Facultad de Ingeniera

Universidad del Trabajo del Uruguay

**Centralización de datos de un sistema multimedia para aplicaciones móviles de museos.**

Entregado como requisito para la obtención del título de

Tecnólogo en Informática

Maximiliano Espillar

Rubén Franco

Javier Pérez

Tupac Villalba

Fernando Walpert

Tutor: Carla Forni

Cliente: Ing. Juan Cardelino

2012 - 2013

***Agradecimientos***

Se agradece al docente Gastón Notte, quien nos brindó sugerencias y su opinión con respecto a las decisiones técnicas abordadas, a la docente Sonia Rocha por haber colaborado con la redacción de la presente documentación, a la coordinadora Ana Iruleguy por estar siempre pendiente de los avances del mismo, a la tutora Carla Forni , a los integrantes del grupo de Ingeniería Eléctrica que pusieron a nuestra disposición su proyecto y nos contestaron dudas sobre el mismo y en particular al cliente, Juan Gardelino, por su constante compromiso con el proyecto.

Contenido

[1 Introducción 6](#_Toc357103791)

[1.1 Marco contextual 6](#_Toc357103792)

[1.2 Descripción de las características principales de la aplicación 8](#_Toc357103793)

[1.2.1 Navegación 8](#_Toc357103794)

[1.2.2 Identificación de Obras 11](#_Toc357103795)

[1.2.3 Realidad Aumentada 12](#_Toc357103796)

[1.2.4 Infraestructura informática 14](#_Toc357103797)

[1.3 Motivación. 15](#_Toc357103798)

[1.4 Introducción al problema 16](#_Toc357103799)

[1.5 Alcance del proyecto y Delegación de responsabilidades. 17](#_Toc357103800)

[1.6 Objetivos 18](#_Toc357103801)

[1.7 Métodos de solución 19](#_Toc357103802)

[2. Definición del problema 20](#_Toc357103803)

[3 Lista de Necesidades en el servidor. 22](#_Toc357103804)

[3.1 Gestión, autentificación y autorización de usuarios en SGRM. 22](#_Toc357103805)

[3.2 Navegación y gestión de salas del museo en SGRM. 22](#_Toc357103806)

[3.3 Gestión e identificación de obras del museo en SGRM. 24](#_Toc357103807)

[3.4 Gestión de contenidos multimedia asociados a salas, obras y marcadores. 25](#_Toc357103808)

[4 Selección de tecnologías. 26](#_Toc357103809)

[4.1 Sistema Operativo (10). 26](#_Toc357103810)

[4.2 Herramienta y Lenguaje de desarrollo. 26](#_Toc357103811)

[4.3 Base de datos. 27](#_Toc357103812)

[4.4 Servidor HTTP 28](#_Toc357103813)

[4.5 Transferencia de archivos 28](#_Toc357103814)

[4.6 Sistema de control de versiones 28](#_Toc357103815)

[4.7 Diagramación 28](#_Toc357103816)

[5 Análisis 29](#_Toc357103817)

[5.1 Esquema de Casos de Usos 29](#_Toc357103818)

[5.2 Actores y Metas. 30](#_Toc357103819)

[5.2.1 Empleado del museo 30](#_Toc357103820)

[5.2.2 Administrador 30](#_Toc357103821)

[5.2.3 Visitante 30](#_Toc357103822)

[5.3 Requerimientos Funcionales. 31](#_Toc357103823)

[5.3.1 Identificación de la sala en que se encuentra el visitante. 31](#_Toc357103824)

[5.3.2 Identificación de una obra. 31](#_Toc357103825)

[5.3.3 Obtener y Reproducir Contenido de una obra. 31](#_Toc357103826)

[5.3.4 Inicio de sesión de los usuarios en el sistema. 31](#_Toc357103827)

[5.3.5 Ingreso de un nuevo usuario al sistema 32](#_Toc357103828)

[5.3.6 Modificar datos de un usuario 32](#_Toc357103829)

[5.3.7 Listar usuarios 32](#_Toc357103830)

[5.3.8 Eliminar usuario 32](#_Toc357103831)

[5.3.9 Configuración de parámetros 32](#_Toc357103832)

[5.3.10 Cierre de Sesión 32](#_Toc357103833)

[5.3.11 Ingresar una nueva obra al sistema. 32](#_Toc357103834)

[5.3.12 Modificar datos de una obra 33](#_Toc357103835)

[5.3.13 Listar obras 33](#_Toc357103836)

[5.3.14 Eliminar una obra 33](#_Toc357103837)

[5.3.15 Agregar nueva sala 33](#_Toc357103838)

[5.3.16 Modificar datos de una sala 33](#_Toc357103839)

[5.3.17 Listar las salas del sistema 33](#_Toc357103840)

[5.3.18 Eliminar salas del sistema 33](#_Toc357103841)

[6 Solución del Problema 34](#_Toc357103842)

[6.1 Organización de los archivos en el servidor 34](#_Toc357103843)

