

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**SISTEMA DE GUÍA MÓVIL PARA EL MUSEO NAVAL Y
MARITIMO DE VALPARAÍSO**

MIGUEL ANDRÉS ALCAÍNO AVILA

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN INFORMÁTICA

Carrera: **Ingeniería de Ejecución en Informática**

SEPTIEMBRE 2011

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**SISTEMA DE GUÍA MÓVIL PARA EL MUSEO NAVAL Y
MARÍTIMO DE VALPARAÍSO**

MIGUEL ANDRÉS ALCAÍNO AVILA

Profesor Guía: **José Rubio León**
Profesor Co-referente: **Iván Mercado**

Carrera: **Ingeniería de Ejecución en Informática**

SEPTIEMBRE 2011

Dedicatoria

A mi abuela y abuelo, por su apoyo y cariño incondicional. Especialmente a él, que siempre sintió gran alegría y orgullo por mí. Le debo mi gusto por el uso de las herramientas para inventar y crear, pilar esencial en mi vida.

Agradecimientos

Agradezco especialmente a mi querida novia, por su apoyo durante este proceso. Por el gran cariño que me ha entregado. Por ser mi inspiración en el valor de la responsabilidad.

Agradezco a mis abuelos y a mi hermano, por saber que siempre puedo contar con ellos.

Índice

| | |
|---|------|
| Resumen..... | viii |
| Abstract | ix |
| Lista de figuras | x |
| Lista de tablas..... | xi |
| 1 Introducción | 1 |
| 2 Descripción del proyecto..... | 2 |
| 2.1 Objetivos | 3 |
| 2.1.1 Objetivo general | 3 |
| 2.1.2 Objetivos específicos | 3 |
| 3 Marco Teórico | 4 |
| 3.1 Desarrollo en dispositivos móviles | 4 |
| 3.1.1 Smartphones | 5 |
| 3.1.2 PDA..... | 5 |
| 3.1.3 Plataformas de programación..... | 6 |
| 3.1.3.1 Java 2 micro edition | 6 |
| 3.1.3.2 .Net compact framework..... | 6 |
| 3.1.3.3 Android | 6 |
| 3.2 Sistemas operativos móviles | 7 |
| 3.2.1 Symbian OS | 7 |
| 3.2.2 Windows Phone | 7 |
| 3.2.3 Android | 8 |
| 3.2.4 iOS | 8 |
| 3.2.5 Palm OS | 9 |
| 3.3 MonGuive Louvre | 9 |
| 4 Museo Naval y Marítimo de Valparaíso | 11 |
| 5 Metodología | 13 |
| 5.1 Plan de trabajo..... | 14 |
| 6 Estudios de factibilidad | 15 |
| 6.1.1 Factibilidad técnica | 15 |
| 6.1.2 Factibilidad económica | 16 |
| 6.1.3 Factibilidad operacional | 18 |
| 6.1.4 Factibilidad legal..... | 18 |
| 7 Herramientas y tecnologías a utilizar | 19 |

| | |
|--|----|
| 7.1 Herramientas | 19 |
| 7.1.1 Herramientas de documentación | 19 |
| 7.1.2 Herramientas desarrollo | 20 |
| 7.2 Tecnologías usadas..... | 21 |
| 7.2.1 Servicios Web RestFul..... | 21 |
| 7.2.1.1 XML..... | 21 |
| 7.2.2 Motor de bases de datos PostgreSQL..... | 22 |
| 7.2.3 Framework Symfony..... | 22 |
| 7.2.4 Android | 22 |
| 7.2.4.1 Arquitectura de Android | 22 |
| 7.2.4.2 Conceptos usados para el desarrollo de la Guía Móvil | 24 |
| 8 Desarrollo..... | 25 |
| 8.1 Guía Móvil – Módulo Ver Piso..... | 25 |
| 8.1.1 Requerimientos | 27 |
| 8.1.1.1 Requerimientos funcionales del Sistema | 27 |
| 8.1.1.2 Requerimientos no funcionales del Sistema | 27 |
| 8.1.2 Diagrama de casos de uso de alto nivel Guía Móvil..... | 28 |
| 8.1.3 Diagrama de casos de uso Ver Piso | 29 |
| 8.1.4 Diagrama de secuencia - Ver Piso | 30 |
| 8.1.5 Diagrama de secuencia - Ver Panorámica | 31 |
| 8.1.6 Diagrama de secuencia –Ver Exposición..... | 32 |
| 8.2 Guía Móvil – Módulo Realizar Recorrido | 33 |
| 8.2.1 Diagrama de casos de uso Realizar recorrido | 33 |
| 8.2.2 Diagrama de secuencia Cargar Recorrido..... | 35 |
| 8.3 Guía Móvil – Ingresar código exposición..... | 36 |
| 8.3.1 Diagrama de casos de uso Ingresar código de exposición | 36 |
| 8.3.2 Diagrama de secuencia – Ingresar código..... | 37 |
| 8.3.3 Diagrama de clases..... | 37 |
| 8.4 Subsistema mantenedor Web | 39 |
| 8.4.1 Requerimientos | 39 |
| 8.4.1.1 Requerimientos funcionales del sistema | 39 |
| 8.4.1.2 Requerimientos no funcionales del sistema | 40 |
| 8.4.2 Diagramas de Casos de Usos | 41 |
| 8.4.2.1 Diagrama de casos de uso de alto nivel | 41 |

| | |
|---|----|
| 8.4.2.2 Diagrama de casos de uso - Gestión Pisos | 42 |
| 8.4.2.3 Diagrama de casos de uso - Gestión Sala..... | 42 |
| 8.4.2.4 Diagrama de casos de uso - Gestión panorámica..... | 43 |
| 8.4.2.5 Diagrama de casos de uso - Gestión Exposición | 44 |
| 8.4.2.6 Diagrama de casos de uso – Gestión de recorridos..... | 45 |
| 8.4.3 Diagramas de secuencia | 46 |
| 8.4.3.1 Diagrama de secuencia Mantenedor Web – Agregar Piso..... | 46 |
| 8.4.3.2 Diagrama de secuencia – Mantenedor Web – Agregar Recorrido..... | 46 |
| 8.4.4 Diagrama de clases..... | 47 |
| 8.5 Arquitectura física del sistema | 49 |
| 8.6 Modelo relacional del sistema..... | 50 |
| 8.7 Interfaces gráficas del sistema | 51 |
| 8.7.1 Interfaz grafica Mantenedor Web | 51 |
| 8.7.2 Interfaz Grafica Guía Móvil..... | 52 |
| 9 Pruebas de software..... | 55 |
| Conclusiones | 57 |
| Referencias | 58 |

Resumen

El Museo Naval y Marítimo de Valparaíso tiene la necesidad de dar a conocer información acerca de las distintas exposiciones que allí se presentan. No obstante, esto se encontraba limitado por dos factores; primero, por la falta de personas guías para los visitantes particulares, y segundo, por la falta de espacio para colocar información adicional en las descripciones de cada exposición.

En la búsqueda de una solución a la problemática antes expuesta, se desarrolló un sistema móvil capaz de mostrar a los visitantes información adicional y actualizada de las distintas exposiciones del museo, llegando a suplir la falta de guías existente en dicha institución, ya que es capaz de conducir al visitante dentro de todo el museo. Por otro lado, se llevó a cabo un sistema capaz de gestionar toda la información que es mostrada en el sistema móvil.

En el presente documento se detalla la información necesaria que se utilizó para llevar a cabo este proyecto, como las etapas de análisis y diseño.

Finalmente, el Museo Naval implementó esta solución permitiendo no solamente entregar información adicional a los visitantes, sino que también, generar ganancias por conceptos de arriendo del sistema móvil. Además, será pionero en Chile en tener un sistema de esta magnitud.

Palabras claves: Servicios web, Android, PHP, Museo

Abstract

The “Museo Naval y Marítimo de Valparaíso” has a need to disclose information about the various exhibits that are there. However, this was limited by two factors: first, the lacks of people to individual visitors guides, and second, the lack of space for additional information in the descriptions of each exposure.

In the search for a solution to the problems set forth above, we developed a mobile system capable of showing to the visitors additional and updated information of the various museum’s exhibitions, supplying the low number of existing exhibitors in the institution, as it is able to lead the visitor into the museum. On the other hand, was carried out a system capable of managing all the information that is displayed in the mobile system.

This document details the information that was used to carry out this project, including analysis and design stages.

Finally, the Naval Museum implemented this solution allowing not only provide information to visitors, but also generate revenue by leasing concepts of mobile system. It will also be a pioneer in Chile to have a system of this magnitude.

Keywords: WebServices, Android, PHP, Museum

Lista de figuras

| | |
|--|----|
| Figura 3.1 – Smartphones | 5 |
| Figura 3.2 – PDA | 5 |
| Figura 3.3 – Symbian S60 y UIQ..... | 7 |
| Figura 3.4 - Windows Phone 6.5 y 7..... | 8 |
| Figura 3.5 – Android | 8 |
| Figura 3.6 – iOS | 9 |
| Figura 3.7 - Palm OS..... | 9 |
| Figura 4.1 - Primer piso Museo Naval Valparaíso..... | 11 |
| Figura 4.2 - Segundo piso Museo Naval Valparaíso..... | 12 |
| Figura 5.1 - Modelo desarrollo incremental..... | 13 |
| Figura 5.2 – Carta Gantt del proyecto..... | 14 |
| Figura 7.1 - Componentes del sistema operativo Android..... | 23 |
| Figura 8.1 - Diagrama de casos de uso de Alto nivel..... | 28 |
| Figura 8.2 - Diagrama de casos de uso Ver piso..... | 29 |
| Figura 8.3 – Diagrama de secuencia – Ver Piso | 30 |
| Figura 8.4 – Diagrama de secuencia – Mostrar Panorámica..... | 31 |
| Figura 8.5 – Diagrama de secuencia – Mostrar Exposición..... | 32 |
| Figura 8.6 - Diagrama de casos de uso Realizar recorridos | 33 |
| Figura 8.7 – Diagrama de secuencia – Cargar Recorrido | 35 |
| Figura 8.8 - Diagrama de casos de uso Ingresar código | 36 |
| Figura 8.9 – Diagrama de secuencia Ingresar código. | 37 |
| Figura 8.10 - Diagrama de clases Guía Móvil | 38 |
| Figura 8.11 - Diagrama de casos de uso de alto nivel..... | 41 |
| Figura 8.12 – Diagrama de casos de uso: Gestionar Piso | 42 |
| Figura 8.13 – Diagrama de casos de uso: Gestionar Sala | 42 |
| Figura 8.14 - Diagrama de casos de uso: Gestionar panorámica | 43 |
| Figura 8.15 - Diagrama de casos de uso: Gestionar exposición | 44 |
| Figura 8.16 - Diagrama de casos de uso: Gestionar Recorridos | 45 |
| Figura 8.17 – Diagrama de secuencia Agregar Piso | 46 |
| Figura 8.18 – Diagrama de secuencia Agregar Recorrido. | 47 |
| Figura 8.19 - Diagrama de clases..... | 48 |
| Figura 8.20 – Diagrama de arquitectura física del sistema | 49 |
| Figura 8.21 – Modelo relacional de la base de datos | 50 |
| Figura 8.22 – Login Mantenedor Web..... | 51 |
| Figura 8.23 – Página principal Mantenedor Web | 51 |
| Figura 8.24 – Menú principal..... | 52 |
| Figura 8.25 – Mostrar pisos | 52 |
| Figura 8.26 – Mostrar Sala..... | 53 |
| Figura 8.27 – Mostrar Exposición..... | 53 |
| Figura 8.28 – Primera sala de un Recorrido..... | 54 |
| Figura 8.29 – Ingreso de código de una exposición..... | 54 |

Lista de tablas

| | |
|---|----|
| Tabla 6.1 Costos Software..... | 16 |
| Tabla 6.2 Costos Hardware..... | 17 |
| Tabla 8.1 – Caso de uso narrativo – Ver piso..... | 29 |
| Tabla 8.2 – Caso de uso narrativo – Realizar recorrido..... | 33 |
| Tabla 8.3 – Caso de uso narrativo – Ingresar código..... | 36 |
| Tabla 9.1 – Prueba de software..... | 55 |

1 Introducción

La forma en que un museo muestra la información relacionada con sus exposiciones debe ser didáctica y explícita para que sea atractivo a sus visitantes. Por esto, se ha acordado en conjunto con el director del Museo Naval y Marítimo de Valparaíso, desarrollar un software móvil que ayudará a los visitantes del museo a ver y escuchar información de las exposiciones que están dentro de este. El sistema permitirá solucionar el problema de falta de personal guía para los visitantes particulares, ya que actualmente, solo están disponibles para visitas masivas a la institución.

El sistema está dividido en dos subsistemas; uno será la guía móvil para el museo naval y el otro será un mantenedor Web de pisos, salas, panorámicas, exposiciones y recorridos para ser mostrados en la guía móvil.

El software móvil consiste en mostrar un mapa (la silueta de la estructura del museo) que puede ser movido por el usuario, para conocer de manera a priori la estructura física del museo y el recorrido que debe seguir el visitante. Este mapa cuenta con puntos seleccionables que, al ser presionados, muestran por pantalla una imagen panorámica del lugar que se vincula a dicho punto, pudiendo apreciar las exposiciones que allí se encuentran y que pueden ser seleccionadas con el fin de acceder a información detallada acerca de ellas, permitiendo ser vista desde un dispositivo móvil. Además, el software cuenta con una caja de texto que puede ser accedida fácilmente desde cualquier sección de este, en la cual el usuario puede ingresar un código, propio de cada exposición, con el fin de acceder a información detallada acerca de esta.

Para llevar a cabo este proyecto, fue necesario trabajar con tecnologías como bases de datos y servicios web, los cuales están encargados de proporcionar los datos al cliente móvil, con el fin de que la información pueda ser modificada desde el mantenedor web, sin afectar el desempeño del cliente móvil. La implementación de la guía se realizó en un Tablet con el sistema operativo Android versión 1.6, el cual posee las ventajas de tener una pantalla de 7" y contar con tecnología *touchscreen*, lo cual permite gran interactividad entre el cliente móvil y el visitante. Para este desarrollo fue necesario el SDK de Android para crear el cliente móvil y de PHP orientado a objetos para crear el mantenedor Web.

El proyecto tuvo una duración de 10 meses, en este tiempo se llevaron a cabo las etapas de análisis y diseño, programación, pruebas y mantenimiento. Estas etapas serán documentadas de la forma clara y detallada en el presente informe.

2 Descripción del proyecto

El Proyecto consistió en crear un sistema de guiado móvil para los visitantes del Museo Naval y Marítimo de Valparaíso. El sistema está dirigido a solucionar la deficiente cantidad de guías para realizar recorridos a sus visitantes, ya que no siempre hay profesores que puedan cumplir con esta labor. Es por eso que se desarrolló una guía móvil que soluciona este problema con la implementación de un sistema software que fue instalado dentro de un dispositivo móvil con el fin de brindar información al visitante que lo posea al momento de realizar una visita al museo.

El desarrollo del proyecto fue llevado a cabo por un desarrollador de Software de la Pontificia Universidad Católica de Valparaíso.

Para lograr el proyecto, fue necesario el uso de tecnologías como son Servicios Web, Bases de datos y lenguajes como Java para Android y PHP orientado a objetos.

El sistema está constituido por dos subsistemas. El primero es una guía móvil y el segundo es un mantenedor Web de exposiciones, salas y recorridos.

El primer subsistema consiste en una guía móvil que mostrará al visitante información acerca de las salas y exposiciones que hay dentro del museo Naval. Para realizar esto, se crearon tres incrementos (módulos). El primer incremento fue “Ver Piso” diseñado de tal forma, que el visitante pueda acceder a una vista del primer o segundo piso del museo y tener la posibilidad de acceder a alguna de las salas para tener una vista más detallada de esta. Una vez seleccionada la sala, se pueden hacer cambios de ángulos de visión y seleccionar alguna exposición que se muestre en las imágenes. Finalmente, el visitante puede obtener información detallada respecto a la exposición seleccionada, pudiendo ser textual, visual y/o auditiva. De esta forma, el visitante obtiene una mayor cantidad de datos adicionales en relación a los que tendría si no tuviera la guía móvil en sus manos.

El segundo incremento consistió en desarrollar una opción que guía a los visitantes del museo a través de las distintas salas con la ayuda de reproducciones de sonidos. Esto significa que el visitante puede (con el uso de audífonos) seguir instrucciones que lo guiarán a través del recorrido cronológico que posee el museo.

El último incremento de la guía, consistió en desarrollar un módulo que permita al visitante ingresar un código para llegar directamente a la información de la exposición correspondiente al código recién ingresado. Esta herramienta fue desarrollada con el afán de darle un acceso directo al visitante a información adicional acerca de una exposición, independiente de la sala en que se encuentre.

Teniendo en cuenta el desarrollo de estos tres incrementos, el museo Naval pudo acceder a una importante herramienta de información.

El desarrollo del siguiente subsistema significó crear un mantenedor Web con el cual fue posible agregar, modificar o eliminar datos de las salas, exposiciones y recorridos del museo. Los datos modificados en este subsistema afectan directamente la información mostrada por la guía móvil, ya que estos dos sistemas operan con la misma base de datos.

La guía móvil fue diseñada para ser compatible con un Tablet Android v1.6 de 7”. El mantenedor Web puede ser visualizado dentro de cualquier computador perteneciente a la red que esté bajo el router destinado al proyecto.

Como se mencionó antes, el proyecto tuvo una duración de 10 meses, en la cual se llevaron a cabo las etapas de análisis y diseño, programación, pruebas y mantenimiento. Estas etapas serán documentadas de forma clara y detallada en el presente informe.

2.1 Objetivos

2.1.1 Objetivo general

Crear un sistema de guía móvil para los visitantes del Museo Naval y Marítimo de Valparaíso, utilizando servicios Web.

2.1.2 Objetivos específicos

- Analizar y modelar el recorrido cronológico que posee el museo a lo largo de sus salas y exposiciones.
- Realizar las etapas de análisis, diseño, implementación, pruebas y puesta en marcha del sistema.
- Poblar la base de datos del sistema con la información referente a las distintas exposiciones ubicadas en los salones del museo.
- Probar y validar el sistema propuesto en las dependencias del museo.

3 Marco Teórico

Se debe tener en cuenta que, para lograr el éxito del proyecto, hay que conocer tanto proyectos similares como información acerca de las tecnologías utilizadas dentro del mismo. A continuación se expone información pertinente que se consideró antes de proceder al desarrollo de este proyecto, como una implementación similar en el Museo Nacional de Francia (Museo el Louvre) y tecnologías que pueden llegar a ser oportunas.

3.1 Desarrollo en dispositivos móviles

La tecnología avanza muy rápido y la necesidad de mantenerse conectado en todo tiempo es primordial. Por esto, la tecnología móvil ha tenido que avanzar y crear dispositivos que cumplan con mantener el acceso a la información en cualquier momento y lugar. Los dispositivos móviles de la actualidad tratan de tener acceso a la información a través de una aplicación, para así acceder a los datos que el usuario requiera. Pero no todos los dispositivos cumplen con las necesidades de las aplicaciones (acceso a internet, localización GPS, aceleración 3D, cámara, etc.), por esto es necesario, al momento de la planificación de una aplicación móvil, limitar a qué tipo de dispositivos va a estar orientada la aplicación y qué recursos del dispositivo ocupará.

Los dispositivos cuentan con limitantes, que no son las mismas mencionadas anteriormente, sino que son el ahorro de energía que lleva a utilizar procesadores con una capacidad de cómputo limitada y baja frecuencia de reloj, el uso intensivo de diferentes recursos de los dispositivos (cálculos, gráficos, sonido, conexión a internet) limita la duración de la batería, la memoria disponible es mucho menor a la de una computadora.

3.1.1 Smartphones

Smartphone o teléfono inteligente es un tipo de dispositivo móvil, que cumple con funciones de un ordenador personal. Los Smartphone se caracterizan por contar con pantallas táctiles, sistemas operativos que permiten la conexión a internet y un completo soporte de correo electrónico, además de poseer las características generales de un teléfono móvil como recibir y enviar llamadas, mensajes de texto, reproductor de audio MP3 y video, cámara digital, conexión bluetooth, etc. Algunos ejemplos de teléfonos inteligentes son los Motorola de la serie MOTO Q, Nokia serie E, Nokia serie N, y los más reconocidos por estas características BlackBerry, iPhone o el Nexus One.



