, páginas, etc. en la ruta /var/www.

4.3. Instalación y configuración de PrimeFaces Mobile

La plataforma sobre la cual se desarrollará el sistema es Ubuntu 12.10 (Linux), por esta razón todos los requerimientos de software deben ser descargados para ser instalados sobre dicha plataforma. La versión de PrimeFaces a instalar es la 3.4.2 y de PrimeFaces Mobile es la 0.9.3.

Debido a que se utilizará a Jboss Application Server como servidor de aplicaciones, los requerimientos de hardware están basados sobre el mismo.

4.3.1. Requerimientos de software

- JDK 1.4 update 45
- Eclipse Indigo
- Jboss Application Server 7.1.1
- Jboss Seam
- Jboss Tools 3.3.2
- PrimeFaces 3.4.2

4.3.2. Requerimientos de hardware

- Un procesador Core 2 Duo de 2 Ghz o superior.
- 2 GB de memoria RAM o más.
- 20 GB o más de espacio de almacenamiento en el disco duro.

Para facilitar la instalación de los componentes, todos los instaladores deben ser copiados en una carpeta, en este caso será la carpeta Desarrollo dentro de la carpeta Home del usuario actual de Ubuntu. A continuación se detallan los pasos para instalación y configuración de cada uno de los requerimientos de software:

4.3.3. JDK 1.6 update 45

1. Descargar el instalador con la extensión .bin de la página:

<http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR>

2. Conceder permisos de ejecución al instalador de JDK

```
chmod +x jdk-6u37-linux-i586.bin
```

3. Correr el instalador del JDK

```
sudo ./jdk-6u45-linux-i586.bin
```

4. Se creara una carpeta jdk1.6.0_45 en el directorio actual, moverla a la ruta convencional /usr/lib/jvm/

```
sudo mv jdk1.6.0_45 /usr/lib/jvm/
```

5. En versiones más recientes de Linux se utiliza variables estándar para la ubicación del JDK, para definir las usar los comandos:

```
sudo update-alternatives --install /usr/bin/javac javac  
/usr/lib/jvm/jdk1.6.0_45/bin/javac 1
```

```
sudo update-alternatives --install /usr/bin/java java  
/usr/lib/jvm/jdk1.6.0_45/bin/java 1
```

6. Se debe escoger los ejecutables definidos con las líneas anteriores como los por defecto a utilizar

```
sudo update-alternatives --config javac  
sudo update-alternatives --config java
```

7. Probar la correcta instalación desde la línea de comandos con:

```
java -version
```

Debe obtenerse el siguiente resultado:

```
java version "1.6.0_45"  
Java(TM) SE Runtime Environment (build 1.6.0_45)  
Java HotSpot(TM) Server VM (build 20.8-b03, mixed mode)
```

8. Crear las variables de entorno para JAVA_HOME y ANT_HOME, para ello abrir el archivo profile con:

```
sudo gedit /etc/profile
```

Agregar al final los comandos de exportación y uso de esas respectivas variables como a continuación:

```
export JAVA_HOME=/usr/lib/jvm/jdk1.6.0_45  
export PATH=$JAVA_HOME/bin:$PATH
```

9. Guardar el archivo y a continuación abrir una consola de comandos, en caso de que en dicha consola saliera algún error es porque hay errores de sintaxis en las líneas alteradas.

4.3.4. Eclipse

1. Descargar en .tar.gz de la página:

<http://www.eclipse.org/downloads/>

2. Descomprimir Eclipse

```
tar -xzvf eclipse-jee-helios-linux-gtk-x86_64.tar.gz
```

3. Moverlo a la ruta destino /opt

```
sudo mv eclipse /opt
```

4.3.5. JBoss Tools

1. Descargar en .zip de la página:

<http://www.jboss.org/jbossas/downloads/>

2. Descomprimir

```
unzip jboss-5.1.0.GA-jdk6c.zip
```

3. Moverlo a la ruta destino /opt

```
sudo mv jboss-5.1.0.GA /opt
```

4.3.6. JBoss (Application Server)

1. Descargar en .tar.gz de la página:

<http://seamframework.org/Download>

2. Descomprimir

```
tar -xzvf jboss-seam-2.3.0.GA.tar.gz
```

3. Moverlo a la ruta destino /opt

```
sudo mv jboss-seam-2.3.0.GA /opt
```

4.3.7. Jboss Tools 3.3.2

1. Descargar en .zip de la página:

<https://www.jboss.org/tools/download/stable.html>

2. Abrir Eclipse Indigo y seleccionar una carpeta Workspace, que es donde se guardarán todos los proyectos a realizar.
3. Seleccionar la opción Workbech
4. En el menú Help, seleccionar la opción Install New Software.
5. En la pantalla de asistente de configuración, pulsar el botón "Add", en Name poner "JBoss Tools 3.3.2 GA File", y en location ubicar el archivo descargado.
6. Aceptar la licencia, pulsar "Finish" y cuando termine reiniciar Eclipse.