[6.2 Descriptores 35](#_Toc357103844)

[6.3 Transferencia de archivos 37](#_Toc357103845)

[6.4 Organización de funciones y métodos públicos. 39](#_Toc357103846)

[6.5 Esquema de la base de datos 41](#_Toc357103847)

[6.5.1 Lógica del problema 41](#_Toc357103848)

[6.5.2 Normalización de la base datos 42](#_Toc357103849)

[6.5.3 Diagrama MER 42](#_Toc357103850)

[6.6 Pagina de configuración del servidor 44](#_Toc357103851)

[7 Creación de una aplicación en Android 46](#_Toc357103852)

[7.1 Requerimientos 48](#_Toc357103853)

[7.1.1 Obtener información de una sala a partir de la lectura de un código QR. 48](#_Toc357103854)

[7.1.2 Procesamiento de imágenes 48](#_Toc357103855)

[7.2 Solución del problema 49](#_Toc357103856)

[7.2.1 Obtener información de una sala a partir de la lectura de un código QR. 49](#_Toc357103857)

[7.2.2 Procesamiento de imágenes 51](#_Toc357103858)

[7.3 Librería Codename One (30) 55](#_Toc357103859)

[8 Conclusión 56](#_Toc357103860)

[8.1 Introducción 56](#_Toc357103861)

[8.2 Conclusiones específicas 57](#_Toc357103862)

[8.2.1 Desarrollo de la solución en el servidor 57](#_Toc357103863)

[8.2.2 Pautas para el desarrollo de una aplicación en Android 58](#_Toc357103864)

[8.3 Conclusiones generales 59](#_Toc357103865)

[8.4 Dificultades del proceso. 60](#_Toc357103866)

[8.5 Comunicación entre los grupos de desarrollo. 61](#_Toc357103867)

[8.6 Comunicación con el cliente. 62](#_Toc357103868)

[8.7 Trabajo a Futuro 63](#_Toc357103869)

[8.7.1 Sobre el servidor 63](#_Toc357103870)

[8.7.2 Sobre la aplicación en Android 63](#_Toc357103871)

[Glosario 64](#_Toc357103872)

[Bibliografía 70](#_Toc357103873)

# 1 Introducción

## 1.1 Marco contextual

El presente proyecto es realizado para la obtención del título de Tecnólogo en informática de la Facultad de Ingeniería en convenio con la universidad del Trabajo del Uruguay.

El proyecto consiste en el desarrollo de un sistema que permita a los visitantes de los museos acceder mediante dispositivos móviles a información sobre los objetos expuestos y disponer de un medio tecnológico destinado a enriquecer la experiencia de visitar un museo, a través de un recorrido interactivo y contenidos multimedia.

Esta aplicación busca estimular la visita en los museos, favoreciendo la interacción entre los visitantes y el objeto cultural de una forma atractiva a la vez que didáctica, brindando una entretenida experiencia educacional, donde el visitante recibirá información del punto que le interesa mientras lo está contemplando, organizando su visita con total libertad y decidiendo cual, durante cuánto tiempo y donde desea ver o escuchar la explicación seleccionada por sí mismo.



Figura - Museos han sustituido las clásicas audiologías, por dispositivos móviles táctiles que permiten al visitante acceder a la información de una manera más fácil. (1)

Desde hace algunos años museos de distintas partes del mundo han comenzado a considerar dispositivos móviles y otras tantas tecnologías que en cierta forma complementan el papel de un guía turístico, permitiendo a los visitantes acceder a más información sobre sus obras de arte como del edificio en sí mismo.

Un ejemplo de esto es la cárcel de alcatraz, ahora transformada en museo que ofrece a los turistas unos audífonos los cuales mediante un sistema de navegación por satélite (GPS), permite identificar la ubicación del visitante y brindar información a medida que el visitante va recorriendo las diferentes secciones del museo.

Otros museos han sustituido las clásicas audiologías, por dispositivos móviles táctiles, tal es el caso del famoso museo Louvre, en París, cuyos visitantes pueden hacer uso de un dispositivo que sirve como guía en tercera dimensión , una alternativa muy interesante para brindar un valor agregado al usuario.

Estos son tan sólo algunos de los ejemplos de diferentes alternativas que han surgido. Sin embargo, esta es un área muy reciente y en la que todavía queda un camino muy largo por recorrer.

El presente proyecto es un complemento del proyecto basado en una aplicación para dispositivos móviles, más específicamente para iPads, que desarrolló como tesis final en el segundo semestre del año 2012 un grupo de estudiantes de la carrera Ingeniería Eléctrica (2).

Por solicitud del Ingeniero Juan Cardelino, quien tendrá el rol de cliente de este proyecto, se pretende continuar con la idea principal planteada, a la cual se le agregara la centralización de los datos en un servidor, la gestión de esos datos y el desarrollo de una aplicación móvil similar a la existente, que cumpla con las mismas funcionalidades que la ya desarrollada para iPads pero implementada para la plataforma Android, buscando así que los museos puedan optar por diferentes alternativas.