Figura 3.1 – Smartphones

3.1.2 PDA

La PDA fue creada originalmente como una agenda electrónica, con funciones como recordatorios, calendario, lista de contactos y bloc de notas. Actualmente las PDA's cumplen funciones como cualquier ordenador personal pero portátil.



Figura 3.2 – PDA

3.1.3 Plataformas de programación

Las plataformas de programación en dispositivos móviles no son los mismos que los lenguajes de programación para ordenadores personales por las limitantes que presentan los dispositivos móviles; existe una gran variedad de lenguajes para móviles como son los lenguajes derivados de lenguajes de programación de ordenador como J2ME (derivado de Java) y .Net compact framework (derivado de .net); también existen lenguajes para dispositivos móviles específicos como Android (java con librerías nativas, no J2ME), iPhone, BlackBerry.[1]

A continuación se explicarán los lenguajes más ocupados en aplicaciones para dispositivos móviles:

3.1.3.1 Java 2 micro edition

Java 2 micro edition está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDA's o electrodomésticos inteligentes. Esta edición tiene unos componentes básicos

que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que requiere sólo unos pocos Kilobytes de memoria para funcionar) en vez del uso de la JVM clásica, inclusión de un pequeño y rápido recolector de basura y otras diferencias. [2]

3.1.3.2 .Net compact framework

El Microsoft .NET Compact Framework es la plataforma de desarrollo .NET para dispositivos pequeños y es la causa principal del éxito de Microsoft en proveer a los usuarios de grandes experiencias en cualquier tiempo, lugar o dispositivo. El .NET Compact Framework trae el mundo del código gestionado (managed code) y Servicios Web XML a los dispositivos móviles, y habilita la ejecución de aplicaciones descargables en dispositivos tales como Asistentes Digitales Personales (PDA's), teléfonos móviles, Pocket PC's, etc.

Debido a que el .NET Compact Framework es un subconjunto del .NET Framework, los desarrolladores pueden fácilmente reutilizar código existente basado en varios entornos. Microsoft entregará extensiones para Visual Studio .NET, llamadas '*Smart Device Programmability*' para Visual Studio .NET. Esto significa que los cuatro millones de desarrolladores que tienen experiencia con la versión completa del .NET Framework pueden ahora desarrollar aplicaciones para pequeños dispositivos que ejecutan la versión compacta del Framework y debido a que cada desarrollador .NET será también un desarrollador de aplicaciones para pequeños dispositivos, crecerá exponencialmente el número de desarrolladores de .NET Compact Framework, más que para cualquier otra plataforma de desarrollo para pequeños dispositivos. [3]

3.1.3.3 Android

Android es un conjunto de software para dispositivos móviles el cual incluye un sistema operativo, *middleware* y aplicaciones llave. EL SDK de Android provee de herramientas y la API para poder comenzar el desarrollo de aplicaciones Android usando el lenguaje de programación Java. Se explicará con más detalle en el punto 4.2.3. [4]

3.2 Sistemas operativos móviles

3.2.1 Symbian OS

Es el sistema operativo más ocupado en los dispositivos móviles actualmente, donde se encuentran dispositivos de marcas como Nokia, Sony Ericsson, LG, Motorola, Samsung entre otros. Consta de dos grandes plataformas como lo son s60 y uiq, donde s60 es ocupado por Nokia y uiq mayormente utilizado por Sony Ericsson, Motorola, etc. Symbian tiene una gran compatibilidad con el desarrollo de aplicaciones en J2ME.



Figura 3.3 – Symbian S60 y UIQ

3.2.2 Windows Phone

Sistema operativo, llamado anteriormente Windows Mobile, creado por Microsoft utilizado en mayor parte en PDA y pocket pc. Se basa en el núcleo del sistema operativo Windows ME y un conjunto de aplicaciones básicas utilizando api de Microsoft Windows. La última versión de Windows Phone es la versión 7, en donde ocupa todos los servicios de Microsoft disponibles como lo son Windows Live, Xbox Live, Bing, Silverlight, Microsoft Office, Zune entre otros, esta versión está todavía en desarrollo.



Figura 3.4 - Windows Phone 6.5 y 7

3.2.3 Android

Android es una variante de Linux para dispositivos móviles, mayormente Smartphone. Fue lanzado por Google con su Smartphone Nexus One, también lo desarrollan HTC, T-Mobile, Qualcomm y Motorola. El desarrollo de aplicaciones para Android se realiza a través del SDK y librerías nativas de Java (sin J2ME).



Figura 3.5 – Android

3.2.4 iOS

Es un sistema operativo de Apple, desarrollado para su Smartphone el iPhone, posteriormente ocupado en iPod touch y iPad. Es un derivado de Mac OS X, consta de 4 capas de abstracción. [5]



Figura 3.6 – iOS

3.2.5 Palm OS

Sistema operativo desarrollado por la empresa Palm para sus PDA, algunas de las aplicaciones disponibles son libreta de direcciones, calculadora, calendario, registro de gastos, libreta de notas, tareas. Además, los equipos que llevan este sistema operativo traen puertos infrarrojos, bluetooth, entre otros elementos.



Figura 3.7 - Palm OS

3.3 MonGuive Louvre

Desde el año 2008 el Museo Nacional de Francia, Louvre, incorporó una guía móvil (MonGuive) enfocada a ser una herramienta lúdica y pedagógica, esencial para todos los visitantes que deseen saber un poco más. El sistema fue creado por la empresa Antenna Audio, que había desarrollado un proyecto similar en el Palacio de Versalles.

El *hardware* de esta guía es denominado como un computador de bolsillo, con una pantalla monocromática. Esto no es problema ya que el objetivo de la guía es mostrar información adicional y guiar a los visitantes dentro del museo de forma automatizada. Por el

lado de *software*, el sistema está enfocado en el modulo en el que se debe ingresar un código para saber información adicional respecto a una exposición dentro del museo.

Lamentablemente en la web no hay demasiada información respecto a este proyecto, salvo artículos periodísticos que mencionan las ventajas que suponen para el museo.

4 Museo Naval y Marítimo de Valparaíso

El Museo Naval y Marítimo de Valparaíso se ubica en el Paseo 21 de Mayo en Cerro Artillería de la región de Valparaíso.

Consta de dos pisos, en donde el recorrido por las salas esta hecho en forma cronológica, desde la época de Bernardo O'Higgins hasta la armada contemporánea. Dentro de ellas se ubican variados objetos, alusivos a los héroes y batallas navales desarrolladas en Chile. Arturo Prat, Bernardo O'Higgins, el combate Naval de Iquique, son algunas de las exposiciones permanentes en las cuales se pueden encontrar llamativos objetos, los cuales van desde el reloj detenido al momento del hundimiento de la Esmeralda hasta el uniforme original de Arturo Prat conservado en una vitrina de una sala dedicada especialmente a él.

Según su página Web la misión del museo es “preservar las colecciones en custodia y difundir el patrimonio marítimo de Chile, con el propósito de contribuir a incrementar la conciencia marítima nacional”. [6]

El museo cuenta con personal dedicado (Profesores de Historia) para guiar las visitas de instituciones que asistan al museo con un alto número de personas. Debido a esto los visitantes particulares del museo no pueden acceder a que uno de estos profesores guíe su recorrido.

En el primer piso del museo se encuentran las salas:

- Sala Bernardo O'Higgins
- Sala glorias navales
- Salas Lord Cochrane
- Sala Blanco Encalada
- Sala Diego Portales
- Cripta al Loor de Chile
- Sala guerra del pacifico
- Sala Arturo Prat

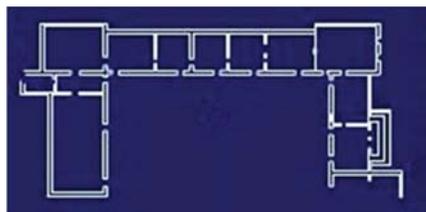


Figura 4.1 - Primer piso Museo Naval Valparaíso

En el segundo piso se encuentran las salas:

- Sala exposiciones temporales
- Sala Capitán Horniers
- Sala hermandad de la costa
- Sala Merino
- Sala infante marina
- Sala didáctica

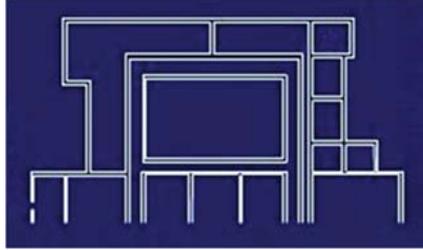


Figura 4.2 - Segundo piso Museo Naval Valparaíso

5 Metodología

Como todo proyecto de desarrollo de software, se tuvo que tomar una decisión muy importante, la que fue elegir un modelo de proceso de software. La elección de un modelo de proceso de software ayuda a establecer las actividades necesarias para llevar a cabo el proyecto, analizar los requerimientos, poder transformarlos en un producto de calidad y que al cliente le agrade. La incorrecta decisión de un modelo puede llevar a una mala planificación, implementación y posteriormente un producto de mala calidad o incluso al fracaso del proyecto.

Se ha llegado a la conclusión, en base a diferentes puntos de vistas de autores y desarrolladores, que no existe un modelo ideal para un proyecto. Por esto las empresas crean sus propios modelos de procesos a partir de diferentes experiencias tomando diferentes características de los modelos más ocupados. Debido a esto, se hace muy importante decidir por un modelo que cumpla con las características necesarias para llevar por buen camino el proyecto, por lo que el modelo guía es el desarrollo incremental. [7]

El modelo desarrollo incremental combina las ventajas del modelo cascada con las ventajas del modelo evolutivo, combina el modelo lineal secuencial con la filosofía interactiva de construcción de prototipos. El modelo plantea realizar incrementos del sistema, tomando cada incremento como un módulo y así en cada incremento ocupar el modelo cascada para implementar cada módulo.

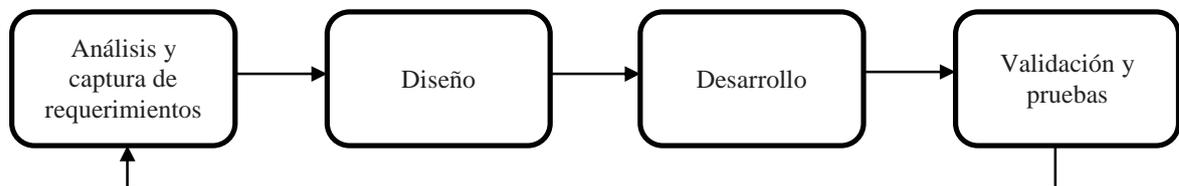


Figura 5.1 - Modelo desarrollo incremental

Los incrementos, que fueron obtenidos según el caso de uso general, pasaron por las etapas que se muestran en la figura 5.1:

- Análisis y captura de requerimientos: etapa donde se analizan y capturan los requerimientos del incremento, siendo funcionales como no funcionales.
- Diseño: a través del lenguaje de modelado UML, se diseñan los requerimientos del incremento para que los desarrolladores los entiendan de mejor manera.
- Desarrollo: según las herramientas de desarrollo se procederá a construir el incremento según los requerimientos y anteriores documentos de diseño.
- Validación y pruebas: el incremento se valida (cumple con los requerimientos) y posteriormente se realizan pruebas y se une al sistema completo, y se prueba el sistema completo. [8]

5.1 Plan de trabajo

Como se mencionó anteriormente, se optó por el modelo iterativo de desarrollo incremental. Como se muestra en la carta Gantt, la tarea de investigación de las tecnologías toma bastante tiempo ya que el desarrollador no tiene conocimiento del desarrollo de aplicaciones en dispositivos móviles. Asimismo, se muestran los incrementos en donde tienen lugar las etapas de análisis, diseño, implementación, pruebas e integración con el sistema.

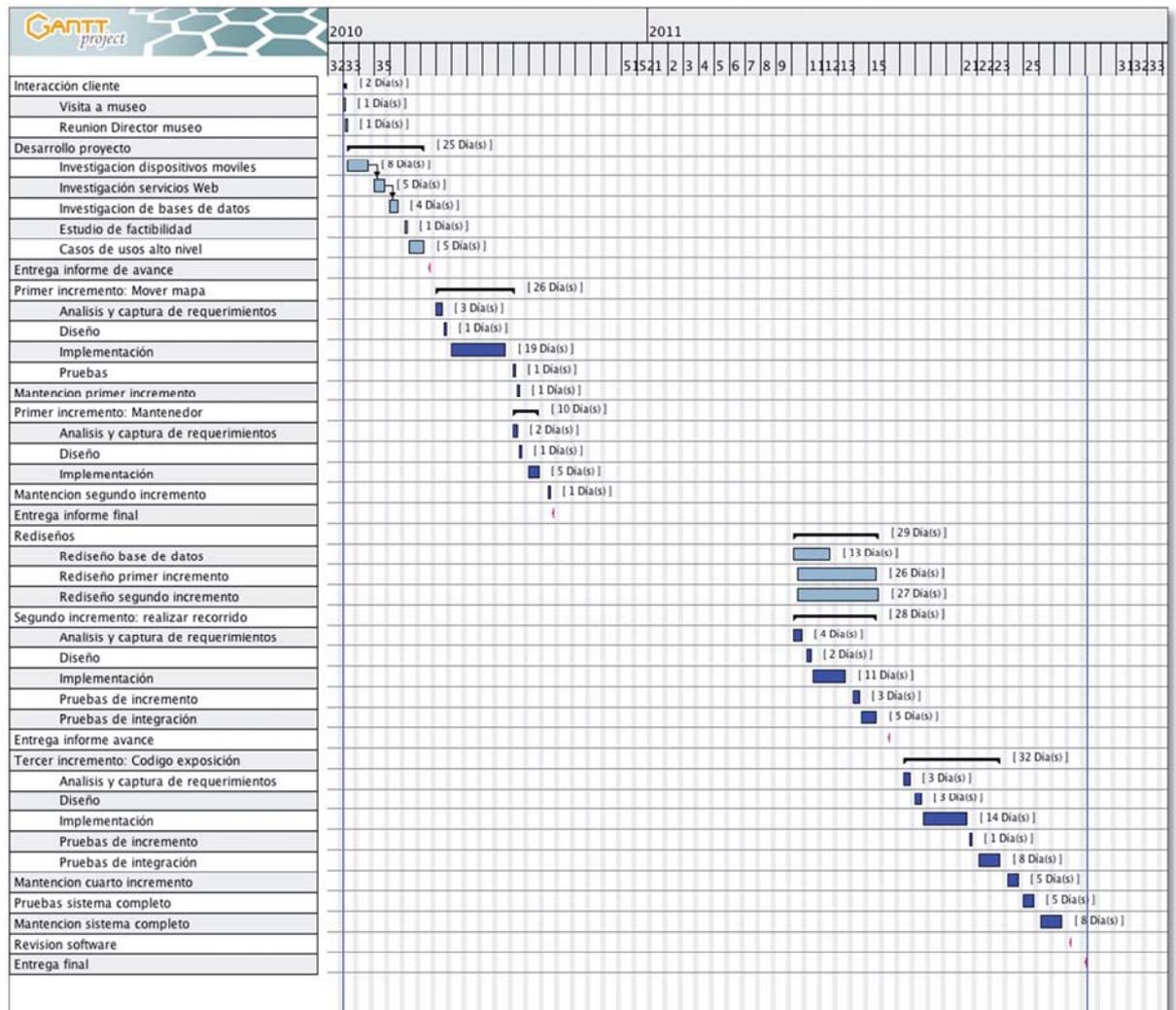


Figura 5.2 – Carta Gantt del proyecto

6 Estudios de factibilidad

Antes de comenzar a desarrollar este proyecto, fue necesario elaborar el siguiente estudio de factibilidad para determinar si era viable o no, lo que además permitió saber de manera oportuna, si existían los recursos necesarios para llevarlo a cabo. Este estudio incluye los objetivos, alcances y restricciones del sistema desarrollar.

A continuación se detallan cuatro estudios de factibilidad, útiles para el desarrollador del proyecto al momento de determinar si este era factible o no.

6.1.1 Factibilidad técnica

Este estudio indica si se cuenta con los recursos de hardware y software necesarios para llevar a cabo el proyecto.

Hardware con el que se cuenta para el proyecto:

- Un notebook del desarrollador HP Pavilion DV2125LA
- Computadores de la sala de proyecto de la escuela de Informática, para realizar pruebas.
- Servidor final que alojará servicios Web y base de datos para ser consumidos por el cliente móvil.
- Router Linksys WRT54GL.
- Tablet Android 1.6 de 7”.

Software con el que se cuenta para realizar el proyecto:

- Eclipse 3.6.1 con ADT Android.
- Netbeans 6.9
- Apache 2.2.11
- PHP5
- Glassfish para publicar los servicios Web creados.
- PostgreSQL 8.4.4-1.
- Microsoft Office 2003, proporcionado por la PUCV para el desarrollo de la documentación.
- Microsoft Windows XP y Microsoft Windows 7.
- Ubuntu server 10.04.1.
- StarUML.

6.1.2 Factibilidad económica

Este estudio indica a través del VAN (valor actual neto) si el proyecto es factible o no, basándose en los costos y beneficios asociados al proyecto durante un periodo de cinco años.

Las etapas asociadas al desarrollo del proyecto son etapa de análisis, diseño, implementación, pruebas y puesta en marcha, que serán realizadas por 2 desarrolladores de software.

Suponiendo que la hora hombre esta valorizada en 0,4UF, trabajando 3 horas por día, durante 22 días al mes y en virtud del uso de un modelo de proceso “desarrollo incremental”, las etapas del desarrollo del software estuvieron presentes hasta terminar con la implementación. Se planea realizar la puesta en marcha del proyecto en Junio del 2011. Pero el desarrollo del proyecto se realizó en 7 meses, por lo que la inversión respecto al recurso humano, esta valorizada en CL\$7.917.940. [9]

Por lo general el recurso software utilizado para este proyecto es de licencia propia o licencia libre. Eclipse, Netbeans, Glassfish, PostgreSQL, Ubuntu Server 10.04.1 y StarUML son software gratuitos descargables de sus respectivas Web, por lo que no significan un gasto para el proyecto. Microsoft Windows XP propiedad del desarrollador, con licencia adquirida al momento de realizar la compra del computador. Microsoft Office 2003, es un software proporcionado por la Universidad, por lo tanto tampoco significa un gasto para el proyecto.

Tabla 6.1 Costos Software.

| Nombre | Descripción | Monto real | Monto ficticio | Precio de adquisición |
|-----------------------|---|------------|----------------|-----------------------|
| Eclipse | http://www.eclipse.org/ | \$0 | \$0 | - |
| Glassfish | https://glassfish.dev.java.net/ | \$0 | \$0 | - |
| PostgreSQL | http://www.postgresql.org/ | \$0 | \$0 | - |
| Ubuntu Server 10.04.1 | http://www.ubuntu.com | \$0 | \$0 | - |
| StarUML | http://staruml.sourceforge.net/en/ | \$0 | \$0 | - |
| Microsoft Windows XP | Licencia incluida con la compra del notebook HP DV2125la | \$0 | \$0 | CL\$118.000 |
| Microsoft Office 2003 | Proporcionado por la Universidad gratuitamente. | \$0 | \$0 | CL\$213.713 |

A nivel de hardware, se debe calcular la depreciación mensual de los activos fijos pertenecientes al desarrollador y el valor total respecto a los activos que se utilizarán mientras el proyecto esté en marcha.

Será necesario utilizar un computador para realizar las etapas de desarrollo del proyecto y la documentación asociada, éste pertenece al desarrollador. Para las pruebas se utilizará el emulador proporcionado por el SDK de Android, posteriormente estas se realizaran en un dispositivo real adquirido por el Museo Naval. Para la puesta en marcha del proyecto será necesario adquirir al menos un Access *point*, el cual proveerá de señal *wifi* al dispositivo móvil. Además, para que éste pueda funcionar, será necesario instalar cable de red RJ45 que será el medio de comunicación entre el servidor (el museo ya dispone de un equipo para este fin) y el/los Access *Points*.