4.3.8. PrimeFaces y PrimeFaces Mobile

1. Descargar la librería de la página

<http://www.primefaces.org/downloads.html>

2. Luego de creado el proyecto, es necesario agregar las librerías de PrimeFaces tomando primeramente el .jar:

```
primefaces-3.4.2.jar  
primefaces-mobile-0.9.3.jar
```

3. Copiar los .jar en la carpeta lib del proyecto
4. Abrir el archivo deployed-jars.list del proyecto y agregar al final las siguientes líneas:

```
primefaces-3.4.2.jar  
primefaces-mobile-0.9.3.jar
```

5. Dar un clic derecho en la raíz del proyecto y seleccionar la opción Propiedades.

6. Seleccionar la opción Java Buld Path.
7. En el Tab Libraries, seleccionar Add JARs...
8. Seleccionar las librerías previamente copiadas y presionar aceptar.
9. En el Tab Order and Exports, seleccionar las librerías previamente importadas
10. Dar clic en OK.

Capítulo V

Manual de desarrollo

En creación del sistema para Inventario de Bienes Inmuebles de la ciudad de Cuenca, se desarrollaron varios componentes. Se tomarán en cuenta solo los componentes más importantes para la construcción de un manual que permita al lector del presente caso de estudio saber cómo desarrollarlos de una manera rápida y fácil.

5.1.Creación de un nuevo proyecto en Jboss Seam 2.3

Al instalar cualquier versión de Jboss Seam, trae un asistente de configuración con el cual se puede crear un proyecto con una conexión directa a una base de datos, que en este caso es PostgreSQL. Este asistente crea un proyecto nuevo que puede ser abierto con Eclipse e utiliza la especificación JPA, la misma que utiliza la implementación Hibernate como framework de persistencias. Para crear un nuevo proyecto se deben seguir los siguientes pasos:

1. Lanzar el asistente de configuración de Jboss Seam.

```
sh /opt/jboss-Seam-2.3.GA/seam setup
```

2. Ingresar los siguientes parámetros en el asisten:

Base de Datos: postgres

Ruta driver JDBC: /opt/jboss-as-7.1.1.Final/standalone/lib/postgresql-8.3-603.jdbc3.jar

Dialecto Hibernate: org.hibernate.dialect.PostgreSQLDialect

JDBC Class Driver: org.postgresql.Driver

JDBC DataSource Class: org.postgresql.jdbc3.Jdbc3ConnectionPool

JDBC URL: jdbc:postgresql://localhost:5432/nombre_base

Usuario: postgres

Clave: clave_postgres

Esquema BD: public

Catalogo BD: nombre_base

Utilizar tablas desde BD: y

Usar script import.sql: n

3. Crear el proyecto.

```
sh seam create-project
```

4. Existe la posibilidad de generar los mantenimientos de todas las tablas en vistas prediseñadas o únicamente generar los archivos de persistencias con los siguientes comandos respectivamente:

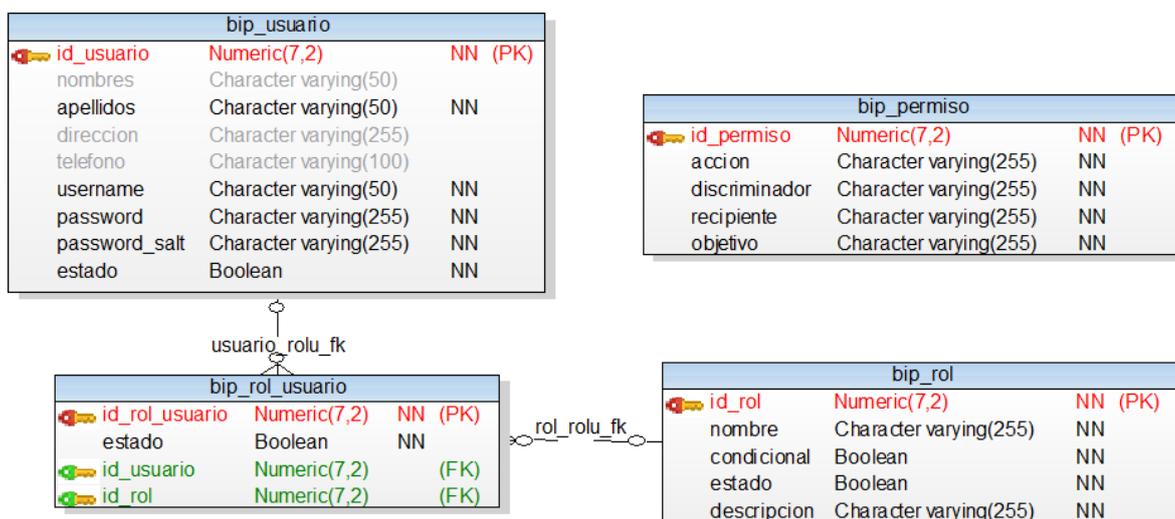
```
sh seam generate
```

```
sh seam generate-model
```