Dado que el área de estudio de la ingeniería eléctrica no tiene como objetivo profundizar en aplicaciones de software, pero si en algoritmos de procesamiento de datos, el proyecto desarrollado por los compañeros de ingeniería hizo hincapié en el estudio e implementación de algoritmos de realidad aumentada y procesamiento de imágenes.

Debido a que somos diez los alumnos que tenemos la posibilidad de obtener el título en este período y de que el tamaño del proyecto es significativo, se nos dividió en dos grupos de cinco integrantes, cada uno con responsabilidades previamente definidas.

## 1.2 Descripción de las características principales de la aplicación

### 1.2.1 Navegación

Es importante determinar la ubicación del visitante dentro del museo, para así brindar información contextual acerca de la sección del museo donde se encuentra.

Esta información podría ser una breve descripción de la sala en la que se encuentra, indicaciones para ir de un lugar a otro, contenidos multimedia o información acerca de las obras contenidas en ella.

Desde el punto de vista del sistema, también es importante porque permite optimizar las consultas a la base de datos para que manejen un menor volumen de datos al momento de identificar las obras, debido a que sabe en qué sala se encuentra el usuario; lo cual implica mayor velocidad de respuesta y una reducción de recursos en el servidor.

Se investigaron las siguientes alternativas para poder determinar la ubicación del visitante.

#### 1.2.1.1 Sistema de Posicionamiento Global (GPS) (3)

Una de las alternativas para poder detectar la ubicación del visitante es el uso de GPS, tecnología de la cual disponen muchos de los últimos modelos de dispositivos móviles.

GPS es un [sistema global de navegación por satélite](http://es.wikipedia.org/wiki/Sistema_global_de_navegaci%C3%B3n_por_sat%C3%A9lite) que permite determinar en todo el mundo la posición de un objeto, en este caso de un dispositivo móvil, con una precisión hasta de centímetros.

Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante "[triangulación](http://es.wikipedia.org/wiki/Triangulaci%C3%B3n)" (método de [trilateración](http://es.wikipedia.org/wiki/Trilateraci%C3%B3n) inversa), la cual se basa en determinar la distancia de cada satélite respecto al punto de medición.

Esta alternativa fue descartada debido a que el dispositivo debe recibir las señales del satélite para funcionar y en interiores, puede que el dispositivo no reciba las señales o no las reciba correctamente.

#### 1.2.1.2 Sistema de Posicionamiento en Interiores (4)

Existen funcionalidades desarrolladas por empresas como Google, Nokia y Microsoft entre otras, que basadas en diferentes tecnologías como por ejemplo la intensidad de la señal de distinto puntos Wi-FI permiten detectar la posición de un dispositivo móvil dentro de interiores de edificios, de una forma similar al GPS, pero en lugares donde el GPS no funciona.

Estas funcionalidades no formaran parte del sistema a desarrollar, ya que por lo requisitos de infraestructura, software y licencias con las empresas que las proveen, son soluciones que deben de ser estudiadas para cada museo en particular, y uno de los objetivos de este proyecto es el desarrollo de una solución genérica.

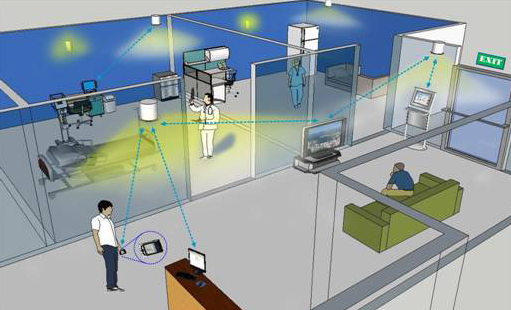


Figura – Sistema de posicionamiento de interiores, como los distintos dispositivos se comunican con los puntos Wi-Fi distribuidos dentro del museo. (4)

#### 1.2.1.3 Procesamiento de imágenes QR.

Otra de las alternativas para poder identificar la sección del museo donde se encuentre el visitante es el uso de los códigos QR; para eso el visitante tomaría una foto con su aplicación móvil del código bidimensional y luego de ir a buscar la información al servidor la aplicación brindará al usuario una lista de opciones y contenidos a visualizar, que previamente fueron asociadas a dicha sección del museo por los empleados del mismo.

El grupo de Montevideo estuvo investigando librerías que a partir de una imagen puedan en primera instancia identificar si se trata de un QR para luego brindar información que permita identificar la sección o sala del museo.

Esta alternativa también requiere que se les brinde a los museos una aplicación generadora de códigos QR, y será responsabilidad del museo, que cada vez que se ingresen nuevas secciones del museo al sistema se genere e impriman en papel los códigos QR, que luego permitan identificar las nuevas salas.