Tabla 6.2 Costos Hardware.

| Nombre | Descripción | Monto real | Monto ficticio |
|---|---|-----------------------|---|
| Notebook hp Dv2125 la (desarrolladores) | Comprado por CL\$700.000 en una multitienda. | \$0 | CL\$700.000/60 meses (vida útil) = CL\$11.670 |
| Linksys Access Point WAP54G (6 equipos) | CL\$56.790 en PCFactory. | CL\$340.740 | CL\$340.740 |
| 10 mts cable RJ45 | CL\$250/mt | CL\$2.500 | CL\$2.500 |
| 10 Cabezas de cable RJ45 | CL\$1.000 | CL\$1.000 | CL\$1.000 |
| Tablet Android 1.6 de 7" (6 equipos) | Dispositivos deben ser adquiridos en la página www.dealextreme.com con un valor de US\$107 | CL\$308.170 (US\$642) | CL\$308.170 |
| Total | | | CL\$664.080 |

Se estima también, que el sistema en funcionamiento generará ganancias al museo, ya que, podrán percibir las en el arriendo de un dispositivo móvil para los visitantes.

El museo en promedio tiene 20 visitas al día, de los cuales se estima que un 30% arrendará el dispositivo móvil para realizar su recorrido, el arriendo está valorizado en CL\$3.000. Por lo tanto anualmente por arriendo de dispositivos se estima una ganancia de CL\$4.320.000 anual.

El proyecto tendrá un costo de mantención semestral valorizado en 6UF.

Ahora, para saber si el proyecto es factible o no, se procederá a realizar el cálculo del valor actual neto (VAN).

Inversión inicial: CL\$8.582.020

Tasa de descuento: 8% (Según IPC estimado)

Flujo anual: CL\$4.068.000

Horizonte de la evaluación: 5 años

$$\text{VAN} = -\text{CL\$}8.582.020 + (\text{CL\$}4.068.000/1.08^1) + (\text{CL\$}4.068.000/1.08^2) + (\text{CL\$}4.068.000/1.08^3) + (\text{CL\$}4.068.000/1.08^4) + (\text{CL\$}4.068.000/1.08^5)$$

$$= \text{\$CL}7.660.324$$

Ya que el VAN es mayor a 0, el proyecto es factible desde el punto de vista económico.

6.1.3 Factibilidad operacional

Probar la factibilidad operacional tuvo como objetivo determinar si el sistema, una vez instalado y puesto en marcha, opera efectivamente dada la organización propia del museo.

El proyecto contó con el apoyo de un profesor guía dedicado exclusivamente a la interlocución entre el desarrollador y el Director del Museo. Además, está encargado de completar la información respecto a las distintas salas. Por lo tanto, cuenta con una persona que está a cargo del mantenimiento de la información incluida en él.

El sistema no causa ningún perjuicio dentro del funcionamiento actual del museo. Por el contrario, permite que los visitantes cuenten con información adicional, si se mira desde el punto de vista del usuario final. Además, permite contar con una base de datos centralizada de la información respectiva de los objetos dentro del museo.

La productividad de los empleados del museo no se ve afectada por este sistema, ya que no busca remplazar la labor de los profesores guía para instituciones, sino que busca aplacar la falta de estos mismo para los visitantes particulares.

El sistema es factible desde este punto de vista, ya que cuenta con apoyo de profesores, y no significa un perjuicio para el funcionamiento actual del museo.

6.1.4 Factibilidad legal

Probar la factibilidad legal tiene como objetivo demostrar que el sistema no transgrede algún artículo de la ley N° 19.223 vigente en Chile, asociada a los delitos informáticos. A continuación se muestran los cuatro artículos de esta ley. [10]

"Artículo 1°.- El que maliciosamente destruya o inutilice un sistema de tratamiento de información o sus partes o componentes, o impida, obstaculice o modifique su funcionamiento, sufrirá la pena de presidio menor en su grado medio a máximo.

Si como consecuencia de estas conductas se afectaren los datos contenidos en el sistema, se aplicará la pena señalada en el inciso anterior, en su grado máximo.

Artículo 2°.- El que con el ánimo de apoderarse, usar o conocer indebidamente de la información contenida en un sistema de tratamiento de la misma, lo intercepte, interfiera o acceda a él, será castigado con presidio menor en su grado mínimo a medio.

Artículo 3°.- El que maliciosamente altere, dañe o destruya los datos contenidos en un sistema de tratamiento de información, será castigado con presidio menor en su grado medio.

Artículo 4°.- El que maliciosamente revele o difunda los datos contenidos en un sistema de información, sufrirá la pena de presidio menor en su grado medio. Si quien incurre en estas conductas es el responsable del sistema de información, la pena se aumentará en un grado."

7 Herramientas y tecnologías a utilizar

7.1 Herramientas

Tanto la implementación como la documentación de este proyecto requirieron de software para llevarlas a cabo. A continuación se da la información de todas aquellas utilizadas en el desarrollo del proyecto.

7.1.1 Herramientas de documentación

La etapa de documentación fue un pilar fundamental en el desarrollo del proyecto, ya que permitió a todas las partes participantes en el proyecto coordinarse para lograr el objetivo deseado. Además, sirvió como guía instructiva para futuras modificaciones del mismo. Para realizar esto se utilizó:

- Microsoft office 2003: suite de aplicaciones de oficina, la cual contiene aplicaciones para procesar texto, crear diapositivas, creación hojas de cálculo. La herramienta esta proporcionada bajo licenciamiento de la PUCV en los laboratorios de la Escuela de Ingeniería de Informática.
- GanttProject 2.0.10: principal funcionalidad la creación de la carta Gantt.
- StarUml: herramienta gratuita que soporta la mayoría de los diagramas del lenguaje de modelado UML, que fueron ocupados en la parte de análisis y diseño de cada módulo del sistema.
- MySQL WorkBench 5.2: Herramienta para el manejo grafico de bases de datos. Gratuito.
- Foxit Reader: lector de archivos pdf gratuito.

7.1.2 Herramientas desarrollo

Para la creación de aplicaciones móviles existen una gran variedad de herramientas. Por esto, fue necesario determinar qué características son las que cumplen con los requerimientos del proyecto. Es por ello que se seleccionaron las siguientes herramientas para lograr los propósitos de este. Estas herramientas son:

- Eclipse Helios 3.6.1: IDE de código abierto multiplataforma para la programación en el lenguaje JAVA. Es posible adquirirlo desde la web <http://www.eclipse.org>.
- Netbeans 6.9: IDE de código abierto multiplataforma para la programación en el lenguaje JAVA. Es posible adquirirlo desde la web <http://www.netbeans.org>.
- Android ADT: *Plugin* disponible en la web de desarrolladores Android para ser integrado con el IDE Eclipse con el fin de desarrollar proyectos Android de manera más ordenada y eficaz. Es posible adquirirlo desde la web <http://developer.android.com/sdk/eclipse-adt.html>.
- Glassfish: es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL. Es el servidor que alberga los servicios Web creados.
- Apache 2.2.11: Servidor HTTP de código abierto multiplataforma que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Este es el servidor que permite el acceso Web al sistema mantenedor.
- PostgreSQL: Es un sistema de gestión de base de datos relacional con la capacidad de operar con el paradigma de la orientación a objetos y libre, publicado bajo la licencia BSD. Es el sistema de gestión de bases de datos libre más avanzado hasta el momento.
- Ubuntu server 10.04.1: Sistema operativo diseñado para operar como servidor de servicios para terceros, creado por la comunidad Ubuntu. Fue instalado en una maquina servidor y contuvo los servicios mencionados en los tres puntos anteriores.
- Android SDK: Conjunto de librerías, ejemplos y emuladores proporcionados por la web de desarrolladores Android. Disponible en <http://developer.android.com/sdk/index.html>.

7.2 Tecnologías usadas

Para llevar a cabo el proyecto “Guía móvil para el Museo Naval de Valparaíso” fue necesario el uso de una arquitectura basada en servicios Web. Esta tecnología es útil para ser consumida por dispositivos móviles y, en general, desde cualquier dispositivo programable que cuente con una conexión a una red que sirva de enlace entre Servicio Web y cliente.

7.2.1 Servicios Web RestFul

REST define un set de principios arquitectónicos por los cuales se diseñan servicios web haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia clientes escritos en diversos lenguajes. REST emergió en los últimos años como el modelo predominante para el diseño de servicios. De hecho, logró un impacto tan grande en la web que prácticamente logró desplazar a SOAP y las interfaces basadas en WSDL por tener un estilo bastante más simple de usar.

Una implementación concreta de un servicio web REST sigue cuatro principios de diseño fundamentales:

- Utiliza los métodos HTTP de manera explícita.
- No mantiene estado.
- Expone las URIs en forma de directorio
- Transfiere XML, JSON o ambos.

Para el desarrollo del servicio web necesario para el proyecto se crearon “clases entidades” a partir de la base de datos, para posteriormente poder crear el servicio web RestFul que sirve de interfaz entre la guía móvil y la base de datos. Todo esto se creó con el ayudante de creación de servicios Web Restful que proporciona Netbeans.

7.2.1.1 XML

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

La guía móvil recibirá el XML de respuesta y será almacenado en un *Document*, un tipo de dato especial para almacenar este tipo de documentos. Para posteriormente ser recorrido por nodos. [11]

7.2.2 Motor de bases de datos PostgreSQL

PostgreSQL es un poderoso sistema de bases de datos objeto-relacionales de código abierto. Tiene la característica de poder ejecutar procedimientos almacenados escritos y diversos lenguajes, como también uno propio PL/pgSQL que es muy similar a PL/SQL. Además de la capacidad de ejecutar triggers y almacenar vistas.

Se optó por este motor de base de datos por ser de código abierto y muy robusto para albergar la base de datos relacional necesaria para este proyecto.

7.2.3 Framework Symfony

Symfony es un *framework* PHP que facilita el desarrollo de aplicaciones web. Ejecuta automáticamente tareas comunes de este tipo de proyectos, logrando una ganancia de tiempo. Así, el desarrollador puede dedicarse exclusivamente a desarrollar características únicas de cada proyecto.

Para lograr estas tareas comunes, Symfony utiliza una capa de abstracción de la base de datos, tal que, pueda ser manipulada mediante métodos de objetos persistentes. La capa de abstracción de la base de datos utilizada para este proyecto se llama Doctrine. El patrón utilizado por este *framework* es el “Modelo-Vista-Controlador”. Este patrón utiliza el controlador como un semáforo entre las vistas y los modelos. Las primeras, son las interfaces gráficas con las que el usuario puede interactuar. La segunda contiene todas las funciones encargadas de extraer e insertar datos desde la base de datos. Por lo tanto, las peticiones se realizan al controlador, este solicita datos al Modelo, los procesa y posteriormente envía datos preparados para ser mostrados en pantalla por la Vista.

7.2.4 Android

Es el sistema operativo propuesto por Google para dispositivos móviles que ha tenido gran aceptación tanto de los usuarios de los dispositivos móviles como de los desarrolladores.

Está basado en un Kernel Linux modificado y hasta el momento está disponible en numerosos dispositivos móviles a nivel mundial, dándole un toque atractivo a estos dispositivos ya que las aplicaciones para Android se pueden conseguir de forma gratuita, la mayoría.

7.2.4.1 Arquitectura de Android

El siguiente diagrama muestra la arquitectura de Android que será explicada con mayor detalle posteriormente.

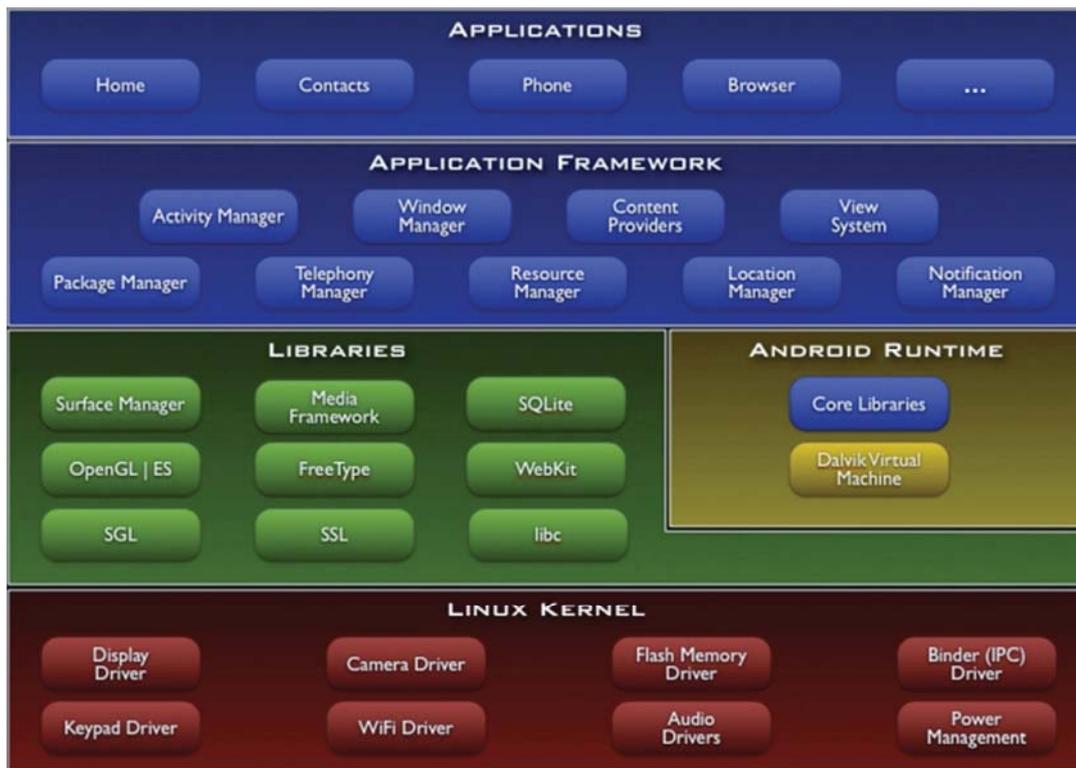


Figura 7.1 - Componentes del sistema operativo Android

- **Aplicaciones:** El sistema operativo Android será vendido con una serie de aplicaciones predefinidas para navegar por internet, visualizar correo administrador de llamadas. Todas las aplicaciones están escritas en JAVA.
- **Framework:** Al proporcionar una plataforma de desarrollo abierto da la posibilidad al desarrollador crear aplicaciones innovadoras al extremo con la misma API que fueron desarrolladas las aplicaciones incluidas por defecto. Detrás de las aplicaciones hay un conjunto de servicios y sistemas tales como :
 - **Vistas:** Estas son un conjunto de listas, cajas de texto, botones, redes e incluso un navegador web embebido. Provistas para construir aplicaciones atractivas para el usuario.
 - **Content Providers:** Permite a las aplicaciones acceder a datos de otras aplicaciones.
 - **Resource Manager:** Facilita el acceso a recurso no-código, como *strings* localizados, gráficos y archivos de *layouts*.
 - **Notification Manager:** Permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.

- Activity Manager: Gestiona el ciclo de vida de las aplicaciones. Una *Activity* es una vista en la aplicación que cuenta con las vista más el código de programación.
- Runtime Android: Android incluye un conjunto de bibliotecas que proporciona la mayor parte de la funcionalidad en las bibliotecas del núcleo del lenguaje de programación JAVA. Cada aplicación Android se ejecuta en su propio proceso, con su propia instancia de la máquina virtual Dalvik. Éste ha sido escrito para que un dispositivo pueda ejecutar múltiples máquinas virtuales de manera eficiente.
- Kernel Linux: Android se basa en la versión 2.6 del kernel de Linux proporcionando servicios básicos como la seguridad, la gestión de memoria, gestión de procesos, pila de red y el modelo controlador. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

7.2.4.2 Conceptos usados para el desarrollo de la Guía Móvil

La guía móvil está pensada para utilizar al máximo las prestaciones que el Hardware del *tablet* y la API de Android. Para esto es necesario tener en cuenta una serie de conceptos para poder continuar la lectura de este documento de manera fluida.

Con el fin de poder interactuar con el servicio web sin que la aplicación deje de funcionar por la demora generada en este proceso, Android proporciona una clase llamada *AsyncTask*, para facilitar el manejo de hilos. Esta clase debe ser heredada en una subclase situada dentro de la clase de la interfaz gráfica, para poder funcionar de forma paralela al hilo principal (hilo de interfaz gráfica). Ésta debe tener al menos un método *doInBackground*, encargado de ejecutar tareas largas sin que el hilo principal envíe un error al usuario diciendo que la aplicación está demorando demasiado. El método *onPostExecute* se encarga de ejecutar un código inmediatamente después de que *doInBackground* ha terminado.

Al crear un objeto del tipo de esta subclase se debe invocar al método *execute* para echar a correr el hilo.

Mientras el hilo interactúa con el servicio Web la aplicación podrá mostrar un cuadro emergente de progreso mientras esta tarea termina. Este cuadro es un objeto de tipo *ProgressDialog* proporcionado por la API de Android.

Los *Layouts* son cajas contenedoras de vistas, tales como botones, imágenes, formularios, etc. Estas son mostradas por pantalla conteniendo estas vistas.

Para el desarrollo de algunas interfaces gráficas (guía móvil) de este proyecto será necesario utilizar *RelativeLayout* el cual ordena los elementos contenidos dentro de él en relación a otros elementos. Esto será muy útil para dibujar los botones encima del piso y de las panorámicas ya que sus posiciones son dinámicas y pueden ir variando en el tiempo.

Entendido esto se procederá a la documentación asociada al primer incremento de la guía móvil. [12]

8 Desarrollo

8.1 Guía Móvil – Módulo Ver Piso

Con el fin de que el museo pueda mostrar a los visitantes toda la información ingresada y organizada en el sistema mantenedor Web, se creó una solución tal que el visitante pueda acceder a ella en cualquier momento, mientras se encuentre dentro del recinto.

Para el desarrollo de la solución se crearon los módulos “Ver piso”, “Realizar Recorrido” e “Ingresar Código” para obtener la aplicación Android que fue instalada en el Tablet y así contar con el sistema “guía móvil”. Para desarrollarlo se utilizó el enfoque orientados a objetos ya que la API de Android está escrita con el mismo.

El primer módulo (“Ver Piso”) consiste en el ingreso, por parte del visitante, a la opción “Ver Piso” la cual muestra una imagen del piso número uno del museo con botones encima de esta. Estos, están ubicados según las coordenadas apuntadas en el mantenedor Web. Una vez seleccionado uno de estos botones, el sistema muestra una imagen panorámica de la sala, con la opción de seleccionar alguna de las exposiciones para mostrar información relacionada a esta.

El segundo módulo (“Realizar recorrido”) consiste en el ingreso por parte del visitante a la opción “Realizar recorrido” dentro del menú principal. Posteriormente, se debe seleccionar el recorrido que desea realizar, para que la guía móvil le muestre la primera sala con la que comienza el recorrido, además de unos controles en la parte inferior de la pantalla que permiten avanzar y retroceder entre las salas, según el orden del recorrido. Al terminar la carga de la sala, comienza una reproducción de un sonido descriptivo de la sala, este sonido puede ser pausado y termina cuando se pasa a la siguiente o anterior sala. Como en el módulo anterior, también se puede acceder a las exposiciones de cada una de las salas.

El tercer y último módulo, corresponde a “Ingresar código”. Se muestra por pantalla un cuadro de texto en donde el visitante puede ingresar un código que obtiene de alguna exposición. Una vez ingresado, el sistema responde mostrando información alusiva a dicha exposición, si no, muestra un mensaje indicando que la exposición no existe o no está registrada en el sistema.

La información mostrada por la guía móvil es recibida a través de las clases parseadoras, las cuales se comunican con el servicio web RestFul especialmente diseñado para este proyecto.

Durante el desarrollo de este sistema se decidió hacer unos cambios en las tecnologías a utilizar en este proyecto, con el fin de que éste fuera factible económica y funcionalmente para el museo. A continuación se muestra la lista de cambios realizados y la explicación respecto a la revisión uno de este informe.

- Cambio de dispositivo Nokia a Tablet Android: Este cambio fue realizado ya que fue encontrando un tablet de dimensiones más grandes que un dispositivo Nokia común y además más económico. En consecuencia también se debió cambiar el sistema operativo sobre el cual instalar el sistema y el SDK para el desarrollo. Por otro lado la programación de la interfaz gráfica en Android da la posibilidad de trabajar con objetos situados por coordenadas sin pasar por el Canvas de Android.