5. Una vez generado el proyecto, únicamente es necesario importarlo en Eclipse como proyecto ya existente.

5.2. Configuración de la seguridad en Jboss Seam 2.3

Seam trae un estándar que debe ser implementado sobre la base de datos para el manejo de la seguridad. Según el nivel de seguridad que se pretenda conseguir, existen modelos estándar a seguir. Para el sistema desarrollado, se utilizó un password con encriptación sha1 con una cadena de verificación md5. Se utiliza la siguiente estructura de base de datos:



Luego de generadas las persistencias con la estructura anterior, es necesario agregar en ellas las etiquetas necesarias para implementar los métodos de seguridad, además de especificar en el archivo components.xml dichas persistencias.

BipPermiso

```

@PermissionAction
//@Column(name = "accion")
public String getAccion() {
    return this.accion;
}

@PermissionDiscriminator
//@Column(name = "discriminador")
public String getDiscriminador() {
    return this.discriminador;
}

@PermissionUser
@PermissionRole
//@Column(name = "recipiente")
public String getRecipiente() {
    return this.recipiente;
}

@PermissionTarget
//@Column(name = "objetivo")
public String getObjetivo() {
    return this.objetivo;
}

```

BipUsuario

```

@UserFirstName
@Column(name = "nombres", nullable = false, length = 50)
@NotNull
@Size(max = 50)
public String getNombres() {
    return this.nombres;
}

@UserLastName
@Column(name = "apellidos", nullable = false)
@NotNull
public String getApellidos() {
    return this.apellidos;
}

@UserPrincipal
@Column(name = "username", unique = true, nullable = false, length = 50)
@NotNull
@Size(max = 50)
public String getUsername() {
    return this.username;
}

```

```

}
@UserPassword
@Column(name = "password", nullable = false)
@NotNull
public String getPassword() {
    return this.password;
}
@PasswordSalt
@Column(name = "password_salt", nullable = false)
@NotNull
public String getPasswordSalt() {
    return this.passwordSalt;
}

@UserEnabled
@Column(name = "estado", nullable = false)
public boolean isEstado() {
    return this.estado;
}
@UserRoles
@OneToMany(fetch = FetchType.EAGER, mappedBy = "bipUsuario")
public Set<BipRolUsuario> getBipRolUsuarios() {
    return this.bipRolUsuarios;
}
}

```

BipRol

```

@RoleName
@Column(name = "nombre", nullable = false)
@NotNull
public String getNombre() {
    return this.nombre;
}
@RoleConditional
@Column(name = "condicional", nullable = false)
public boolean isCondicional() {
    return this.condicional;
}
@RoleGroups
@OneToMany(fetch = FetchType.LAZY, mappedBy = "bipRol")
public Set<BipRolUsuario> getBipRolUsuarios() {
    return this.bipRolUsuarios;
}
}

```

components.xml

```

<drools:rule-base name="securityRules">
    <drools:rule-files><value>/security.drl</value></drools:rule-files>
</drools:rule-base>

```

```

<security:rule-based-permission-resolver security-rules="#{securityRules}"/>
<security:identity authenticate-method="#{authenticator.authenticate}" remember-me="true"/>
<security:identity-manager
    identity-store="#{jpaIdentityStore}"
    role-identity-store="#{jpaIdentityStore}"
/>
<security:jpa-identity-store
    user-class="com.bipcc.models.BipUsuario"
    role-class="com.bipcc.models.BipRol"
/>
<security:jpa-permission-store user-permission-class="com.bipcc.models.BipPermiso"/>

```

5.3. Parametrización de una búsqueda utilizando MapServer

A continuación se presentan los requerimientos mínimos que tendría un MapFile para poder implementar una búsqueda sobre la base de datos alfanumérica. Los resultados de la búsqueda son reflejados mediante la implementación de un servicio WMS. Tomar en cuenta que el valor del atributo `consulta_validation_pattern` de la sección `metada` del `layer bip_bien_inmueble` debe tener el valor `^[a-zA-Z0-9%\,]+$` para agregar una máscara al valor de la cadena de búsqueda y evitar la posibilidad de ejecutar un SQL Injection²⁹.