Un código QR (5) es un módulo para almacenar información en un [código de barras](http://es.wikipedia.org/wiki/C%C3%B3digo_de_barras) bidimensional creado por la compañía [japonesa](http://es.wikipedia.org/wiki/Jap%C3%B3n) Denso Wave en 1994 y se caracteriza por los tres cuadrados que se encuentran en las esquinas que permiten detectar la posición del código al lector y por qué permite que su contenido se lea a alta velocidad.

La inclusión de [software](http://es.wikipedia.org/wiki/Software) que lee códigos QR en [teléfonos móviles](http://es.wikipedia.org/wiki/Tel%C3%A9fonos_m%C3%B3viles), ha permitido nuevos usos que se manifiestan en comodidades para el consumidor, como el dejar de tener que introducir datos de forma manual en los teléfonos.



Figura – Ejemplo de un código QR siendo escaneado.

### 1.2.2 Identificación de Obras

#### 1.2.2.1 Procesamiento de imágenes

El grupo de Ingeniería realizo una investigación acerca de algoritmos de detección de características de imágenes con el objetivo de poder identificar desde una base datos una obra del museo solo con su fotografía, sin necesidad de que se indique ninguna otra información.

El proceso de detección de imagen se efectúa cuando el visitante toma una fotografía y consiste en la comparación de la fotografía obtenida con fotografías de obras que ya se encuentren en la base de datos.

Previo a realizarse la comparación entre las imágenes, ellas son procesadas y pasan por diferentes etapas con la finalidad de simplificar la imagen e identificar los bordes y esquinas (Figura 3 y 4).

La comparación entre las imágenes posee cierta similitud a un algoritmo de detección de huellas digitales, donde se busca la fotografía más aproximada, ya que existen ciertos parámetros que varían como por ejemplo lo son el ángulo en que fue tomada la imagen y las sombras existentes.

|  |  |
| --- | --- |
| C:\Users\Martincho\Desktop\innova2.png  Figura - Imagen de entrada | Figura - Imagen de salida |

### 1.2.3 Realidad Aumentada

La realidad aumentada combina en tiempo real elementos reales y virtuales obtenidos por una cámara. Y consiste sobreimprimir información virtual, que pueden ser videos, imágenes o animaciones en 3d a la información real ya existente.

La imágenes que se muestran a continuación corresponden a una aplicación de realidad aumentada desarrollada por la compañía Toyota para sus exposiciones de automóviles, observando el vehículo solamente a través del dispositivo se hace difícil distinguir cuales elementos son reales y cuales forman parte de la escena.





Figura - Realidad aumentada para un auto. Desarrollada por la compañía Toyota para sus exposiciones de automóviles (6)

#### 1.2.3.1 Etapas de la realidad aumentada

En el siguiente diagrama puede apreciarse el proceso en el cual se logra la realidad aumentada

#### 

Figura - Etapas de la realidad aumentadas.

El primer bloque corresponde a la cámara que captura una imagen. En el segundo bloque dicha imagen es procesada con el objetivo de detectar en estas características. Estas características pueden ser segmentos, esquinas u otros. El tercer bloque corresponde a la estimación de pose, busca estimar en qué posición se encuentra la cámara respecto de cierto eje de coordenadas previamente definido y hacia dónde está apunta. En el cuarto bloque con la información anterior, se debe poder representar (*rendering* en inglés) una escena de manera consistente con la pose de la cámara, para así entonces lograr la salida del sistema, que será una imagen con la realidad aumentada incorporada.

### 1.2.4 Infraestructura informática

#### 1.2.4.1 Comunicación con servidor

La arquitectura informática a utilizar es la Cliente-Servidor, que tiene como principio fundamental la agrupación de las aplicaciones de un sistema en dos grupos fundamentales, uno de ellos es el servidor el cual provee los servicios, y el otro grupo son los clientes, los cuales realizan peticiones a los servicios publicados previamente por el servidor.

Una de las ventajas de esta arquitectura es que permite disponer de un único servidor y muchos clientes, de esa manera se pueden centralizar los datos en un único lugar.

#### 1.2.4.2 Servidor

El servidor es un software y hardware, el cual publica servicios que resuelven las peticiones que le hacen dispositivos clientes, entregándole lo que ha solicitado si es posible o, en algunos casos, indicando que no tiene permisos para obtener lo solicitado, o que lo que ha solicitado no existe.

En el sistema a desarrollar, el servidor brindara servicios que permitirán a los empleados del museo ingresar los datos y recursos del museo a través de una interfaz de usuario instalada en dispositivos que serán los clientes del servidor, también son clientes los dispositivos móviles de los visitantes los cuales consumirán otro grupo de servicios del servidor.

#### 1.2.4.3 Interfaz de Usuario

La interfaz de usuario es el medio con que el usuario puede comunicarse con un sistema y comprende todos los puntos de contacto entre el [usuario](http://es.wikipedia.org/wiki/Usuario) y el sistema.