- Cambio de SDK Nokia a SDK Android: Este cambio se produce a raíz del cambio anterior.
- Cambio del campo oid y atributos asociados de las tablas por campos varchar en la base de datos: Este cambio fue realizado porque en un principio se planeaba trabajar con *streaming* o paso de *arrays* binarios para los archivos alojados en la base de datos en forma de oid. Esto se puede hacer a través de servicio Web con mensajes SOAP en la implementación JAX-WS de Java, pero Android no puede consumir este tipo de servicios web. Por lo tanto fue necesario hacer el cambio de campo a varchar el cual almacenará la url de los archivos alojados en el servidor.
- Cambio de servicio web JAX-WS por servicio web RestFul: Este cambio se realizó ya que Android no cuenta con librerías nativas para la lectura servicios web SOAP, pero si cuenta con librerías nativas para acceder a servicios web RestFul.

A continuación se procede al análisis y diseño de este sistema.

8.1.1 Requerimientos

A continuación se detallan los requerimientos del sistema. Luego se expone el análisis y diseño con UML.

8.1.1.1 Requerimientos funcionales del Sistema

- El sistema deberá comunicarse con el servicio web RestFul para recuperar los datos almacenados en la base de datos. (oculto)
- Mostrará un mapa de cada piso del Museo (son dos), siempre mostrando el primer piso primero.(visible)
- Encima del mapa del piso deben haber botones que al ser presionados conduzcan a una vista panorámica de la sala seleccionada.(visible)
- La posición de los botones es traída en coordenadas desde a base de datos. (oculto)
- Mostrará una vista panorámica de una sala (seleccionada en el mapa del piso).(visible)
- Encima de la vista panorámica deben haber botones invisibles encima de las exposiciones con información disponible. Estos botones al ser presionados conducirán a la información de la exposición seleccionada.(visible)
- La información de las exposiciones será mostrada en una plantilla igual para todas.(visible)
- Si una exposición tiene más de una imagen asociada, se deberá mostrar una galería de fotos que permita cambiar la imagen visible. (visible)
- Si una exposición tiene audio asociado, este será reproducible desde la misma plantilla que contendrá a la información.(visible)
- Al presionar el botón de *play/pause* dentro de un recorrido este deberá detenerse o reproducirse, según corresponda.(visible)
- Al presionar el botón avanzar o retroceder en un recorrido, se deberá avanzar o retroceder en el recorrido, según corresponda. (visible)
- Al ingresar el código incorrecto de alguna exposición, se debe mostrar un mensaje que mencione el error. (visible)

8.1.1.2 Requerimientos no funcionales del Sistema

- La posición de los botones encima de las imágenes deberá ser precisa, es decir, debe estar ubicado encima de una sala y no en otro lugar.
- Las imágenes de los pisos y las panorámicas del museo deberán ser *scrolleables*, con el fin de que si las dimensiones de la imagen son mayores a las de la pantalla estas puedan ser desplazadas por el usuario.
- La información de las exposiciones deberá ser mostrada encima de un fondo de pantalla acorde al museo.
- La velocidad de carga de las imágenes o de la información no deberá tardar en ser mostrada en el dispositivo.

- Deberá tener un botón que le permita al usuario volver en cualquier momento a hacia la pantalla anterior.

8.1.2 Diagrama de casos de uso de alto nivel Guía Móvil

El sistema tiene tres opciones con las que el usuario puede interactuar en primera instancia: Ver piso, realizar recorrido e ingresar un código. Las dos primeras opciones representan una forma indirecta de llegar a la información relacionada con una exposición determinada por el visitante. En cambio con la última opción se puede llegar de forma directa.

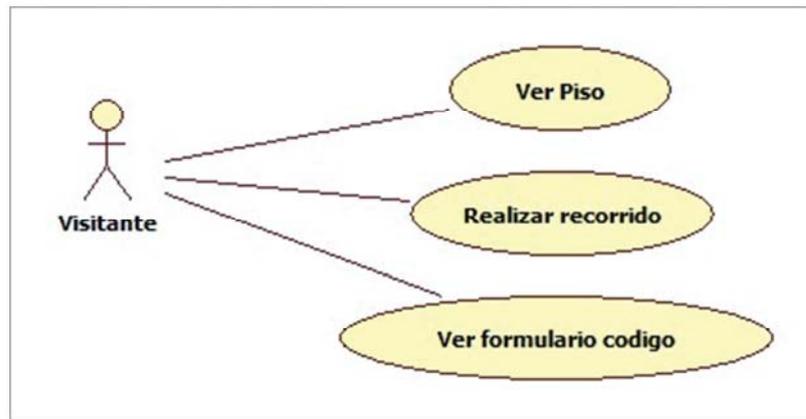


Figura 8.1 - Diagrama de casos de uso de Alto nivel

8.1.3 Diagrama de casos de uso Ver Piso

Este diagrama de casos de uso representa la interacción del visitante estando dentro de la opción “Ver Piso”. Con esta opción, el visitante puede ver el mapa del museo, las salas y las exposiciones de este.

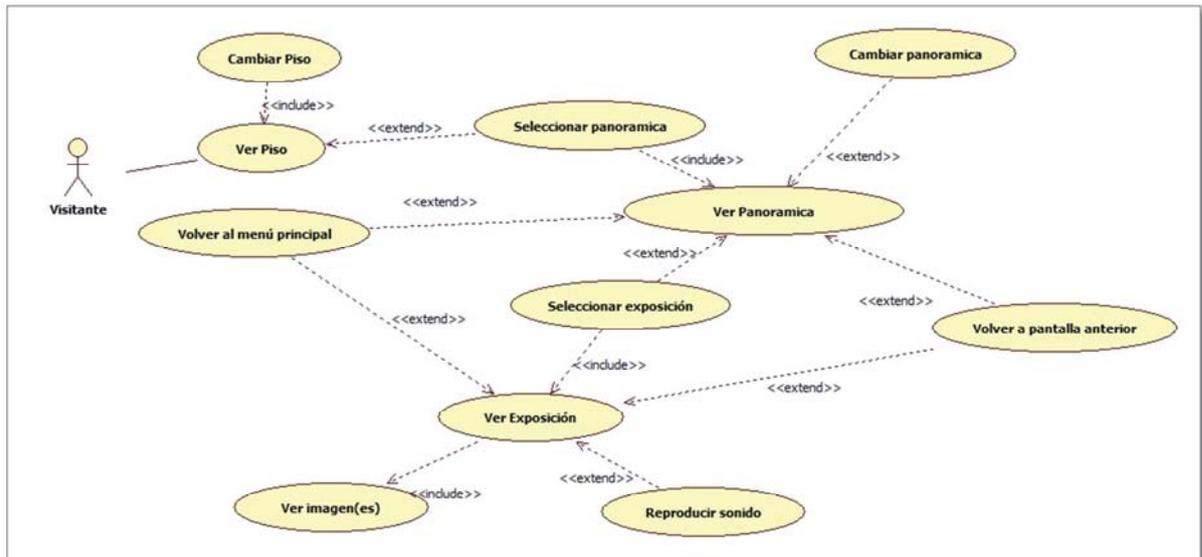


Figura 8.2 - Diagrama de casos de uso Ver piso

A continuación la forma narrativa de este diagrama.

Tabla 8.1 – Caso de uso narrativo – Ver piso

| | | |
|-------------------------------|--|---|
| Nombre del caso de uso | Ver Piso | |
| Actores | Visitante | |
| Propósito | Mostrar al visitante una vista de uno de los dos pisos del museo, con la opción de cambiarlos (los pisos), pasar por una vista panorámica de una de las salas y finalmente seleccionar una exposición para ver la información relacionada a ella. | |
| Descripción | El visitante, al tener el dispositivo en su mano, selecciona la opción “Ver Piso” la cual le muestra la imagen del primer piso del mapa, con la opción de cambiar al piso dos. Posteriormente puede seleccionar un botón que aparece encima de cada una de las salas del piso el cual lleva a mostrar una vista panorámica de lo recién seleccionado. En esta parte, la panorámica muestra las exposiciones disponibles dentro de la sala, las cuales son seleccionables para mostrar la información relacionada a ésta. El usuario además tiene la opción de cambiar la panorámica que esta viendo con el fin de tener otro ángulo de visión de la sala. La información de la exposición puede ser en forma de imágenes, texto y/o sonidos. | |
| Flujo principal | Paso | Acción |
| | 1 | El visitante selecciona la opción Ver Piso del menú principal. |
| | 2 | El sistema muestra el primer piso. |
| | 3 | El visitante puede seleccionar uno de los botones para que se muestre una imagen panorámica de la sala. |
| | <extends> | |
| | 4 | El sistema muestra una imagen panorámica. |
| | 5 | El visitante puede seleccionar uno de los botones situados encima de |

| | | |
|-------------|-----------------|---|
| | <<extends>> | las exposiciones expuestas en la imagen panorámica. |
| | 6 | El sistema muestra la información de la exposición seleccionada. |
| Extensiones | 2.1 - 4.1 - 6.1 | El visitante puede volver al menú principal. |
| | 4.1 - 6.1 | El visitante puede volver a la pantalla anterior |
| | 2.2 | El visitante puede cambiar el piso. El museo solo tiene dos pisos. |
| | 4.2 | El visitante puede cambiar la imagen panorámica mostrada por otra de las que estén disponibles. |
| | 6.2 | El visitante puede ver una galería de imágenes, siempre y cuando la exposición cuente con una. |
| | 6.3 | El visitante puede reproducir un sonido correspondiente a la exposición. |

8.1.4 Diagrama de secuencia - Ver Piso

Este diagrama muestra la secuencia de acciones necesarias para poder capturar los datos desde el servicio Web para finalmente dibujar en la pantalla del Tablet el mapa con los botones correspondientes a las distintas salas del museo.

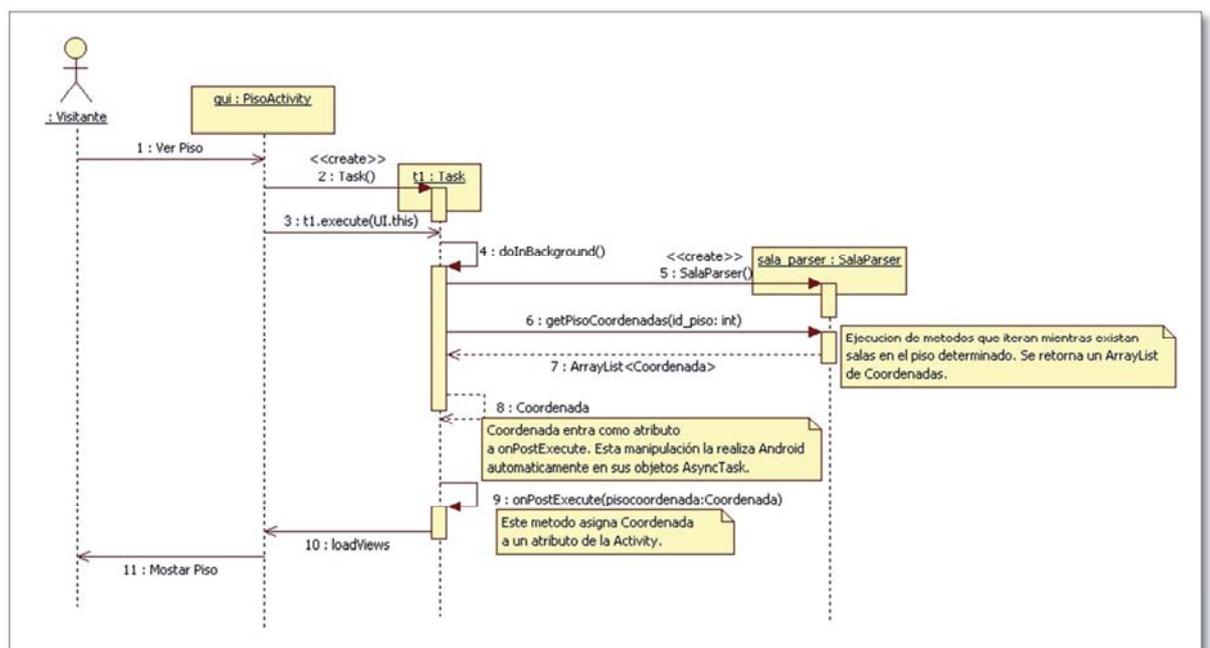


Figura 8.3 – Diagrama de secuencia – Ver Piso

La clase *PisoActivity* es una clase que hereda la clase *Activity*, propia de la API de Android. Estas clases (*Activity*) son clases que al instanciarlas muestran por pantalla los elementos que el programador disponga.

El método *loadViews* pertenece a la clase *PisoActivity* y es ejecutado desde el hilo paralelo cuando este ya ha capturado toda la información proveniente de los *Parsers*. Se encarga de pintar en la pantalla del dispositivo la imagen solicitada y los botones situados sobre ella.

8.1.5 Diagrama de secuencia - Ver Panorámica

Este diagrama representa la secuencia de acciones necesarias para poder mostrar una vista panorámica de una sala determinada por el usuario.

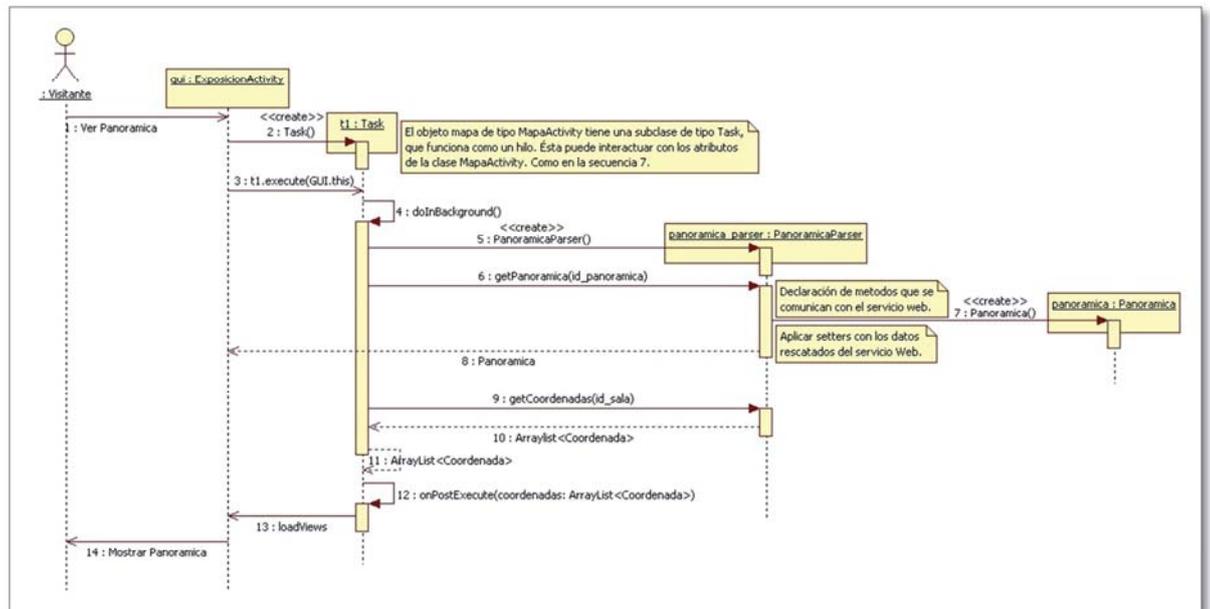


Figura 8.4 – Diagrama de secuencia – Mostrar Panorámica

Para llevar a cabo esta secuencia debe existir una precondition, el id de la panorámica debe ser transferido desde el hilo que mostraba el piso a esta nueva *Activity* (*ExposicionPanoramica*). Esto es posible gracias a la clase *Intent* que posee Android para lograr la comunicación entre *Activities* distintas.

8.1.6 Diagrama de secuencia –Ver Exposición

Este diagrama representa la secuencia de acciones necesarias para mostrar una exposición en la pantalla del Tablet.

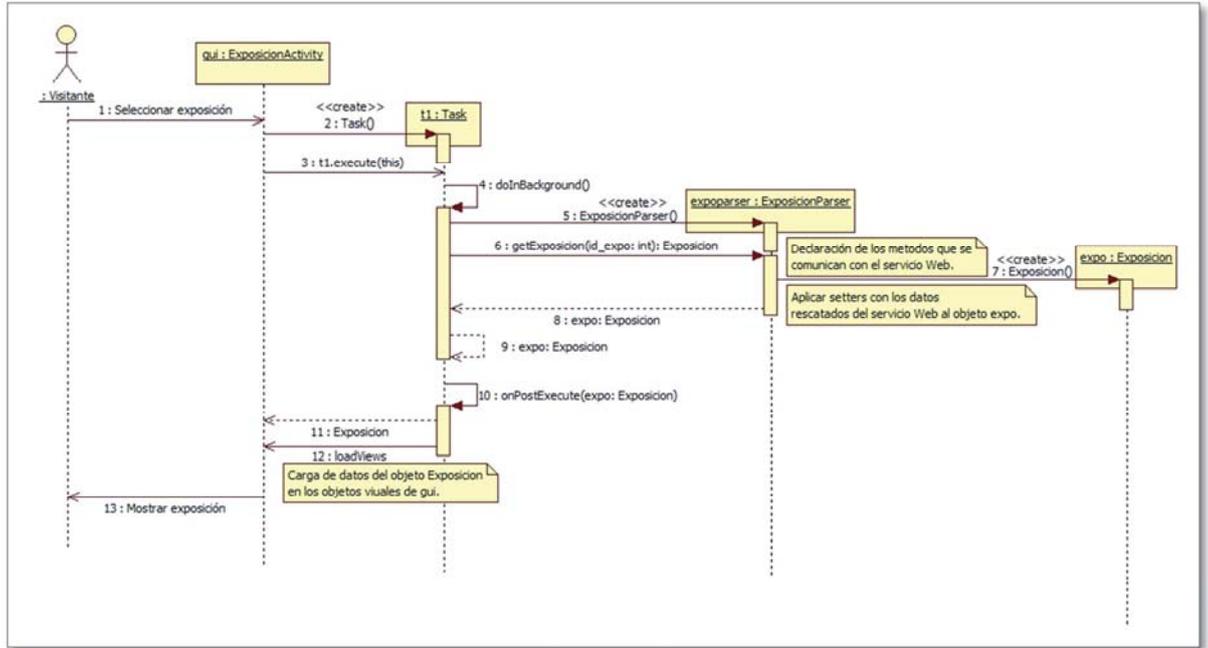


Figura 8.5 – Diagrama de secuencia – Mostrar Exposición

Este es el diagrama de clases correspondiente al cliente móvil. En este no se muestran las clases de las interfaces gráficas ni las clases Activities, ya que este diagrama está acotado a mostrar las clases que necesita el sistema para poder funcionar.

8.2 Guía Móvil – Módulo Realizar Recorrido

El desarrollo del segundo incremento consiste en el análisis y diseño del módulo “Realizar recorrido”. Este modulo tiene como fin crear la parte del software en la que el visitante puede realizar un recorrido virtual del museo a través del cliente móvil.

Para este incremento es necesario realizar el diagrama de casos de uso correspondiente, los diagramas de secuencia necesarios, la construcción del modulo y poblar la base de datos de algunas grabaciones de prueba. Este último punto requiere de la directa intervención del museo naval, ya que, la tarea de realizar las grabaciones correspondientes es de esta entidad.

8.2.1 Diagrama de casos de uso Realizar recorrido

El siguiente diagrama de casos de uso muestra la interacción del visitante accediendo a la opción “Realizar recorrido”. Esta tiene el objetivo de guiarlo a través de un recorrido virtual por el museo. La guía cuenta con controles para moverse hacia delante y hacia atrás dentro del circuito de salas con la posibilidad de escuchar un audio descriptivo de ella.