MapFile

```

NAME "BIPCC"
SIZE 640 432
UNITS meters
EXTENT 128087.036757 9427774.141962 1518304.553312 10178984.197660
FONTSET 'fonts.txt'
SYMBOLSET 'symbols.txt'
PROJECTION
    'proj=utm'
    'zone=17'
    'south'
    'datum=WGS84'
    'units=m'
    'no_defs'
END
IMAGECOLOR 255 255 255
IMAGEQUALITY 95
IMAGETYPE png
OUTPUTFORMAT
    NAME png
    DRIVER 'AGG/PNG'
    MIMETYPE 'image/png'
    IMAGEMODE RGBA

```

²⁹ SQL Injection: Técnica aplicada para alterar o atacar sitio o servidor mediante inserción de comandos SQL adicionales al comando original.

```

EXTENSION 'png'
END
LEGEND
  IMAGECOLOR 255 255 255
STATUS ON
KEYSIZE 18 12
LABEL
  TYPE BITMAP
  SIZE MEDIUM
  COLOR 0 0 89
END
END
WEB
  IMAGEPATH '/tmp/'
  IMAGEURL '/tmp/'
METADATA
  'wms_title'          'Bienes Inmuebles Patrimoniales'
'wms_srs'              'EPSG:32717'
  'wms_enable_request' '*'
  'queryable'         'true'
  'wms_feature_info_mime_type' 'text/html'
END
  TEMPLATE 'fooOnlyForWMSGetFeatureInfo'
END
LAYER
  NAME 'bip_bien_inmueble'
  TYPE POINT
  DUMP true
  TEMPLATE fooOnlyForWMSGetFeatureInfo
  EXTENT 76053.663826 9427774.141962 1570337.926242 10178984.197660
  CONNECTIONTYPE postgis
  CONNECTION "dbname='bipcc' host=localhost port=5432 user='postgres' password='phantom'
  sslmode=disable"
  DATA 'geom from (select p.id_bien_inmueble as gid, p.geom as geom, p.clave_catastral as
clave_catastral, p.denominacion as denominacion, p.propietario_actual
as propietario_actual, p.descripcion as descripcion from
bip_bien_inmueble p ) as subquery USING UNIQUE gid USING srid=32717'
  FILTER (lower(clave_catastral) like '%%consulta%%' or lower(denominacion) like
'%%consulta%%' or lower(propietario_actual) like '%%consulta%%' or
lower(descripcion) like '%%consulta%%' or gid in (SELECT DISTINCT
bi.id_bien_inmueble AS ID_BIEN FROM bip_valor as v, bip_valor_bien_inmueble as
vb, bip_bien_inmueble as bi WHERE vb.id_bien_inmueble = bi.id_bien_inmueble AND
vb.id_valor = v.id_valor AND (LOWER(v.descripcion) LIKE LOWER('%%CONSULTA%%')
OR LOWER(vb.valor) LIKE LOWER('%%CONSULTA%%'))))
METADATA
  'ows_title'          'Bienes Inmuebles Patrimoniales'
  'wfs_title'          'Bienes Inmuebles Patrimoniales'
  'wfs_srs'            'EPSG:32717'
'gml_include_items' 'all'
  'gml_featureid'     'GID'

```

```
'wfs_enable_request' '*'  
'consulta_validation_pattern' '^[a-zA-Z0-9%], ]+$'  
  
'default_consulta' '%'  
END  
STATUS OFF  
TRANSPARENCY 100  
PROJECTION  
'proj=utm'  
'zone=17'  
'south'  
'datum=WGS84'  
'units=m'  
'no_defs'  
END  
TEMPLATE 'bip.html'  
CLASS  
  NAME 'bip_bien_inmueble'  
  STYLE  
    SYMBOL "bip"  
    SIZE 24  
    OUTLINECOLOR 0 0 0  
    COLOR 77 158 104  
  END  
END  
END
```

Una vez Implementado el MapFile se utiliza la siguiente URL para probar que se estén filtrando y mostrando los datos mediante WMS.

http://localhost/cgi-bin/mapserv?map=/var/www/bipcc/bipcc2.map&service=wms&VERSION=1.1.1&REQUEST=GETMAP&layers=egr_provincia,bip_bien_inmueble&srs=EPSG:32717&bbox=76053.663826,9427774.141962,1570337.926242,10178984.197660&FORMAT=image/png&width=1280&height=700&consulta=cadena_busqueda

5.4. Recuperación de metadatos de una Geometría

Para la recuperación de metadatos es necesario crear un archivo de template con extensión .html especificando los metadatos a extraer desde el query ejecutado en el layer respectivo. En el caso del sistema desarrollado, se toma el campo gid del query realizado en el MapFile. El gid representa la clave primaria con la que se puede obtener una geometría en específico.

bip.html

```
<!-- MapServer Template -->
[gid]
```

Debe implementarse un procedimiento que tome la URL de la cual se realiza el pedido para la extracción de metadatos. Dicho pedido puede ser enviado desde un layer implementado por OpenLayers en el momento en el cual se da un clic sobre el mapa. Luego de obtener el id del bien inmueble, se tiene acceso a todos los datos del mismo.