Poseen interfaces de usuario las aplicaciones que serán instaladas en los dispositivos clientes del sistema, que son aquellos que harán uso los empleados del museo y también los dispositivos móviles que utilizaran los visitantes.

Es importante que la interfaz de usuario sea lo más fácil e intuitiva posible para el usuario y disponga de un diseño que le resulte cómodo al usuario.

## 1.3 Motivación.

El grupo en su totalidad comparte que la principal motivación para este proyecto final es que la realización y aprobación del mismo nos permitirá concluir y obtener el título de la carrera Tecnólogo en Informática.

El Proyecto laboral constituirá una experiencia de aplicación de los conocimientos que adquirimos a lo largo de la carrera, a la solución de un problema real. También cumplirá la función de relacionarnos con el mercado laboral en el que tendremos que desempeñarnos, a través de una experiencia de integración a un trabajo real.

Por otra parte, el proyecto planteado por el cliente nos resultó muy atractivo ya que, además de estar relacionado con dispositivos móviles de última generación, como lo son los tabletas, y tecnologías muy modernas, como los Servicios Web (*Web Services* en inglés) en el servidor, es un complemento a un proyecto multimedia sumamente innovador realizado por los compañeros de Montevideo.

También es importante destacar que para este proyecto tendremos que mantener una comunicación fluida con el otro grupo de nuestra carrera y con los compañeros de Ingeniería Eléctrica ya mencionados. Esto nos permitirá obtener nuevas experiencias en el trabajo en equipo, especialmente teniendo en cuenta que la integración de varios grupos en diferentes módulos, pero con un fin en común.

## 1.4 Introducción al problema

La tesis final realizada por los compañeros de ingeniería eléctrica de Montevideo, tenía como objetivo principal establecer las bases para el desarrollo de una aplicación para dispositivos móviles para museos, evaluando la capacidad de procesamiento de dichos dispositivos para aplicaciones de procesamiento de imágenes y estudiando el desempeño de diferentes algoritmos relacionados a la realidad aumentada.

Debido a las limitaciones de tiempo que una tesis final implica y de las dimensiones de un proyecto de estas características, la aplicación para museos fue enfocada desde el área de lo que la ingeniería eléctrica respecta; quedando muchos puntos a resolver antes de implantar la aplicación en un museo.

Continuando y pretendiendo finalizar el trabajo comenzado, surge una lista de necesidades las cuales nos fueron comunicadas por el cliente desde las primeras instancias de este proyecto.

Sin entrar en detalle acerca de las funcionalidades del sistema, podemos afirmar que dichas necesidades se pueden clasificar en cuatro categorías:

1. Centralización de los datos en una base de datos en el servidor, y optimización de las comunicaciones entre el cliente y el servidor.
2. Sistema que permita a los empleados del museo la gestión de los datos con los que los que los visitantes puedan luego interactuar. Este sistema puede a su vez ser dividido en dos partes, una es la interfaz de usuario que podrá ser instalada en varias terminales y la otra radica en el servidor, el cual es único y se encarga de la persistencia de dichos datos en una base de datos.

Como aclaración del lector, cuando hablamos de gestión de datos, ahora y en adelante nos referimos a lo que en informática conocemos como CRUD (que son las siglas en ingles de “Crear, Obtener, Actualizar y Borrar”), también se conoce con el término ABM (Altas, Bajas y Modificaciones).

1. Correcciones en la aplicación móvil para iPads desarrollada por el grupo de Ingeniería Eléctrica, para adaptarla a las nuevas funcionalidades y al cambio en la comunicación con el servidor.
2. Desarrollo de una nueva aplicación para dispositivos móviles que posean el sistema operativo Android y que permita leer códigos qr, procesar imágenes de forma similar a la aplicación para iPads pero con una herramienta diferente y él envió de los datos al servidor.

## 1.5 Alcance del proyecto y Delegación de responsabilidades.

Desde un comienzo del proyecto se nos delegaron responsabilidades, las cuales consistían en un mismo grupo de funcionalidades, pero teniendo en cuenta la arquitectura cliente servidor del proyecto, la división de responsabilidades estaba claramente definida.

**Grupo 1.**

* Interfaz de usuario del sistema de gestión de recursos del museo.

**Grupo 2 (nuestro grupo)**

* Interfaz de comunicaciones del sistema de gestión de recursos del museo y centralización de datos en el servidor.

También hubo otros requerimientos planteados por el cliente, que teniendo en cuenta las limitaciones de tiempo con la que contamos para el desarrollo de nuestra tesis, en un principio no fueron delegadas a ninguno de los dos grupos y su implementación dependía del tiempo que nos llevara la finalización del sistema de gestión de recursos del museo.

Luego de transcurrido un periodo de tiempo de comenzada la tesis y viendo que las necesidades delegadas en primera instancia se podían concluir en el tiempo estimado, fue cuando se acordó delegar los otros requerimientos planteados de la siguiente forma.

**Grupo 1.**

* Correcciones en la aplicación para IOS (iPads).