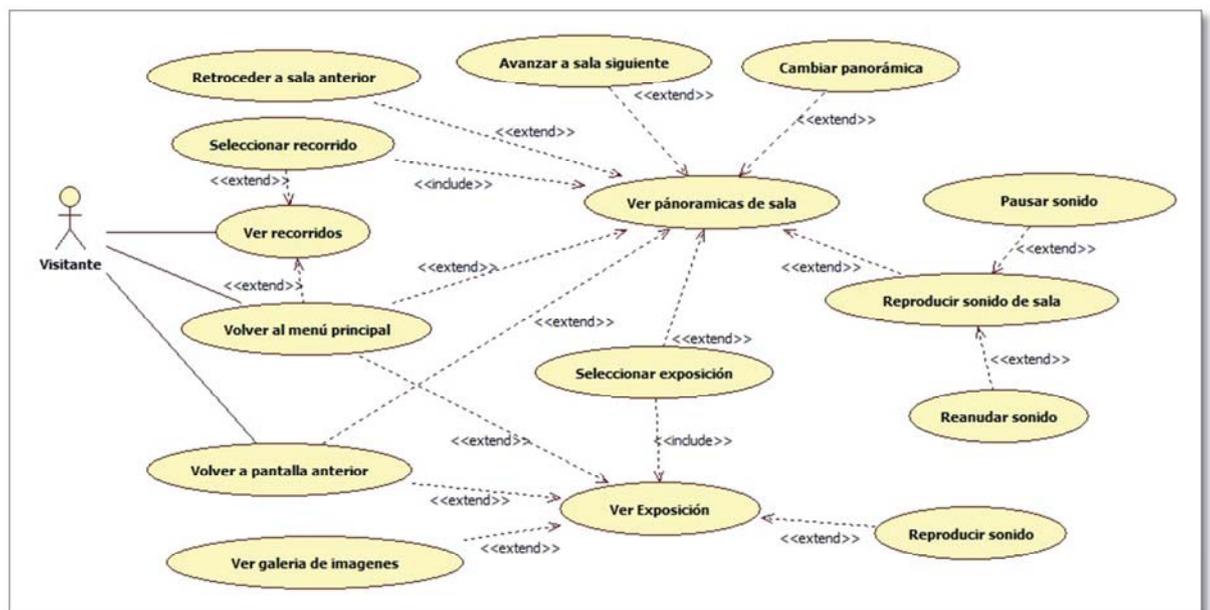


Figura 8.6 - Diagrama de casos de uso Realizar recorridos

A continuación la forma narrativa de este diagrama.

Tabla 8.2 – Caso de uso narrativo – Realizar recorrido

| | |
|-------------------------------|---|
| Nombre del caso de uso | Realizar recorrido |
| Actores | Visitante |
| Propósito | Realizar un recorrido virtual guiado por grabaciones a través de las distintas salas del museo. |
| Descripción | El visitante al tener el dispositivo en su mano selecciona la opción “Realizar |

| | | |
|------------------------|---|---|
| | recorrido”. Esto muestra una lista de los recorridos creados dentro del sistema mantenedor, uno de estos puede ser seleccionado por el visitante. Posteriormente se expone por pantalla las panorámicas de la primera sala del recorrido. En este punto el visitante puede avanzar a la siguiente sala del recorrido con un botón y también reproducir el sonido asociado a la sala. Estando dentro de la sala, el visitante puede seleccionar las exposiciones que estime convenientes y ver una descripción de ellas. | |
| Flujo principal | Paso | Acción |
| | 1 | El visitante selecciona la opción “Realizar recorrido” en el menú principal. |
| | 2 | El sistema muestra una lista de recorridos disponibles. |
| | 3 <extends> | El visitante selecciona una de las salas para comenzar el recorrido. |
| | 4 | El visitante puede seleccionar un botón para reproducir el sonido asociado a la sala. |
| | 5 | El sistema muestra una vista panorámica de la sala. |
| | 6 <extends> | El visitante avanza a la sala siguiente |
| | 7 <extends> | El visitante retrocede a la sala anterior |
| | 8 <extends> | El visitante selecciona una exposición. |
| | 9 | El sistema muestra información de la exposición seleccionada. |
| Extensiones | 2.1 - 5.1 - 9.1 | El visitante puede volver al menú principal. |
| | 5.1 - 9.1 | El visitante puede volver a la pantalla anterior |
| | 4.1 | El visitante puede pausar la reproducción del sonido de guiado. |
| | 4.2 | El visitante puede reanudar la reproducción del sonido de guiado. |
| | 4.3 | El visitante puede cambiar la imagen panorámica mostrada por otra de las que estén disponibles. |
| | 5.2 | El visitante puede cambiar a otra panorámica, siempre y cuando la sala cuente con más de una. |
| | 5.3 | El visitante puede avanzar a la sala siguiente para continuar con el recorrido. |
| | 5.4 | El visitante puede retroceder a la sala anterior. |
| | 6.2 | El visitante puede ver una galería de imágenes, siempre y cuando la exposición cuente con una. |
| | 6.3 | El visitante puede reproducir un sonido correspondiente a la exposición. |

8.2.2 Diagrama de secuencia Cargar Recorrido

Este diagrama de secuencia muestra la secuencia de acciones necesarias para que el visitante visualice un recorrido previamente seleccionado. El objeto recorrido_parser: RecorridoParser cuenta con un método “llenarOrden”, el cual se encarga de recoger el orden de la salas dentro del recorrido en la base de datos y lo carga dentro del objeto. A través de métodos estáticos para obtener ese orden, es posible obtener el orden determinado al cambiar de activities (Instancia en la cual se instancian nuevamente la mayoría de la clases de la Activity).

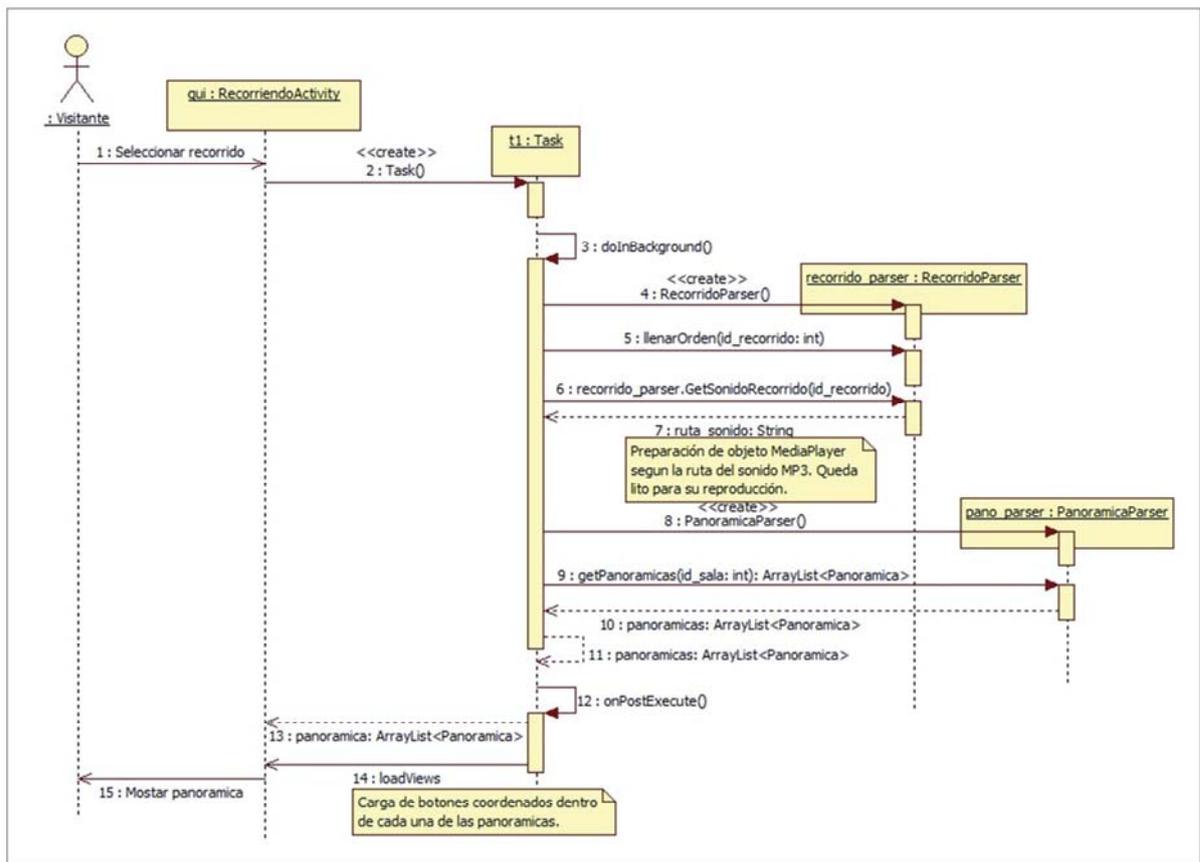


Figura 8.7 – Diagrama de secuencia – Cargar Recorrido

8.3 Guía Móvil – Ingresar código exposición

El tercer incremento de la guía móvil corresponde al desarrollo del modulo “Ingresar código”. El desarrollo de este incremento tiene como fin construir la parte del software en la que el usuario puede acceder directamente a la información de una determinada exposición a través del ingreso de un código. Este está impreso en cada una de las exposiciones del museo, con el objetivo de ser vistos por los visitantes para ingresarlos en esta parte del software. Los códigos son generados automáticamente por el sistema mantenedor y es responsabilidad del museo posicionarlos encima de cada exposición.

8.3.1 Diagrama de casos de uso Ingresar código de exposición

El siguiente diagrama de casos de uso muestra la interacción del visitante cuando accede a la opción “Ingresar código”. Esta tiene como objetivo mostrar información de una exposición dado un código ingresado por el visitante.

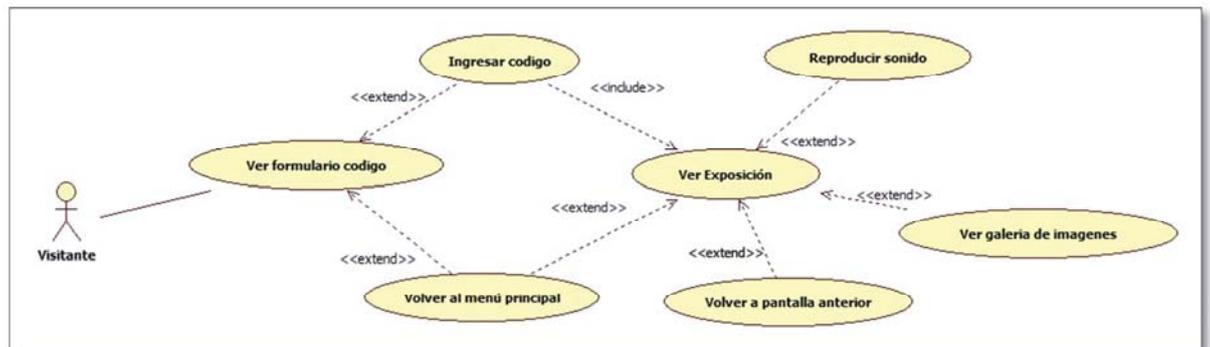


Figura 8.8 - Diagrama de casos de uso Ingresar código

A continuación la forma narrativa de este diagrama.

Tabla 8.3 – Caso de uso narrativo – Ingresar código

| | | |
|-------------------------------|--|---|
| Nombre del caso de uso | Ingresar código de exposición | |
| Actores | Visitante | |
| Propósito | Acceder a una exposición determinada dado un código ingresado por el visitante. | |
| Descripción | El visitante al tener el dispositivo en su mano selecciona la opción “Ingresar código”. Esto mostrará un formulario en cual el visitante debe ingresar un código provisto de alguna de las exposiciones del museo. | |
| Flujo principal | Paso | Acción |
| | 1 | El visitante selecciona la opción “Ingresar código” en el menú principal. |
| | 2 | El sistema muestra un formulario numérico. |
| | 3 | El visitante ingresa un código obtenido visualmente de alguna exposición dispuesta en el museo. |
| | <extends> | |
| | 4 | El sistema muestra información de la exposición correspondiente al código ingresado. |
| Extensiones | 2.1 – 4.1 | El visitante puede volver al menú principal. |
| | 4.1 | El visitante puede volver a la pantalla anterior |

| | | |
|--|-----|--|
| | 4.2 | El visitante puede ver una galería de imágenes, siempre y cuando la exposición cuente con una. |
| | 4.3 | El visitante puede reproducir un sonido correspondiente a la exposición. |

Este incremento es sencillo ya que utiliza “ExposicionActivity” (Figura) posterior al ingreso de un determinado código. El desarrollo de la interfaz gráfica del panel numérico para ingresar el código de alguna exposición es propio de la API del SDK de Android.

8.3.2 Diagrama de secuencia – Ingresar código

Este diagrama de secuencia representa la secuencia de acciones e invocaciones necesarias para que al momento de que el visitante ingrese el código de una exposición determinada, esta se muestre.

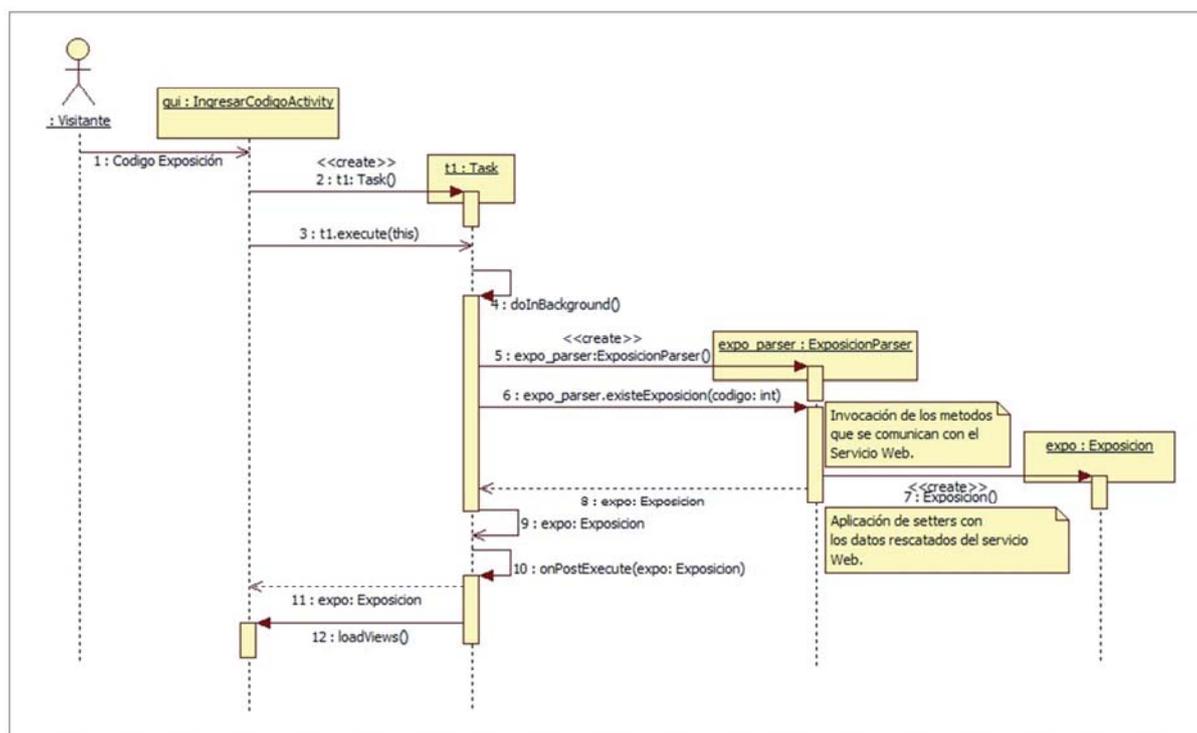


Figura 8.9 – Diagrama de secuencia Ingresar código.

8.3.3 Diagrama de clases

La clase Parser y las que la extienden son las encargadas de comunicarse con el servicio Web RestFul para solicitar datos desde la base de datos.

La clase Ayudante cuenta con un solo método que retorna un objeto Drawable a partir de un String url.

Por otro lado, la clase Coordinada guarda las coordenadas x e y correspondientes a los botones que se deben situar encima del mapa del piso uno o dos y encima de las vistas

panorámica correspondientes a las exposiciones. Esta clase cuenta con el atributo destino, el cual guarda el Id del destino dependiendo de si el objeto fue instanciado desde la clase PanorámicaParser o SalaParser.

Las demás clases son las que interactúan con la interfaz gráfica para mostrar los mapas, panorámicas e información de las exposiciones.

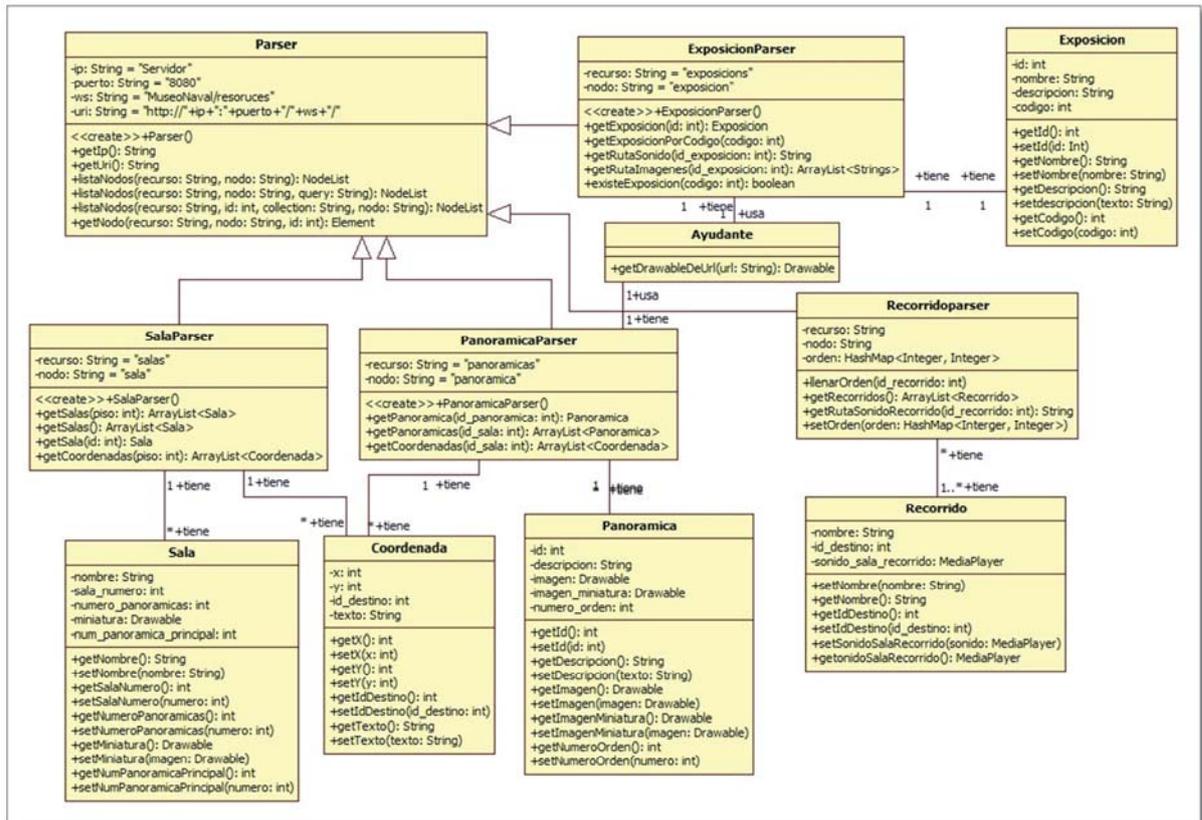


Figura 8.10 - Diagrama de clases Guía Móvil

8.4 Subsistema mantenedor Web

Con el fin de poder gestionar la información que es mostrada en la guía móvil, fue necesario construir un mantenedor el cual permita la edición de los contenidos en la guía móvil. Para desarrollar este sistema, según la metodología adoptada, se debió llevar a cabo el primer incremento de este. Su objetivo es gestionar y organizar adecuadamente los recorridos, exposiciones, panorámicas, salas y pisos del museo mediante una interfaz Web que puede ser accedida desde cualquier computador del museo. El actor encargado de manejarlo es un “Administrador” único, el cual tiene un usuario y contraseña para acceder a él. La construcción de este sistema (Mantenedor Web) se construyó utilizando el *framework* PHP Symfony 1.4 y estará montado sobre un servidor HTTP Apache.