Mapa.java

```
public void cargarURLInfo()
{
    setPredioMetaDatos(new BipBienInmueble());
    setComandoAbrirMatadatos("");
    log.info("URL destino: "+getUrlInfo());
    String valor = "";
    try
    {
        URL google = new URL(getUrlInfo());
        URLConnection yc = google.openConnection();
        BufferedReader in = new BufferedReader(new
            InputStreamReader(yc.getInputStream()));
        String inputLine;
        while ((inputLine = in.readLine()) != null)
        {
            log.info(inputLine);
            valor = inputLine;
        }
        in.close();
        if (!valor.trim().equalsIgnoreCase(""))
        {
            int idPredio = Integer.parseInt(valor);
            log.info("Numero a consultar: [{"+idPredio}]",idPredio);
            String sqlPredioMeta = "SELECT P FROM BipBienInmueble P WHERE
P.idBienInmueble="+idPredio;
            log.info(sqlPredioMeta);
            Query queryPredioMeta = entityManager.createQuery(sqlPredioMeta);
```

```

BipBienInmueble p = (BipBienInmueble)queryPredioMeta.getSingleResult();
if (p!=null)
{
    setPredioMetaDatos(p);
    cargarFotos(p.getIdBienInmueble());
    setComandoAbrirMatadatos("abrirVentanaMetadatos;");
}
}
}
catch (Exception e)
{
    e.printStackTrace();
}
}

```

5.5. Agregar una geometría de puntos a una tabla en PostGIS

No existe un asistente para agregar directamente una geometría dentro de una tabla, por lo cual debe ser realizado mediante una instrucción SQL:

```

SELECT AddGeometryColumn
    ('nombre_tabla',
     'nombre_geometría',
     sistema_referencia,
     'tipo_geometría',
     numero_dimensiones);

```

Para el sistema a desarrollar se utilizará una geometría tipo POINT con el sistema de referencia WGS84 UTM 32717 [1]zona 17 del suren la tabla bip_bien_inmueble:

```

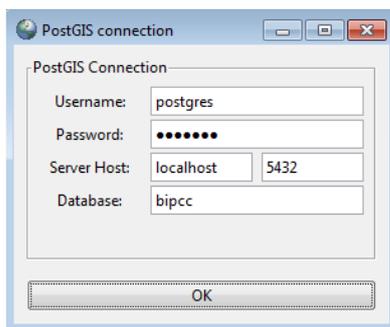
SELECT AddGeometryColumn
    ('bip_bien_inmueble',
     'geom',
     32717,
     'POINT',
     2);

```

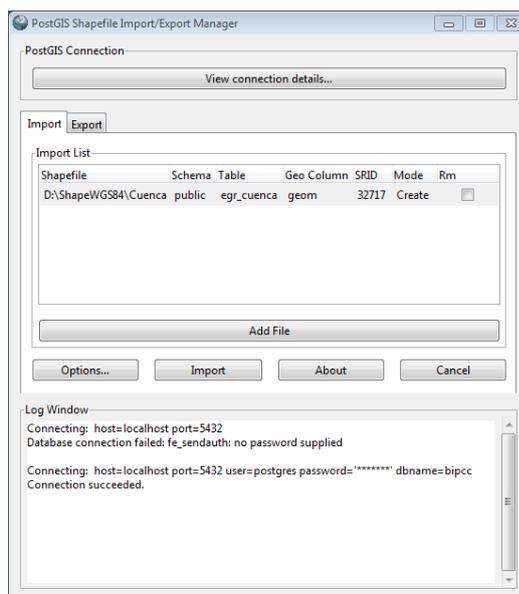
5.6.Importar archivos shape hacia PostGIS

PostGIS trae una herramienta propia para la importación de archivos shape hacia la base de datos. PostGIS 2.0 shapefile and DBF Loader Exporter es capaz de tomar un shape en un cierto sistema de referencia e importarlo como si fuera una tabla de la base de datos con columnas geoespaciales y alfanuméricas. Para realizar esta acción se siguen los siguientes pasos:

1. Abrir PostGIS 2.0 shapefile and DBF Loader Exporter.
2. Seleccionar la Opción View Connection Details e ingresar los detalles de la conexión a la base de datos.



3. Seleccionar el Tab Import
4. Clic en Add File y seleccionar el archivo shape
5. Al salir el archivo en el listado de importación, es necesario cambiarle los atributos Table para el nombre de la tabla y SRID (32717 [1] en este caso) para el sistema de referencia a utilizar.



6. Clic en Import

CONCLUSIONES

Todos los objetivos propuestos al inicio del presente caso de estudio han sido cumplidos de forma satisfactoria y durante su desarrollo se ha llegado a las siguientes conclusiones:

- Se ha logrado desarrollar una aplicación SIG móvil basada en web mediante PrimeFaces Mobile para el manejo información Bienes Inmuebles Patrimoniales de la ciudad de Cuenca. Dicha aplicación tiene módulos destinados a dispositivos móviles para el levantamiento de información y módulos web para equipos de escritorio o portátiles encargados de la administración.
- Se ha creado una aplicación SIG destinada para dispositivos móviles, la misma que se encarga del ingreso directamente desde el campo de la información de los Bienes Inmuebles. Debido a las limitaciones que tienen los dispositivos móviles en cuanto al ingreso de información vía texto, se ha dado más énfasis a la utilización de capacidades como posicionamiento vía GPS y toma de fotos. Los datos de texto ingresados desde el dispositivo son pocos y serán posteriormente aumentados mediante una aplicación de administración para agilizar la alimentación de SIG.
- Se construyó una aplicación web de administración destinada para navegadores de equipos de escritorio o portátiles. Esta aplicación es capaz de extender la información que es ingresada mediante el dispositivo móvil. Se creó también una interfaz dinámica mediante la cual el usuario tiene la posibilidad de agregar más campos descriptivos a la base datos y llenarlos para cada bien inmueble, por ejemplo si posteriormente se necesita ingresar la fecha de venta del bien inmueble, únicamente se debe especificar en qué categoría y subcategoría de información está el campo a ingresar y este será dinámicamente agregado a las características adicionales de todos los bienes inmuebles.
- La exposición de los datos hacia el público es vital para este tipo de aplicaciones, por lo cual se ha desarrollado una interfaz con la cual se puede visualizar en un mapa de la ciudad de Cuenca todo el inventario de bienes inmuebles mediante la implementación de un servicio WMS con MapServer y OpenLayers. Además brinda la posibilidad de un filtrado tanto a nivel de registros (tabla de coincidencias) como a nivel del mapa para posteriormente poder extraer la información del inmueble realizando una llamada a la función de extracción de metadatos de MapServer.
- Para realizar un aporte a la comunidad de desarrollo se ha realizado un manual con las partes cruciales necesarias para el desarrollo de aplicaciones SIG móviles. No se ha profundizado demasiado en el manual debido a lo extenso de las herramientas, pero explica de la mejor manera como comenzar.

- Al realizar un análisis de la librería PrimeFaces Mobile y una comparación con tecnologías similares, se determinó que es una de las más fuertes del mercado y con mayor flexibilidad al momento del diseño de interfaz.
- Para implementar la librería PrimeFaces Mobile, es necesario tener conocimientos sobre tecnologías como Java Server Faces, el paradigma de programación MVC, manejo de frameworks de persistencias, entre otros.
- Actualmente muchos gobiernos en Latinoamérica tienden a eliminar el licenciamiento y preferir tecnologías de código abierto o freeware. PrimeFaces Mobile y todas las herramientas utilizadas cumplen estas expectativas.
- Las características de hardware que tienen los dispositivos móviles como posicionamiento GPS, cámara de fotos, portabilidad, posibilidades de acceso a internet, etc., brindan al desarrollador grandes posibilidades para la creación de aplicaciones que los aprovechen.
- PrimeFaces Mobile tiene problemas de compatibilidad con algunas librerías como RichFaces, APIs de Google, IceFaces. Todo esto es debido a la forma en que implementan AJAX dentro de la vista o a la diferencia de versiones de JQuery que utilizan.
- Existen ciertos dispositivos con los que PrimeFaces Mobile no es compatible. Funciona de mejor manera con pantallas táctiles.
- Existen problemas de compatibilidad con los GPS de ciertos dispositivos móviles, por lo que en los dispositivos no compatibles esta opción no estará disponible.

RECOMENDACIONES

Al momento del desarrollo del presente caso de estudio se encontraron algunos aspectos a tomar en cuenta:

- Realizar un análisis minucioso de las herramientas que se va a utilizar. Todo esto se debe a la cantidad de recursos de procesamiento que tiene JSF y su servidor de aplicaciones, el tipo de compilación que tiene el lenguaje Java, compatibilidad entre tecnologías, etc.
- No solo centrarse en la librería de componentes, tecnologías o frameworks que se va a utilizar, sino también tomar en cuenta la geo base de datos que se va a utilizar y el servidor de mapas que va a interactuar con la misma.
- Diseñar correctamente la base de datos, ya que al momento de implementar sobre el sistema es más difícil realizar cambios y más aún si se realiza un mapeo de la base de datos sobre el proyecto.
- Tomar en cuenta que al momento del desarrollo del MapFile no es necesario crearlo desde cero en líneas de comandos, sino que hay herramientas como KOSMO, GVSIG, QuantumGIS, etc. que facilitan su creación y personalización.
- Verificar que en el diseño de interfaz de la aplicación móvil no se sobrecargue las vistas con demasiados controles, debido a que la página se renderiza según las propiedades de la pantalla del dispositivo móvil, lo cual puede interferir con el diseño. Se debe ser lo más genérico posible.
- Utilizar herramientas como JPA con Hibernate para la conexión e interacción con la base de datos. Estas herramientas facilitan enormemente el desarrollo y son totalmente compatibles con librerías de componentes como PrimeFaces.
- Tomar en cuenta que al extraer la posición desde el dispositivo móvil, esta vendrá por defecto en el sistema de coordenadas que trae Google por defecto (WGS84 UTM4326) y la base de datos geoespacial está en el sistema de coordenadas de Ecuador (WGS84 UTM32717 [1]). En este caso es necesario aplicar funciones de conversión.
- Revisar la compatibilidad de la información de los archivos Shape con la geo base de datos debido que estos traen información alfanumérica con una codificación de caracteres que puede ser diferente a la que se ha implementado en la base de datos local.