**Grupo 2 (nuestro grupo)**

* Investigación de los problemas a resolver para implementar una aplicación para Android, que cumpla con funcionalidades similares de la que posee la aplicación desarrollada para IOS.

Por lo tanto este proyecto se limitará a la implementación de los servicios que se alojarán en el servidor, mientras que la interfaz de usuario del sistema de gestión de recursos del museo, será desarrollada por el otro equipo de la carrera Tecnólogo en Informática con el  cual mantendremos una constante cooperación para el análisis de los requerimientos y a fin de sincronizar las comunicaciones entre el servidor y la interfaz de usuario destinada a los empleados del museo.

## 1.6 Objetivos

El objetivo principal de este proyecto es cumplir con las necesidades del cliente, realizando un análisis exhaustivo de los requerimientos que se presenten durante todo el ciclo de vida del proyecto. Esto permitirá participar de manera activa en las solicitudes del cliente, sugiriendo ideas alternativas y mejoras a los requerimientos presentados originalmente.

Asimismo, mediante una investigación temprana de las tecnologías a trabajar, se podrán establecer las limitaciones, ventajas y riesgos que presenten esas herramientas, y cómo éstas pueden incidir en la calidad del software a desarrollar.

Es importante mantener una perspectiva global, no sólo centrándonos en lo que nuestro grupo debe implementar, sino también comprendiendo todo el proyecto como una unidad, debido a que todos los módulos están interrelacionados.

En conclusión, nuestro objetivo consiste en la entrega en tiempo y forma de un producto usable, seguro, multiplataforma, estable, debidamente documentado y de buena calidad, que cumpla con las necesidades del cliente.

## 1.7 Métodos de solución

El primer paso en el desarrollo de este proyecto consiste en detallar y profundizar las necesidades planteadas por el cliente durante las reuniones mantenidas, sin enfocarnos tanto en la solución global del proyecto, sino que nos enfocaremos más en las necesidades que estén comprendidas dentro de las responsabilidades que fueron delegadas a nuestro grupo y que figuran en la sección 1.5 de este documento.

Debido a que nuestro trabajo en este proyecto consiste en aplicar una solución informática al problema planteado, el segundo paso consiste en investigar los protocolos a utilizar para las comunicaciones con el servidor, y justificar la selección de ellos.

Como tercer paso se investigará los lenguajes de programación y tecnologías a utilizar en el servidor que más se adecuen a la solución del problema en cuestión.

El cuarto paso consiste en el análisis del sistema de gestión de recursos del museo desde la perspectiva del servidor, donde se podrá apreciar diagrama de casos de uso y de la base de datos. No se incluirá en esta documentación diagramas de clases o de colaboración, ya que la capa del dominio del sistema se encontrará en la interfaz de usuario, responsabilidad que no fue delegada a nuestro grupo.

Todos los datos a los cuales podrá acceder el visitante desde su dispositivo móvil, así sean descripciones de obras, salas o autores; contenidos multimedia y animaciones en 3D mediante el proceso de realidad aumentada, tendrán que ser previamente gestionados por los usuarios del Sistema de Gestión de Recursos del

Museo, en adelante SGRM.

El quinto paso consiste en la implementación de las funcionalidades en el servidor que a través del entorno y de las herramientas seleccionadas responda eficientemente las diferentes peticiones que llegaran los dispositivos móviles y a la interfaz del SGRM.

El sexto y último pasó consistirá en el análisis y desarrollo de una aplicación para dispositivos móviles con sistema operativo Android que deberá de comunicarse con las operaciones públicas del servidor y contar con algunas de las funcionalidades con las que cuenta el sistema para iPad del grupo de ingeniería de Montevideo, como lo son el procesamiento de imágenes y la lectura e identificación de salas del museo a través del escaneo de códigos Qr.

# 2. Definición del problema

Una vez conocido el contexto y las generalidades del proyecto, es pertinente especificar una lista de las necesidades planteadas por el cliente.

Esta lista fue actualizada en diferentes ocasiones a partir de las charlas con el cliente y de las investigaciones llevadas a cabo durante la realización del proyecto. Adjunto a este documento, se encuentra el Acta de Reuniones, en él se puede ver la evolución del proyecto, donde se detalla en orden cronológico como fueron surgiendo y variando las necesidades solicitadas por el cliente.

Debido a que el proyecto a realizar no es un proyecto creado desde cero sino que partimos de una base de investigación e implementación realizada por el grupo de Ingeniería Eléctrica, que puso a nuestra disposición las fuentes de la aplicación móvil, del servidor y la documentación de su tesis, por lo tanto es importante especificar y dejar claramente definido cuál es nuestro punto de partida.