Durante el desarrollo de este incremento se optó por cambiar el paradigma con el que estuvo siendo construido este sistema. El nuevo paradigma utilizado es el de la Orientación a objetos, ya que permite realizar un diseño ordenado y escalable en el tiempo. En consecuencia, fue necesario rediseñar completamente el subsistema Mantenedor Web.

8.4.1 Requerimientos

El subsistema mantenedor Web tiene la función de gestionar y organizar adecuadamente toda la información que ocupa el cliente móvil que es mostrada a los visitantes del museo. A continuación se detallan los requerimientos del sistema y posteriormente se detallan a través de los modelos de casos de usos.

8.4.1.1 Requerimientos funcionales del sistema

- El sistema deberá contar con módulos para administrar los pisos, salas, panorámicas, exposiciones y recorridos del museo.(visible)
- Cada uno de los módulos deberá permitir agregar, eliminar y editar. (visible)
- Para que una sala se encuentre activada dentro de los dos sistemas; edición y mostrar, para el mantenedor y la guía respectivamente, deben contar con una o más imágenes panorámicas asociadas a ella (a la sala). (visible)
- El sistema deberá permitir, al agregar un nuevo recorrido, realizar el orden del circuito que se pretende que el visitante recorra. (visible)
- Si una sala que se encuentre dentro de un determinado recorrido es desactivada por alguna razón, el o los recorridos a los cuales ésta pertenece, deberá ser desactivado.(oculto)
- El sistema deberá permitir situar sobre una imagen otra más pequeña, indicando donde se ubicarán las salas en un determinado piso y las exposiciones en una determinada panorámica.(visible)
- El sistema deberá permitir agregar imágenes a una determinada exposición.(visible)
- El sistema deberá permitir agregar sonidos a las salas, para ser usados en los recorridos. Además deberá permitir agregarlos en las exposiciones.
- Las imágenes de los pisos deberán ser de una dimensión de 800x480 px. (oculto)

- Las imágenes de las panorámicas deberán ser de una dimensión de 800x415 px. (oculto)
- Una sala no estará activada hasta que tenga al menos una panorámica asociada.(visible)

8.4.1.2 Requerimientos no funcionales del sistema

- El sistema tendrá un menú contextual el cual permitirá al Administrador moverse entre los distintos módulos del sistema.
- El modulo de salas deberá mostrar todas las salas de un determinado piso, con opciones para editar y eliminar. Además, contará con un acceso rápido a sus panorámicas y exposiciones asociadas.
- El módulo de recorrido deberá mostrar una columna izquierda con las exposiciones disponibles para armarlo y una derecha con las exposiciones ordenadas del mismo.
- El sistema deberá manejar una iconología visualmente descriptiva, tal que el administrador pueda comunicarse con el sistema de manera intuitiva.

8.4.2 Diagramas de Casos de Usos

Los diagramas de casos de uso definen el comportamiento e interacción que tiene el actor con el sistema. En este, el actor es el Administrador el cual puede acceder a una serie de opciones que se detallan a continuación a través de los diagramas de casos de uso.

8.4.2.1 Diagrama de casos de uso de alto nivel

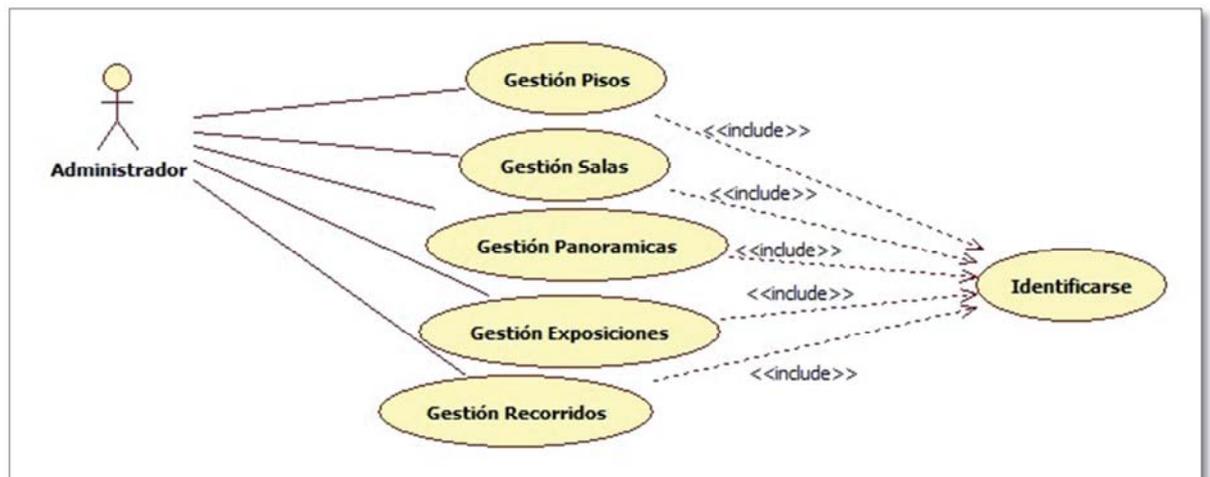


Figura 8.11 - Diagrama de casos de uso de alto nivel

En la figura 8.11 se pueden apreciar las distintas operaciones que puede realizar el Administrador dentro del sistema mantenedor.

8.4.2.2 Diagrama de casos de uso - Gestión Pisos

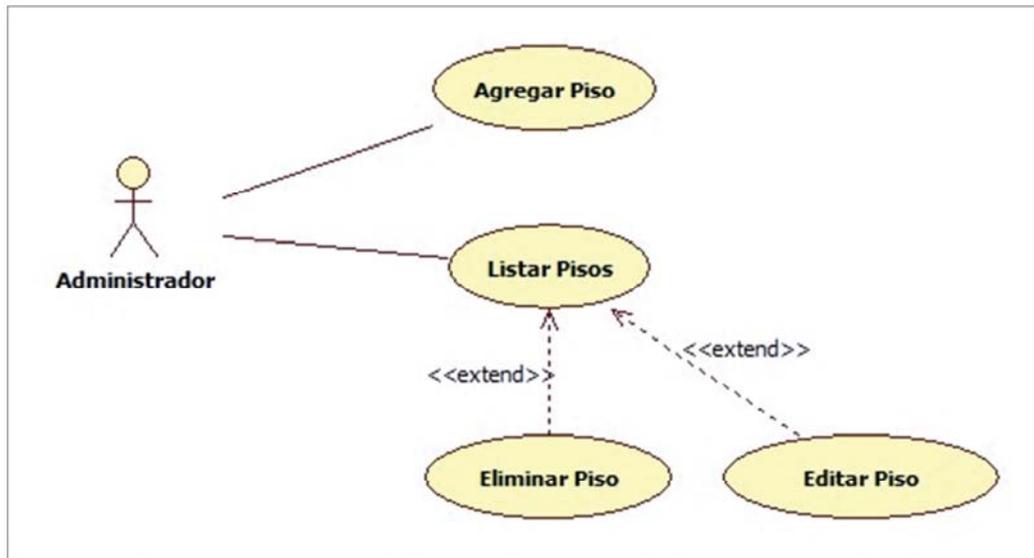


Figura 8.12 – Diagrama de casos de uso: Gestionar Piso

En la figura 8.12 se ve la explotación del caso de uso gestionar recorrido, se muestran las principales funciones de este modulo: Nuevo recorrido, listar recorridos, ver recorrido, eliminar recorrido y modificar recorrido.

8.4.2.3 Diagrama de casos de uso - Gestión Sala

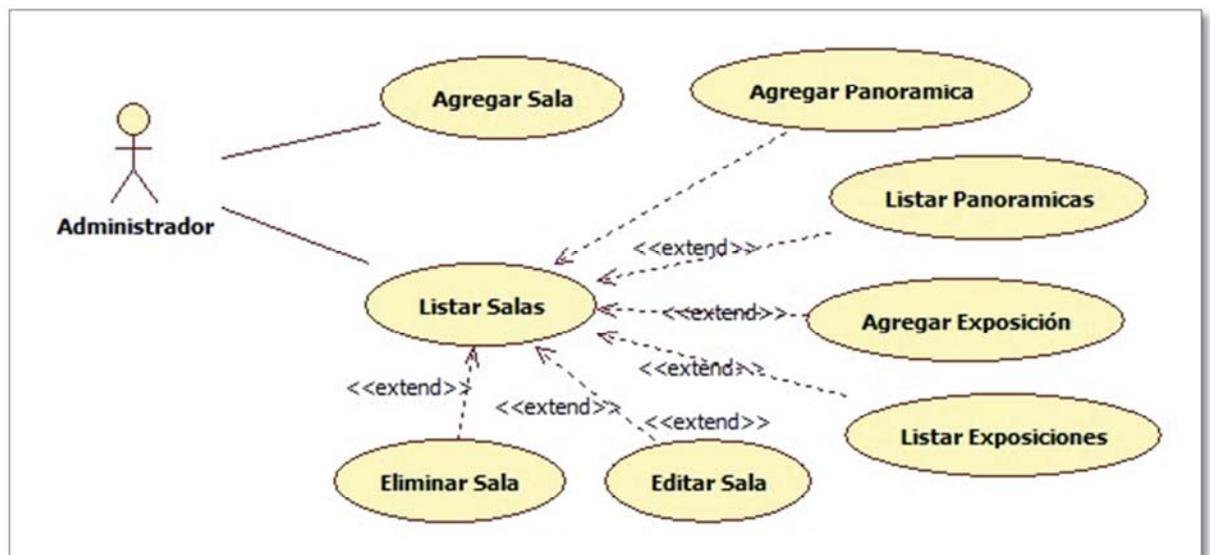


Figura 8.13 – Diagrama de casos de uso: Gestionar Sala

En la figura 8.13 se muestra el caso de uso que corresponde a gestión sala, donde se ven las funciones de este módulo como: agregar, editar, eliminar y listar salas. Al ingresar a esta última opción se debe poder acceder desde ese mismo lugar a las opciones “Agregar panorámica” a la sala y “Lista panorámicas” de la misma. Además de poder acceder a la opción “Agregar exposición” a la sala y “Listar exposiciones” de la sala.

8.4.2.4 Diagrama de casos de uso - Gestión panorámica

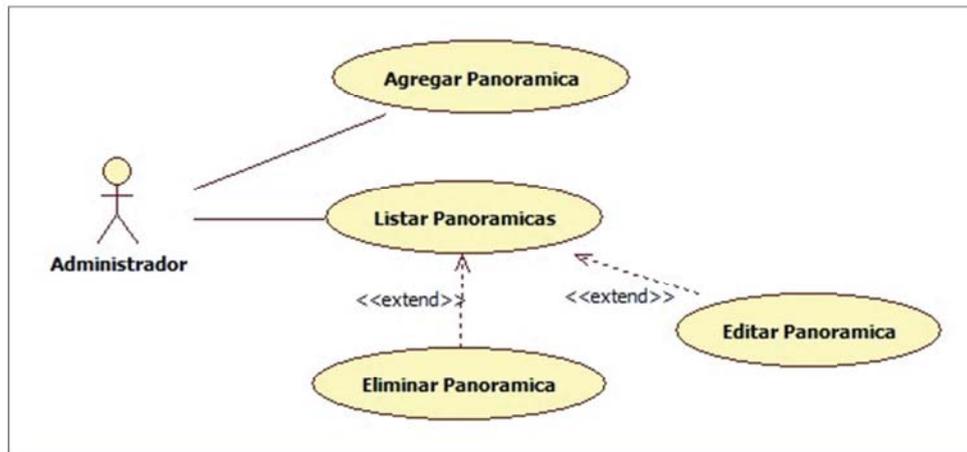


Figura 8.14 - Diagrama de casos de uso: Gestionar panorámica

En la figura 8.14 se muestra el caso de uso que corresponde a gestionar panorámica, se ven las funciones donde el Administrador tiene interacción como: agregar, modificar y eliminar panorámica.

8.4.2.5 Diagrama de casos de uso - Gestión Exposición

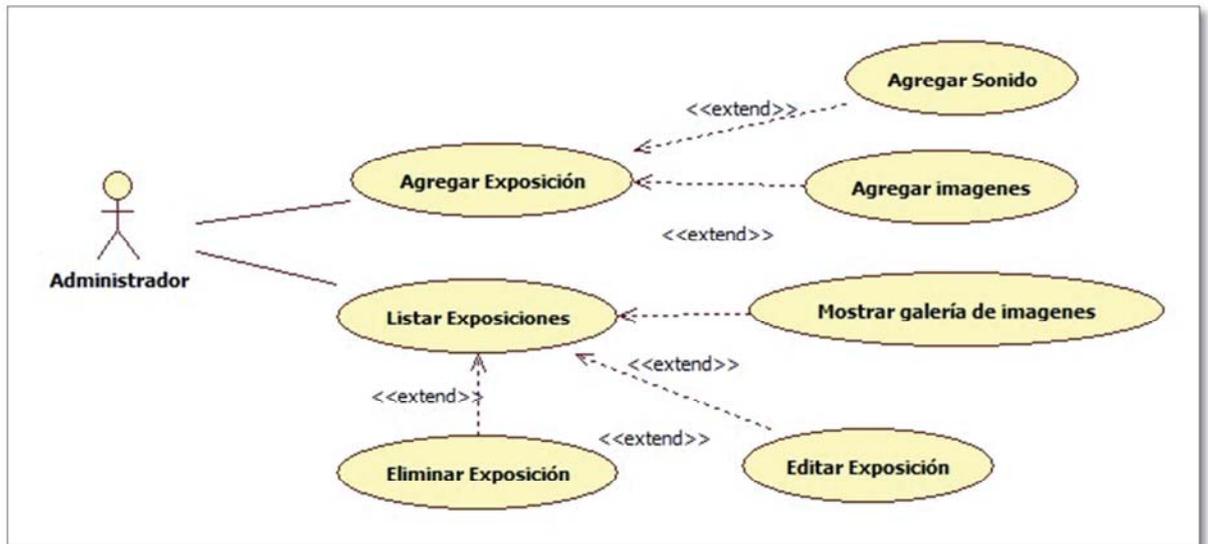


Figura 8.15 - Diagrama de casos de uso: Gestionar exposición

En la figura 8.15 se muestra el caso de uso que corresponde a gestionar exposición en donde se ven las funciones o módulos donde el usuario tiene interacción como: agregar, modificar, eliminar. Dentro del caso de uso agregar el Administrador debe tener la opción de agregar imágenes a la exposición, así como un sonido referente a ella.

8.4.2.6 Diagrama de casos de uso – Gestión de recorridos

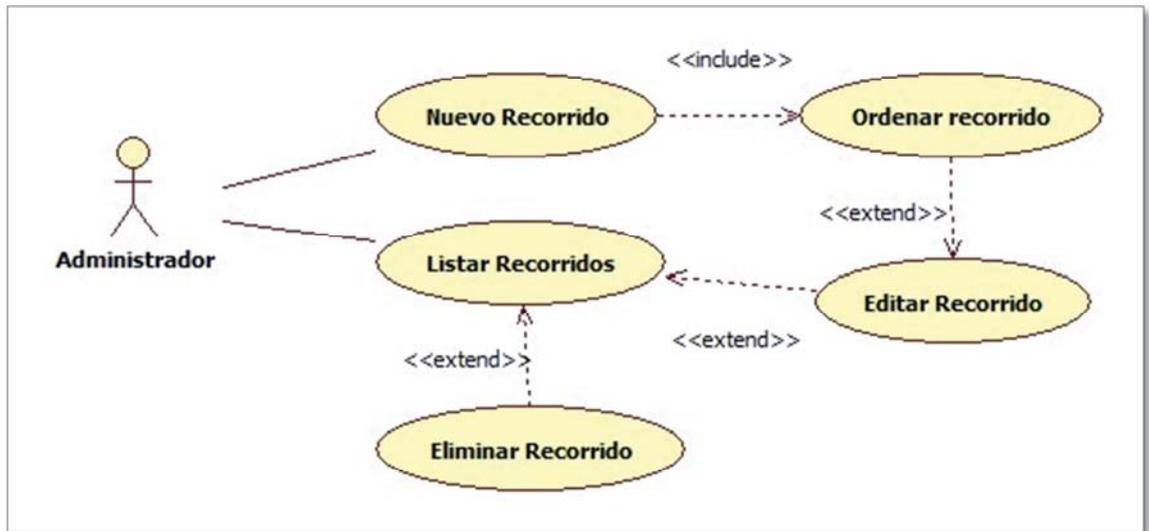


Figura 8.16 - Diagrama de casos de uso: Gestionar Recorridos

En la figura 8.16 se pueden apreciar las opciones que tiene acceso el Administrador para gestionar los recorridos, el cual, al igual que los demás cuenta con las opciones agregar, editar, listar y eliminar recorridos. Además debe contar con la opción de ordenar las salas que conforman al recorrido.

8.4.3 Diagramas de secuencia

Para graficar la forma en que interactúan los distintos objetos del sistema para lograr un objetivo específico. Para esto se utilizan los diagramas de secuencia, donde principalmente se muestran los mensajes (invocación a métodos) que se transmiten de un objeto a otro.

En los siguientes diagramas se muestran generalmente los objetos vista, controlador y el objeto correspondiente a la lógica

8.4.3.1 Diagrama de secuencia Mantenedor Web – Agregar Piso

Este diagrama de secuencia muestra la secuencia de acciones que se deben realizar para agregar un Piso al sistema. La misma secuencia se repite para agregar Salas, Panorámicas y Exposiciones. Se debe tener en cuenta que al agregar una sala, esta estará desactivada hasta que se le asocie por lo menos una imagen panorámica.

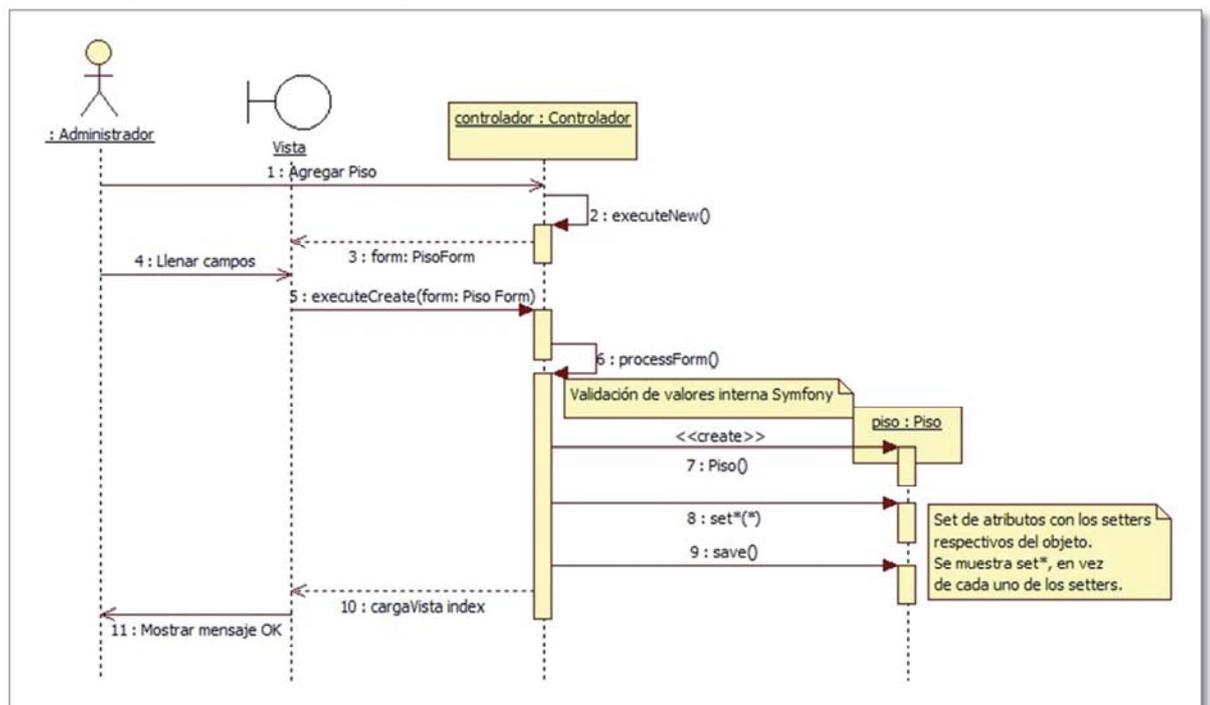


Figura 8.17 – Diagrama de secuencia Agregar Piso

8.4.3.2 Diagrama de secuencia – Mantenedor Web – Agregar Recorrido

Este diagrama de secuencia muestra la secuencia de acciones que se deben realizar para agregar un recorrido. Este diagrama varía del anterior ya que se deben realizar validaciones particulares.