BIBLIOGRAFÍA

- ESRI. Ayuda de ArcGIS 10.1. s.f. Enero de 2013
 <<http://resources.arcgis.com/es/help/main/10.1/index.html#/na/00qn0000001p000000/>>.
- ESRI. «GeoDatabase.» 1995. Enero de 2013
 <<http://support.esri.com/en/knowledgebase/GISDictionary/search>>.
- ESRI. «SIG.» 1995. Enero de 2013
 <<http://support.esri.com/en/knowledgebase/GISDictionary/search>>.
- Herrera, Andres. «Geo Sin Enredos.» Enero de 2009. Marzo de 2013
 <<http://andresherreracali.blogspot.com/2009/01/geo-sin-enredos-guia-de-postgresql.html>>.
- ISO. Estándares ISO. s.f. Junio de 2012 <<http://www.iso.org/iso/home>>.
- Loor, José Miguel. «JSF Java Server Faces.» Octubre de 2009. Febrero de 2013
 <<https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CDgQFjAA&url=http%3A%2F%2Fwww.sicuma.uma.es%2Fsicuma%2Fformacion%2Fdocumentacion%2FJSF.pdf&ei=-QzoUb3iHZG09gTZmYCYBg&usg=AFQjCNE1uVSTsBgqtPEGWw64Mh-92Q1yg&bvm=bv.49478099,d.e>>.
- Moncayo, Diego Patricio. «Análisis y configuración de un SIG corporativo en la plataforma UNIX con software de la empresa ESRI para la Municipalidad de Cuenca.» Abril de 2008. Diciembre de 2012 <<http://dspace.ups.edu.ec/handle/123456789/56>>.
- Open GeoSpatial Consortium. Estándares de OGC. s.f. Diciembre de 2012
 <<http://www.opengeospatial.org/standards>>.
- Orellana, Manuel Gerardo. Tesis Evaluación de frameworks realizados en java para aplicaciones on-line. Biblioteca de la Universidad del Azuay. Cuenca, Cuenca, Mayo de 2013.
- Pavón, Juan. «Estructura de las Aplicaciones Orientadas a Objetos.» Septiembre de 2008. Febrero de 2013
 <<https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCoQFjAA&url=http%3A%2F%2Fwww.fdi.ucm.es%2Fprofesor%2Fjpavon%2Fpoo%2F2.14.mvc.pdf&ei=sRDoUZWcG4m68ATCoIGoDA&usg=AFQjCNER5YNC4fA80WbqOTglyBEhKye0dw&bvm=bv.49478099,d.eWU>>.
- PEM. «Metadatos.» Septiembre de 2011. Marzo de 2013
 <https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCoQFjAA&url=http%3A%2F%2Fwww.cp-idea.org%2Fdocumentos%2FMetadatos%2Fmetadatos_2011.pdf&ei=NRL0UdGkEpSC9Qsv2oFI&usg=AFQjCNH6_y1bNxJjp7kfxwv2kmgF5UU_bw&bvm=bv.49478099,d.eWU>.

PrimeFaces. «PrimeFaces 0.9.3 Mobile User's Guide.» Junio de 2012. Diciembre de 2012
<<http://www.primefaces.org/downloads.html>>.

PrimeFaces. «PrimeFaces 3.4.2 User's Guide.» Septiembre de 2012. Diciembre de 2012
<<http://www.primefaces.org/downloads.html>>.

Rodriguez, Juan. Tesis Análisis, Diseño e Implementación de un Sistema de Información Geográfico para Seguimiento de Egresados de la Universidad Politécnica Salesiana. Cuenca, Agosto de 2011.

SICUMA. «Tutorial de Java Server Faces.» Abril de 2007. Enero de 2013
<<https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCwQFjAA&url=http%3A%2F%2Fwww.sicuma.uma.es%2Fsicuma%2FFormacion%2Fdocumentacion%2FJSF.pdf&ei=CBLouYq7M4L68QSokoDIBQ&usg=AFQjCNEm1uVSTsBgqtPEGWw64Mh-92Q1yg&bvm=bv.49478099,d.e>>.

Spatial Reference. SpatialReference.org. s.f. Febrero de 2013
<<http://www.spatialreference.org/ref/epsg/32717/>>.

Tutoriales Aprende a Programar. Aprende a Programar. s.f.
<http://www.aprenderaprogramar.com/index.php?option=com_content&id=392:la-maquina-virtual-java-jvm-o-java-virtual-machine-compiler-e-interprete-bytecode-cu00611b&Itemid=188>.