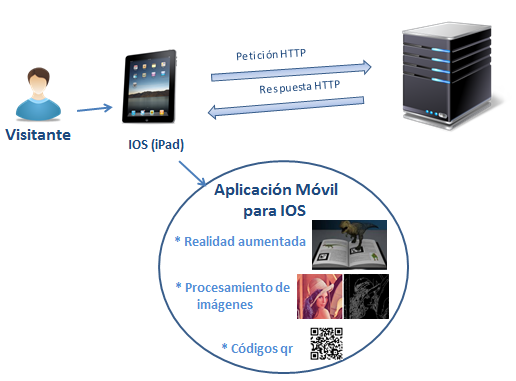


Figura - Arquitectura del sistema desarrollado por los estudiantes de Ingeniería.

En la Figura 8, se puede apreciar la arquitectura del proyecto realizado por los compañeros de Ingeniería, que consistía en una aplicación para iPads desarrollado en el lenguaje Objective-C para sistemas operativos IOS. También se pueden apreciar las funcionalidades más importantes de la aplicación móvil y la comunicación entre el dispositivo y el servidor el cual es mediante el protocolo HTTP (7) tanto para la solicitud de datos, como para él envió o recepción de archivos.

La aplicación del servidor realizada por el grupo de Montevideo, la cual fue realizada en PHP, fue destinada solo para pruebas básicas de recepción y envió de solicitudes HTTP y crear una iniciativa para que otro grupo continuara con su trabajo.

La siguiente imagen muestra un diagrama de la arquitectura del proyecto en su totalidad, la base del proyecto realizado por los estudiantes de Ingeniera (imagen anterior) más los proyectos que llevarán a cabo ambos grupos de la carrera Tecnólogo en Informática.

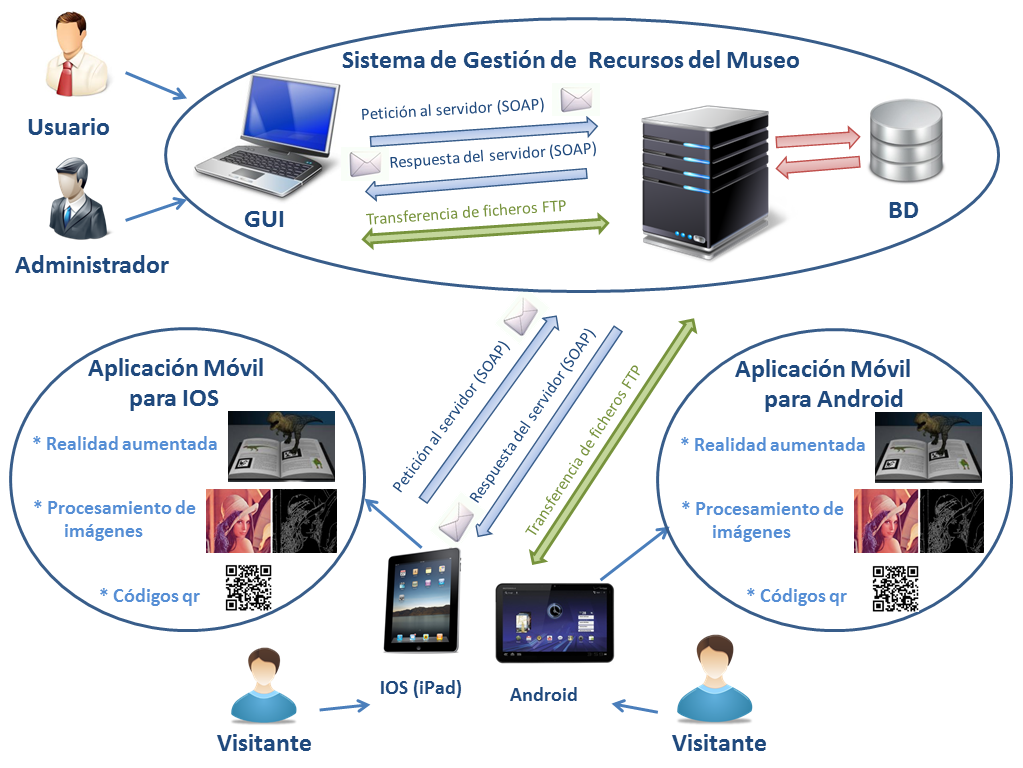


Figura – Representación del sistema en iOS y Android trabajando en conjunto.

# 3 Lista de Necesidades en el servidor.

La lista de necesidades del sistema de gestión de recursos del museo (SGRM), se pueden dividir en los siguientes puntos:

* + Gestión, autentificación y autorización de usuarios.
  + Navegación y gestión de salas del museo.
  + Gestión e identificación de obras del museo en SGRM.
  + Gestión de contenidos multimedia asociados a salas, obras y marcadores.

3.1 Gestión, autentificación y autorización de usuarios en SGRM.

Todos los usuarios podrán iniciar sesión en el sistema mediante el nombre de usuario y contraseña. Una vez ingresado esos datos, el sistema verificara que los datos sean válidos y permitirá o denegará el inicio de sesión del usuario al sistema.