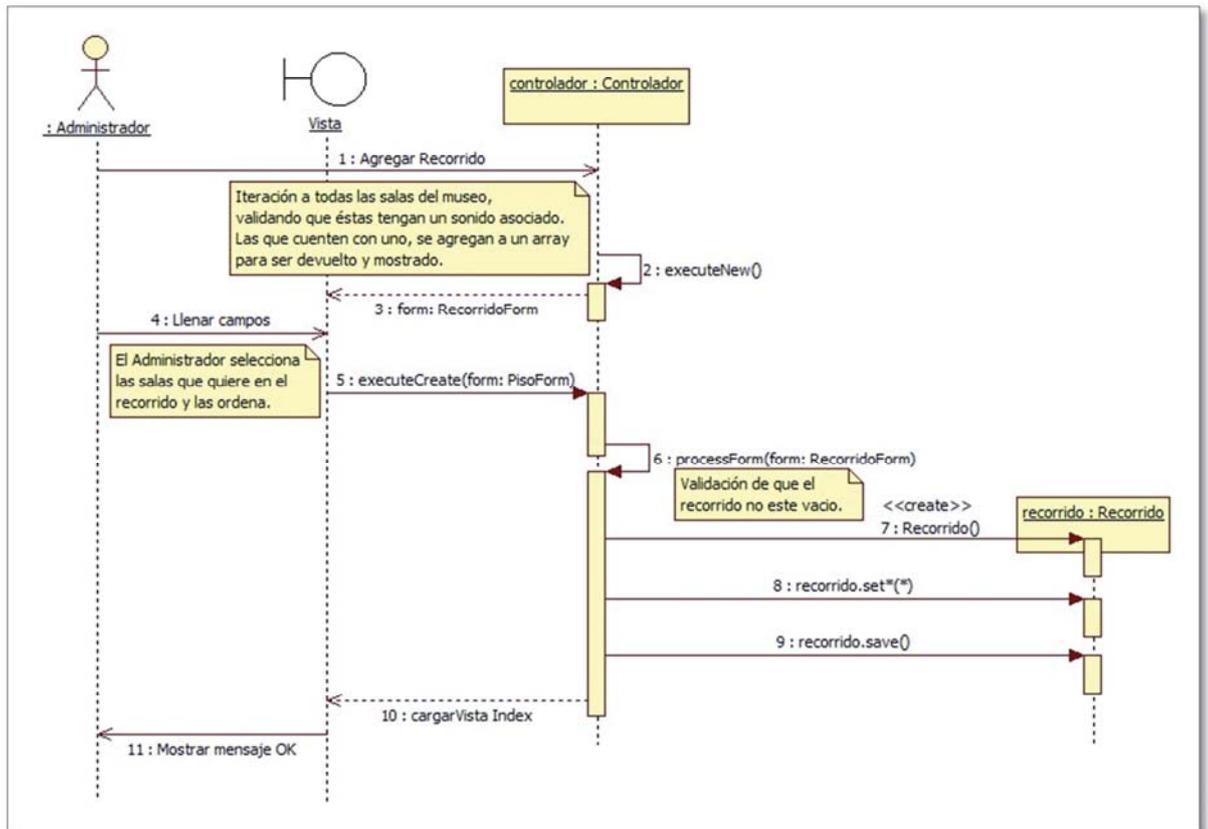


Figura 8.18 – Diagrama de secuencia Agregar Recorrido.

8.4.4 Diagrama de clases

El diagrama de clases muestra de qué forma están relacionadas las distintas clases del sistema, sus respectivos atributos y métodos. Además muestra cuan accesible son cada uno de estos.

Ya que el *framework* de trabajo utilizado es Symfony 1.4, las clases de este sistema fueron creadas automáticamente según el modelo de la base de datos y en relación al patrón Modelo-Vista-Controlador, explicado con anterioridad. Symfony, crea muchas clases por cada tabla de la base de datos del sistema. Estas son:

- Clases Modelo: Por cada tabla de la base de datos se crean dos de estas clases. La primera, cuenta con los métodos necesarios para comunicarse directamente con la base de datos, a través del ORM Doctrine. La segunda, es una clase que hereda de la anterior y no cuenta con ningún método ni atributo. Dentro de esta clase se pueden escribir métodos personalizados.
- Clases Formulario: Por cada tabla de la base de datos se crean dos de estas clases. La primera, crea métodos y atributos automáticamente. Los atributos corresponden a los campos y los métodos tienen el objetivo de validar y agregar

restricciones a los distintos campos para ser mostrados dentro de un formulario en la Vista. La segunda, es una clase que hereda de la anterior y permite personalizar el formulario creado por defecto por Symfony.

- Clases Persistentes: Por cada tabla de la base de datos se crean dos de estas clases. La primera, corresponde a una abstracción en forma de clase de la tabla dentro de la base de datos. Sus atributos son los mismos campos de la tabla y los métodos son los clásicos *getters* y *setters*. Los atributos y métodos de esta clase son creados automáticamente por el *framework*. La segunda corresponde a una clase que hereda de la anterior y permite agregar métodos y atributos a la clase creada por defecto. Estas últimas clases mencionadas, corresponden a las que se instanciarán para ser utilizadas dentro del sistema.

En este diagrama de clases sólo se grafican las clases persistentes, ya que, estas son las que se relacionan directamente con el funcionamiento del sistema. Las demás responden a un funcionamiento interno de Symfony.

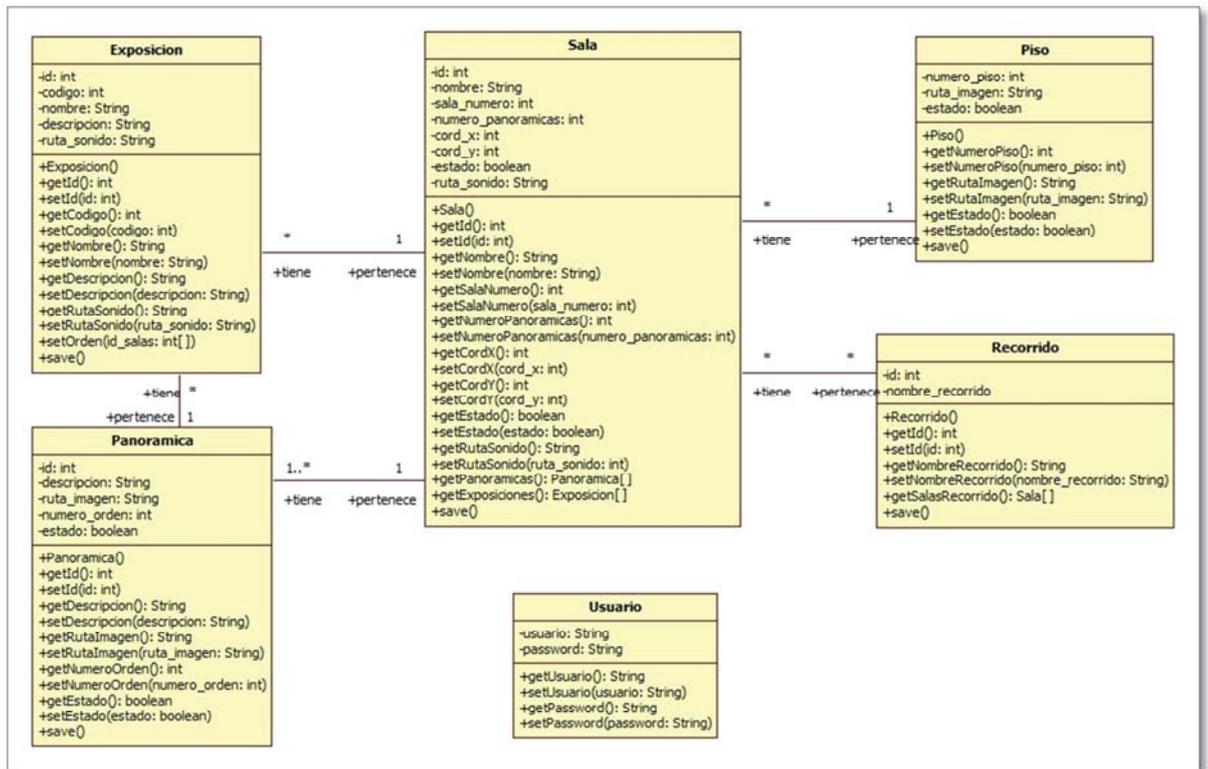


Figura 8.19 - Diagrama de clases

En la figura anterior se pueden apreciar las clases Piso, Sala, Panorámica, Exposición, Recorrido y Usuario. Estas cuentan con los mismos atributos que los campos de las tablas dentro de la base de datos. Los métodos son *getters* y *setters* de los atributos, pero también se agregaron algunos métodos necesarios para el funcionamiento del sistema como *getPanoramicas()* y *getExposiciones()* de la clase Sala.

8.5 Arquitectura física del sistema

El sistema completo necesita de hardware apropiado para operar adecuadamente, es por esto que se instalarán servidores y clientes. La figura 8.1 muestra el diagrama de arquitectura física del sistema completo. En él, se puede apreciar una maquina con cuatro servicios ejecutados (Servidor HTTP Apache, Servidor GalsshFish, Servidor PHP y Servidor PostgreSQL), un router, un punto de acceso *wifi*, un servicio Web RestFul y dos clientes, uno para el mantenedor y el otro para la guía móvil. Debido al grosor de las paredes del museo naval, es posible que la señal wifi deba ser “repetida” por repetidores wifi donde esta sea débil.

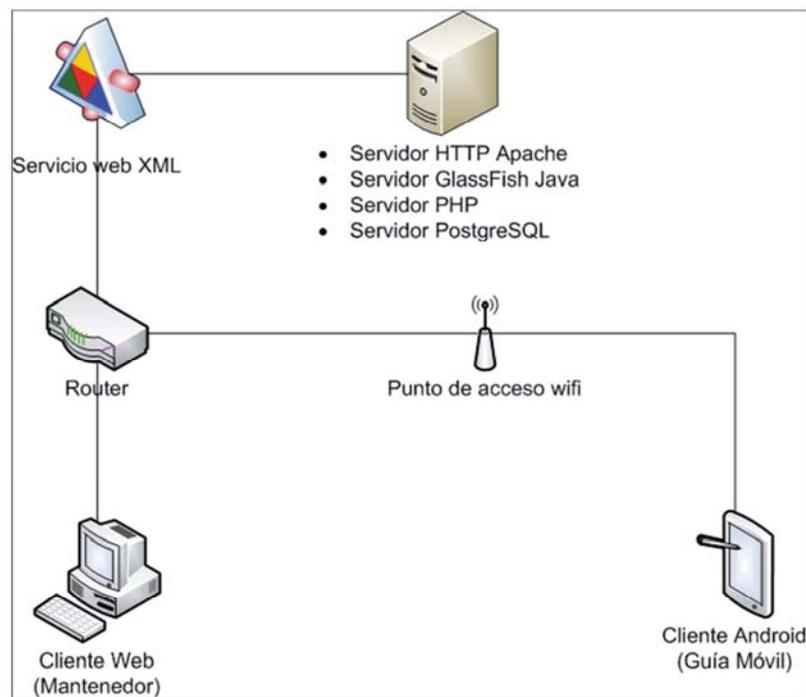


Figura 8.20 – Diagrama de arquitectura física del sistema

Puede llamar la atención el hecho de que todos los servicios estén instalados en una misma máquina, pero esto no es un problema ya que no es un sistema de alta demanda.

El servicio Web RestFul es impartido por el servidor GlassFish el cual está encargado de ser un mediador entre el cliente móvil y la base de datos. Esto se debe a que Android no posee la capacidad de comunicarse directamente con motores de bases de datos, excepto SQLite.

El cliente mantenedor puede ser visualizado desde cualquier computador conectado a la red establecida por el router.

La guía móvil puede funcionar en *tablets* que posean el sistema operativo Android 1.6 o superior, con una dimensión de 7”, densidad de 160 y que esté conectado a la red impartida

por el punto de acceso wifi. La puesta en marcha se planea contó con seis Tablets Android, proporcionados por el Museo.

8.6 Modelo relacional del sistema

A continuación se muestra el modelo relacional diseñado para la base de datos del sistema. En él se pueden apreciar las tablas con las que cuenta el sistema. Las tablas más importantes son “exposición”, “panorámica”, “sala” y “detalle_recorrido”. Con esta última tabla es posible crear nuevos recorridos para el museo en caso de que éste lo llegue a necesitar en algún momento. Además, guarda el orden en el que deben ser mostradas las salas en tal recorrido. Las tablas “exposición_sonido” y “grabación_recorrido” fueron creadas para usarlas en una siguiente versión del software, en la cual se cuente con multi-idioma. Actualmente la relación es 1:1 entra estas tablas y las exposiciones y salas.

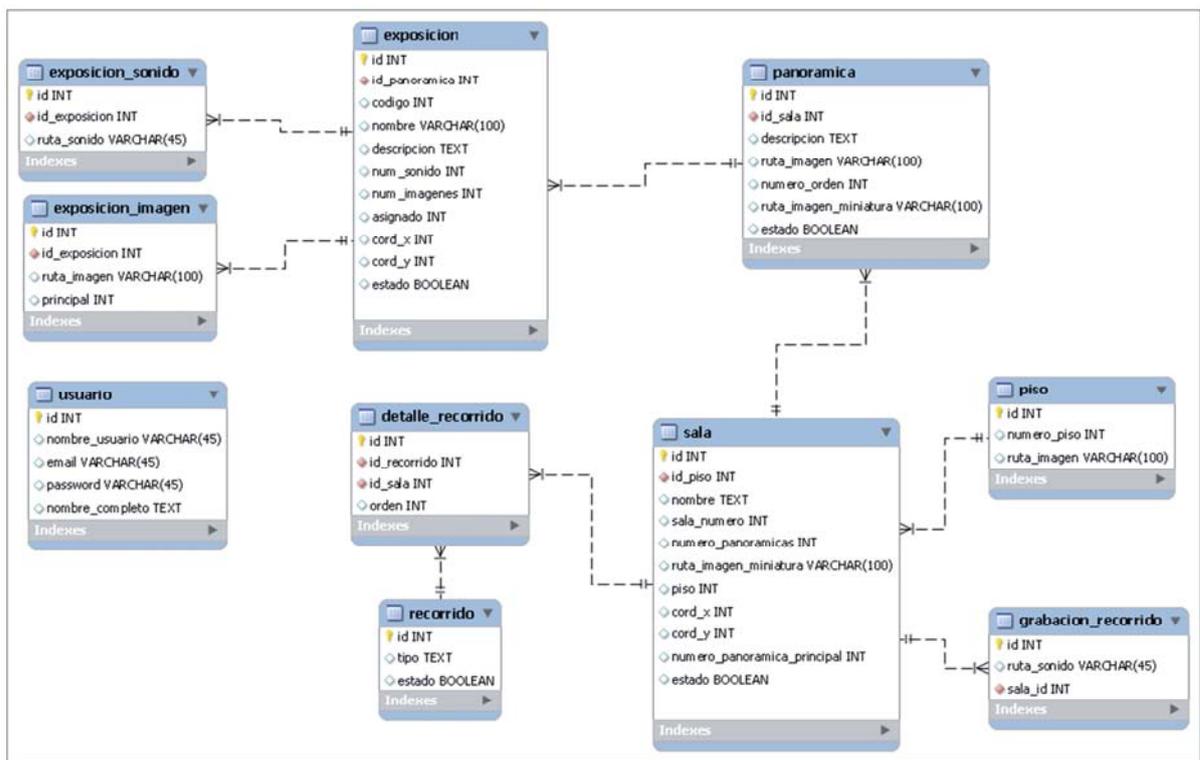


Figura 8.21 – Modelo relacional de la base de datos

8.7 Interfaces gráficas del sistema

A continuación se documentan capturas de pantalla de los dos subsistemas, tal que, se pueda comprender donde esta cada funcionalidad del sistema.