ANEXOS

8. VALORACION DEL INMUEBLE: BAREMO

8.1 Antigüedad					Puntuación	
Prehispánica Hasta 1534	PH	15	PH	15		
Colonial Siglo XVI - XIX (1534-1822)	CO	15	CO	15	x	15
Republicana 1 - Siglo XIX (1822-1830)-1900	RP1	12	RP1	12		
Republicana 2 - Siglo (1901-1960)	RP2	9	RP2	9		
Republicana 3 - Siglo XIX-XXI (1961-2010)	RP3	3	RP3	3		
					15	

8.2 Estético Formal					Puntuación	
Identificación Estilística	IE	1 a 3	IE+CF	5	x	5
Comparación Formal	CF	1 a 2	IE+CF+AM	3		
Alteraciones Altas	AA	-3	IE+CF+AA	2		
Alteraciones Medias	AM	-1	IE+AM	2		
		-2	CF+AM	1		
					5	

8.3 Tipológico Funcional					Puntuación	
Conserva Identificación Tipológica	CIT	2 a 4	CIT+CUO	5		
Conserva Usa Original	CUO	1	CIT+CUO+AM	4	x	4
Nueva Usa	UN	-1	CIT+CUO+AA	2		
Nueva Usa No Compatible	NUNC	-2	CIT+AM	3		
Alteraciones Medias	AM	-1	CIT+AA	3		
Alteraciones Altas	AA	-3	CIT+NUNC	2		
			CIT+NUNC+AM	1		
			CIT+UN	3		
					4	

8.4 Técnico Constructivo					Puntuación	
Tecnología y Materiales Tradicionales	TMT	5	TMT	5		
Tecnología y Materiales Contemporáneos	TMC	3	TMT+ECR	4		
Tecnología y Materiales Mixtos	TMR	2	TMT+ECM	2	x	2
Estado de Conservación Regular	ECR	-1	TMT+SMTM	3		
Estado de Conservación Mala	ECM	-3	TMT+SMTA	1		
Sustitución, Materiales y Tecnología Alta	SMTA	-4	TMT+STMM+ECR	2		
Sustitución, Materiales y Tecnología Media	SMTM	-2	TMC	3		
			TMC+ECR	2		
			TMC+SMTM	1		
			TMR	2		
			TMR+ECR	1		
					2	

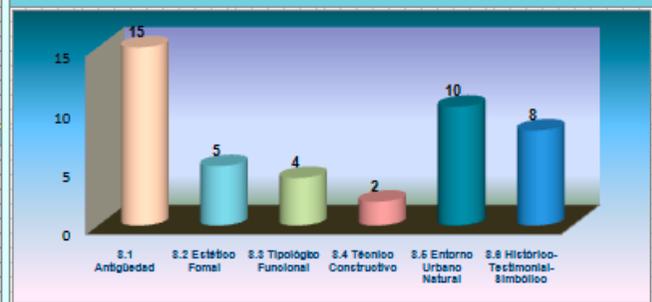
8.5 Entorno Urbano Natural					Puntuación	
Integrada Trama Homogénea con Valor	ITHOV	10	ITHOV	10	x	10
Integrada Trama Homogénea	ITHO	7	ITHO	7		
Trama Homogénea	THO	5	THO	5		
Trama Heterogénea	THE	2	THO+DT	8		
Distorsión Trama	DT	3	THO+HINT	2		
No Integrada al Trama	HINT	-3	THE+DT	5		
Integrada Entorno Natural	IEN	3	THE	2		
Conserva el Ambiente Utiliza Materiales Locales	CAUMT	2	IEN+CAUMT	5		
			IEN	3		
					10	

8.6 Histórico- Testimonial- Simbólico					Puntuación	
Valor Simbólico Socio Cultural (Identidad)	VHSC	10	VHSC	10		
Inmueble de Interés Histórico Testimonial	IHT	8	IHT	8	x	8
Hito Urbana	HU	6				
Autor Representativa	AR	4				
Distinción del Inmueble	DI	6				
Innovación Tecnológica Significativa	ITS	4				
					8	

8.7 VALORACIÓN Y GRADO DE PROTECCIÓN

Grupo	Tipo de Protección	Puntaje	44	
A	Absoluta	36 - 50	PROTECCION ABSOL	ALTO VALOR PATRIMONIAL
B	Parcial	26 - 35		
C	Condicionada	16 - 25		
D	Sin Protección	01 - 15		

8.8 GRÁFICO DE VALORACIÓN: Diagrama de barras



8.9 CRITERIOS DE VALORACIÓN

8.1 Antigüedad:	
8.2 Estético Formal:	
8.3 Tipológico Funcional:	
8.4 Técnico Constructivo:	
8.5 Entorno Urbano-Natural:	
8.6 Histórico- Testimonial- Simbólico:	

8.10 RESPONSABLES VALORACION

Entidad Ejecutora:	
Elaborada por:	Fecha (DD/MM/AA):
Revisada por:	
Aprobada INPC:	