8.7.1 Interfaz grafica Mantenedor Web



Figura 8.22 – Login Mantenedor Web

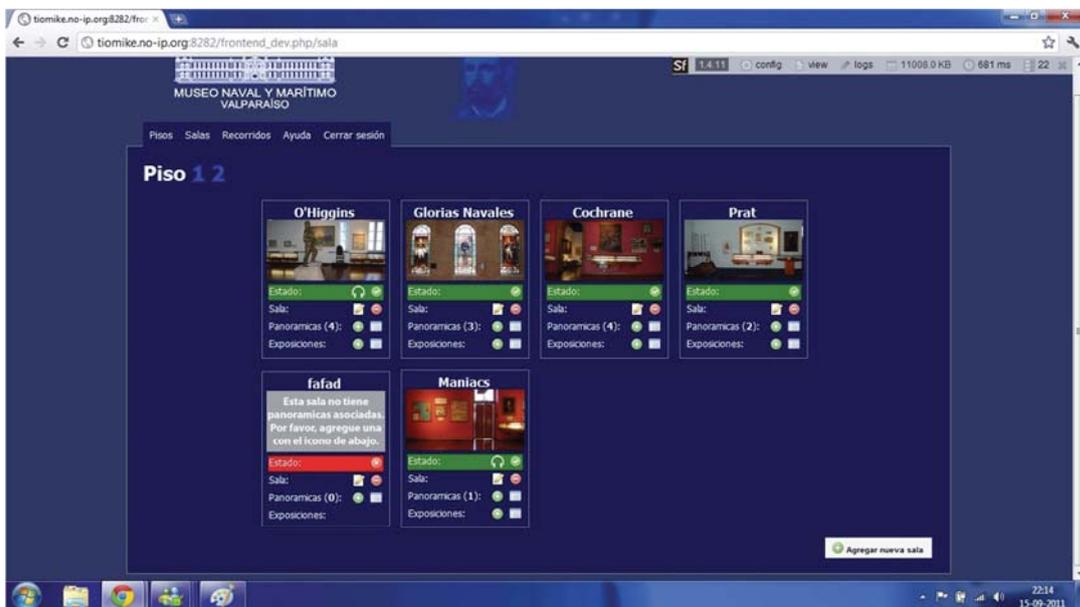


Figura 8.23 – Página principal Mantenedor Web

8.7.2 Interfaz Grafica Guía Móvil

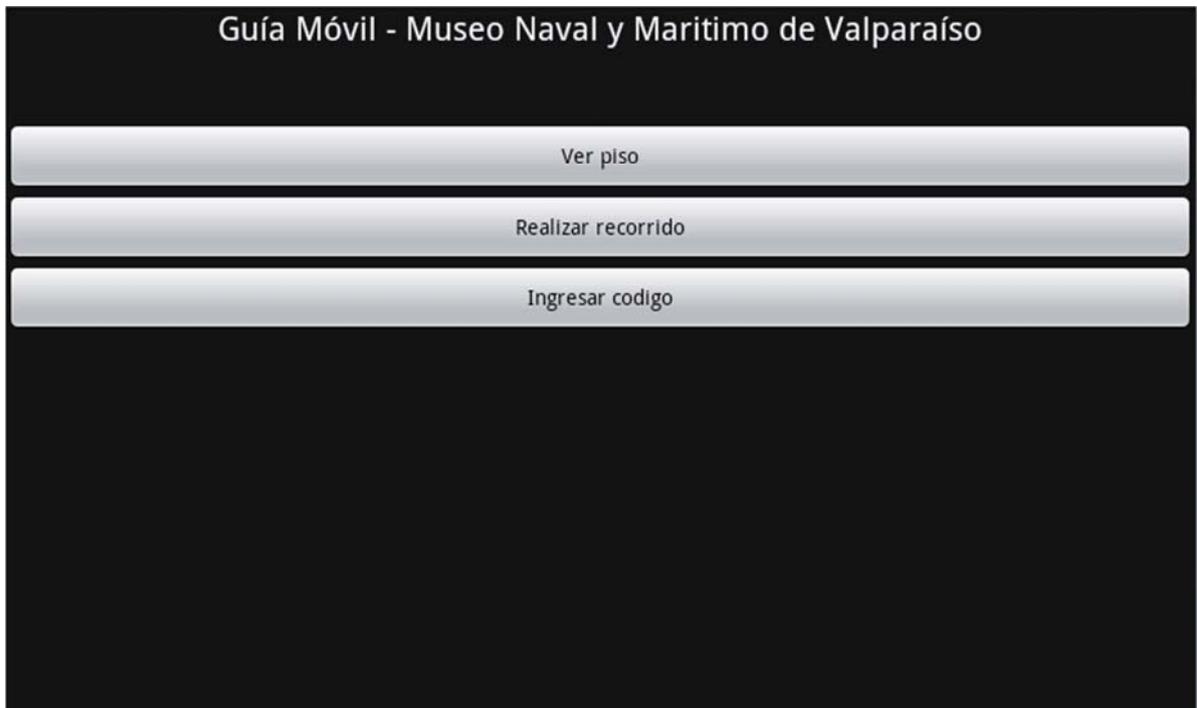


Figura 8.24 – Menú principal

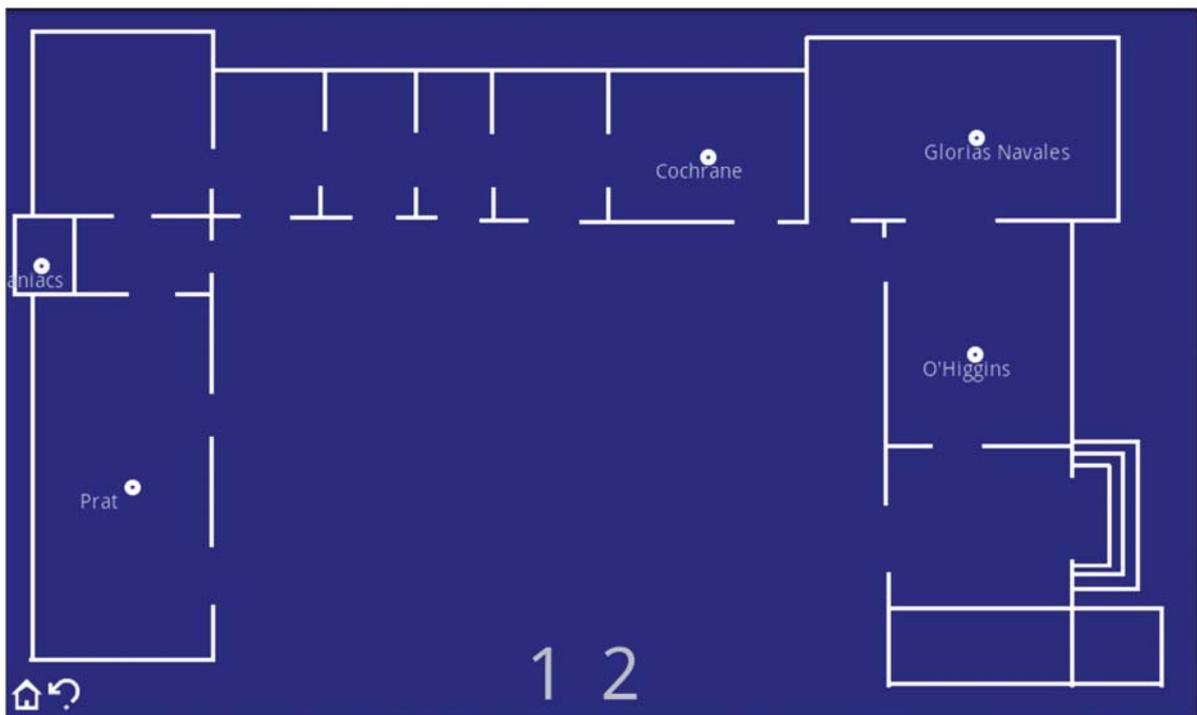


Figura 8.25 – Mostrar pisos



Figura 8.26 – Mostrar Sala

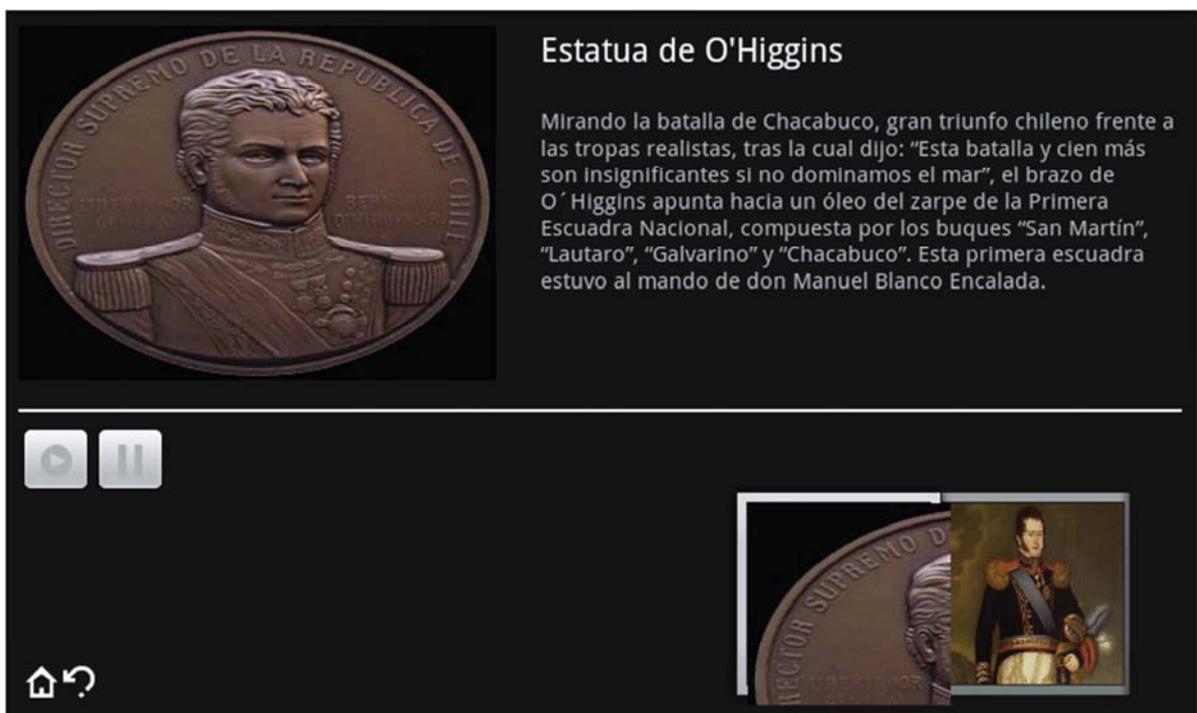


Figura 8.27 – Mostrar Exposición

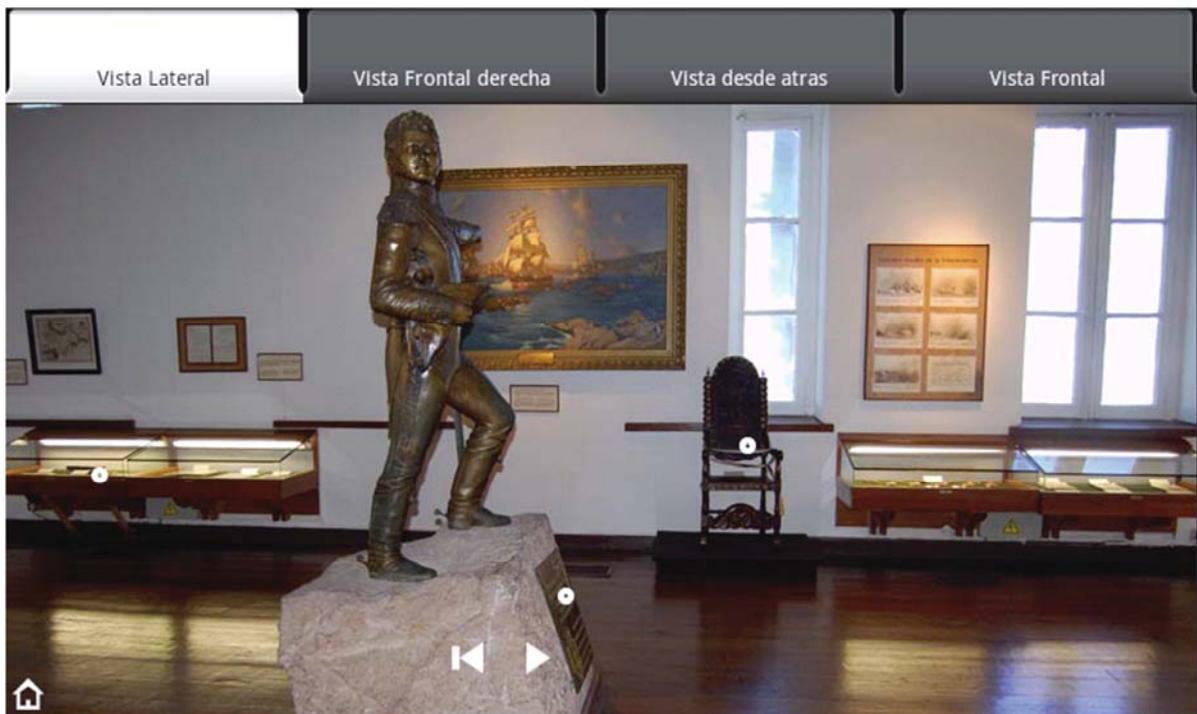


Figura 8.28 – Primera sala de un Recorrido. Esta tiene controles en a parte inferior de la pantalla.

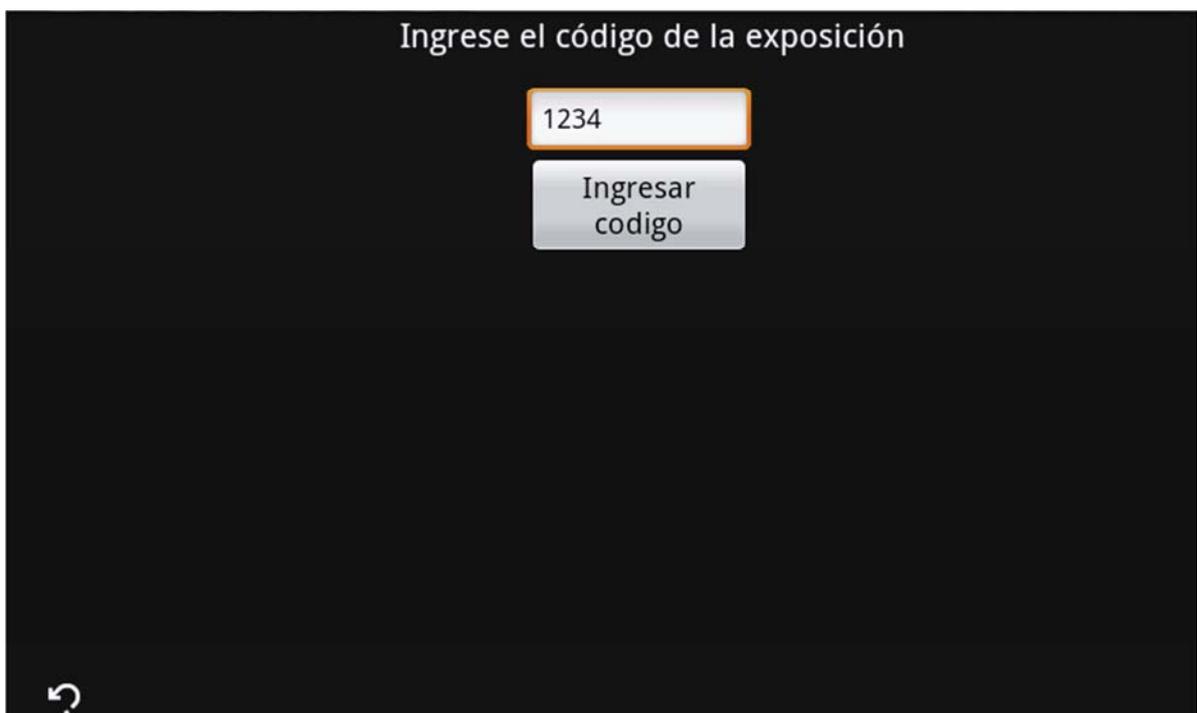


Figura 8.29 – Ingreso de código de una exposición

9 Pruebas de software

Las pruebas de software o *testing* son pruebas realizadas al software construido tal que se pueda asegurar el correcto funcionamiento de este, basándose en los requerimientos funcionales del sistema. El tipo de prueba usada para el sistema fue el de caja negra, en el cual se registran datos de entrada y un resultado esperado.

A continuación se muestra una tabla con una serie de casos:

Tabla 9.1 – Prueba de software

| N° | Subsistema | Módulo | Dato(s) o eventos de entrada | Prerrequisito | Resultado esperado | Resultado de prueba |
|----|----------------|-----------------|---|--|---|---------------------|
| 01 | Mantenedor Web | Sala | Agregar una sala | Existencia de algún piso. | Sala agregada un piso, pero desactivada. | OK |
| 02 | Mantenedor Web | Panorámica | Agregar una Panorámica | Existencia de una sala. | Panorámica agregada a la sala. Esta última se activa de forma automática. La sala puede ser accedida desde la guía móvil | OK |
| 03 | Mantenedor Web | Exposición | Agregar una exposición | Existencia de al menos una panorámica. | Exposición agregada a panorámica. La exposición puede ser accedida desde la guía móvil. | OK |
| 04 | Mantenedor Web | Exposición | Agregar imágenes para la galería | Exposición creada. | Imágenes agregadas y visibles en la descripción de la exposición dentro de la guía móvil. | OK |
| 05 | Mantenedor Web | Recorrido | Agregar un recorrido vacío. | Salas con audio disponible. | Notificación de que se debe contar con al menos una sala para el recorrido. | OK |
| 06 | Mantenedor Web | Recorrido | Editar el orden del recorrido agregado. | Recorrido creado. | Guardado exitoso del orden y plasmado en la guía móvil igualmente. | OK |
| 07 | Mantenedor Web | Piso | Eliminar piso | Debe existir al menos un piso en el sistema. | Toda la jerarquía bajo el piso se debe eliminar. Los recorridos asociados a las salas que sean eliminados deben desactivarse. | OK |
| 08 | Guía Móvil | Ingresar código | Ingresar código inexistente: "9999" | No debe existir una exposición con dicho código. | Notificación de que no existe una exposición con ese código. | OK |
| 09 | Guía Móvil | Exposición | Reproducir sonido y pausarlo | La exposición debe tener un sonido explicativo asociado. | Reproducción y pausa del sonido sin ningún inconveniente. | OK |

| | | | | | | |
|----|------------|-----------|--|--|--|----|
| 10 | Guía Móvil | Recorrido | Ingresar a una exposición mientras se reproduce el sonido del recorrido. | La sala debe tener exposiciones asociadas. | El sonido del recorrido debe pausarse para poder ser reanudado posteriormente. | OK |
|----|------------|-----------|--|--|--|----|

Conclusiones

Todo el trabajo e investigación anteriormente expuestos, retratan que un trabajo ordenado, planificado y responsable puede llevar al éxito de un proyecto informático.

El proyecto se encuentra completamente desarrollado, es decir, tiene el sistema Guía móvil y Mantenedor Web totalmente operativos.

Ahora solo hace falta la implementación en las dependencias del museo, la marcha blanca y la capacitación de los funcionarios.

El museo ganará un tremendo valor agregado implementando el proyecto dentro de él, ya que, por un lado, será el primer museo en todo Chile en contar con una guía móvil con tecnología de punta y, por otro lado, aumentará su capacidad de enseñar la historia naval y marítima de Chile con esta herramienta que proporcionará a los visitantes valiosa información.

En las últimas reuniones con la Directiva del museo se han planteado unas modificaciones al sistema con el fin de contar con soporte para Multi-idioma, soporte para códigos QR y la posibilidad de contar con el software para instalarlo dentro de cualquier dispositivo Android. Estas implementaciones se espera que sean desarrolladas por otros dos alumnos de la Escuela de Informática de la PUCV en su proyecto de título.

El museo, además ha tenido que adquirir una antena omnidireccional de alta potencia de transmisión, ya que la señal *wifi* no llegaba a todas las salas del museo, esto debido al grosor de más de un metro de cada pared del Museo.

Otro requerimiento, que será implementado antes de la puesta en Marcha, es la traducción de la guía móvil al inglés. Esto debido a que el museo planea enfocar la guía en los turistas que visiten el museo. El desarrollador se ha comprometido a crear las traducciones pertinentes al contenido estático de la guía, quedando en manos del profesor a cargo (Profesora de Historia, Pía Rieutord) realizar las traducciones del contenido de cada sala.

El sistema planea ser lanzado el día 21 de Mayo del 2012. Ese día participarán del evento la directiva del Museo Marítimo Nacional, el alumno desarrollador y su profesor guía. Aparte estarán presentes importantes medios de prensa de la zona para documentar el evento en los periódicos o sitios web respectivos.

Este proyecto muestra, además, la importancia que tiene la implementación de nuevas tecnologías a entidades como los museos de Chile, ya que, con estas, es posible enriquecer y explorar las distintas formas de mostrar información. El museo Naval ganará un importante valor agregado al implementar esta tecnología en sus dependencias.

Referencias

- [1] Profesor Antonio Jesús Rueda Ruiz. Universidad de Jaén, Apunte obtenido de su pagina Web (2008). <http://www.di.ujaen.es/asignaturas/progav/progav-tema6.pdf>. Recuperado el 24 de Agosto del 2011.
- [2] Martin Segado. Programación de dispositivos móviles con Visual Studio .Net (2009). [https://www.luarna.com/Documentos%20compartidos/Ejemplos%20de%20lectura/Programaci%20n%20de%20dispositivos%20m%20viles%20con%20Visual%20Studio%20.NET%20\(ejemplo\).pdf](https://www.luarna.com/Documentos%20compartidos/Ejemplos%20de%20lectura/Programaci%20n%20de%20dispositivos%20m%20viles%20con%20Visual%20Studio%20.NET%20(ejemplo).pdf). Recuperado del 10 de Mayo del 2011.
- [3] Sergio Gálvez Rojas y Lucas Ortega Díaz. Java a tope: J2ME (2003). <http://www.lcc.uma.es/~galvez/J2ME.html>. Recuperado el 10 de Mayo del 2011
- [4] Developers Android (2010), Web de Documentación oficial de los desarrolladores Android <http://developer.android.com/>. Recuperado el 10 de Mayo del 2011
- [5] Apple Inc. Información sistema operativo del iPhone. (2010) <http://www.apple.com/es/iphone/ios4/>. Recuperado el 10 de Mayo del 2011.
- [6] Museo Naval y Marítimo de Valparaíso. (2010) <http://www.museonaval.cl/>. Recuperado el 10 de Mayo del 2011
- [7] Guillermo Cabrera. Apuntes Cátedra Ingeniería de Software Unidad 1, tema 5, Carrera Ingeniería de Ejecución Informática, Pontificia Universidad Católica de Valparaíso. (2009). McGraw Hill (2010), *Ingeniería de Software: Un enfoque práctico*.
- [8] Guillermo Cabrera. Apuntes Cátedra Ingeniería de Software Unidad 3, Carrera Ingeniería de Ejecución Informática, Pontificia Universidad Católica de Valparaíso. McGraw Hill (2010), *Ingeniería de Software: Un enfoque práctico*.
- [9] Antonio Moya. Gerente Área Desarrollo de Solem Corp. y profesor de Informática, Apuntes catedra (2010). McGraw Hill (2010), *Ingeniería de Software: Un enfoque práctico*.
- [10] Figuras penales relativas a la informática, Ley Chile, Congreso Nacional. (1993) <http://www.leychile.cl/Navegar?idNorma=30590>. Recuperado el 20 de Junio del 2011.
- [11] W3c. Oficina España. Información servicios Web. (2010) <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>. Recuperado el 10 de Mayo del 2011
- [12] Developers Android (2010), Web de Documentación oficial de los desarrolladores Android <http://developer.android.com/>. Recuperado el 10 de Mayo del 2011

Anexos