SGRM contará en primera instancia solo con un usuario administrador, el cual poseerá permisos para gestionar a otros usuarios comunes o administradores. Las funcionalidades de SGRM de gestionar usuarios estarán restringidas a usuarios comunes, permitiendo a solo los usuarios administradores hacer uso de ellas.

3.2 Navegación y gestión de salas del museo en SGRM.

La importancia de este punto es brindar información contextual acerca de la ubicación del visitante dentro del museo.

Esta información podría ser una breve descripción de la sala en la que se encuentra, indicaciones para ir de un lugar a otro, o contenidos multimedia que brinden información acerca de la sección del museo en la que se encuentra y las obras contenidas en ella.

Desde el punto de vista del sistema, también es importante porque permite optimizar las consultas a la base de datos para que manejen un menor volumen de datos al momento de identificar las obras, ya que sabe en qué sala se encuentra el usuario; lo cual implica mayor velocidad de respuesta y una reducción de recursos en el servidor.

La identificación de la sección del museo donde se encuentre el visitante se realizaría mediante códigos QR; para eso el visitante tomaría una foto con su aplicación móvil del código bidimensional y luego de ir a buscar la información al servidor la aplicación brindará al usuario una lista de opciones y contenidos a visualizar, que previamente fueron asociadas a dicha sección del museo por los empleados del mismo.

El grupo de Montevideo estuvo investigando librerías que a partir de una imagen puedan en primera instancia identificar si se trata de un QR para luego brindar información que permita identificar la sección o sala del museo.

Esa información, así como un ejemplo del uso de una librería con ese fin fue puesta a disposición de los dos grupos, en una de las charlas que mantuvimos.

Se brindará a los museos una aplicación generadora de códigos QR, pero será responsabilidad de los museos y de sus empleados, que cada vez que se ingresen nuevas secciones del museo al sistema se genere e impriman en papel los códigos QR, que luego permitan identificar las nuevas salas.

3.3 Gestión e identificación de obras del museo en SGRM.

Los empleados del museo, usuarios de SGRM, podrán agregar, actualizar y eliminar obras del sistema.

Para agregar una nueva obra, primero deberán de seleccionar la sala a la cual pertenece. También es imprescindible que el empleado provea el nombre de la obra, una descripción de la misma y una imagen que será enviada al servidor mediante un cliente FTP (8).

Una vez finalizado el envío de la imagen al servidor, se invocará mediante un servicio web a una función del servidor pasando por parámetro información acerca de la ubicación de la imagen.

Esa función ejecutará un algoritmo de detección de características de imágenes llamado SIFT (9), el cual fue utilizado por el grupo de Ingeniería de Montevideo y es una de las piezas principales de todo este proyecto ya que permite al sistema identificar las obras del museo con el contenido de la imagen.

SIFT es un algoritmo desarrollado en C, por lo cual deberá de transformarla en una librería a la cual se pueda acceder desde PHP que es el lenguaje empleado en el servidor.

Cuando se crea o modifica una obra, este algoritmo guarda un conjunto de datos (descriptores, que funcionan de manera análoga a una huella digital) de cada obra en archivos de texto, para luego, cuando el visitante saca una fotografía de la obra, aplicar el mismo proceso en búsqueda de la obra que más se aproxime a la imagen brindada por el empleado.

3.4 Gestión de contenidos multimedia asociados a salas, obras y marcadores.

Tanto las obras como las salas dispondrán de diversos tipos de contenidos que podrán ser asociados a ellos desde SGRM.

Estos contenidos podrán ser archivos de audio, imágenes, videos o animaciones en 3D provenientes de la interfaz de usuario de SGRM que serán enviados por ftp al servidor y dependiendo del tipo de contenido almacenados en diferentes carpetas.

Estos contenidos, todos ellos, dispondrán de un título y una breve descripción, datos que serán mostrados en primera instancia al visitante cuando fotografié una obra o el código QR de una sala.

Por lo tanto, el usuario dispondrá de un menú con el nombre de cada uno de los contenidos de la obra o sala que fotografió, y podrá acceder a visualizar cada uno de ellos de a uno a la vez.

Las animaciones en 3D serán mostradas al visitante mediante un proceso de realidad aumentada desarrollada en el lenguaje C por el grupo de Ingeniería de Montevideo y es otra de las características del proyecto más importante e innovadoras.

La realidad aumentada consiste en mostrar un video o una animación sobre una sección de la pantalla (punto de referencia), sobreponiendo la animación a la secuencia de imágenes captados en tiempo real por la cámara del dispositivo móvil.

Una vez comenzada la reproducción de la animación en el dispositivo móvil, de forma dinámica la animación se reubicará y girará en la pantalla siguiendo el punto inicial de referencia.

Tanto la cámara del dispositivo móvil, como un componente llamado giroscopio generan eventos imprescindibles para que la realidad aumentada sea posible.

# 4 Selección de tecnologías.

## 4.1 Sistema Operativo (10).

Teniendo en cuenta la solicitud del cliente de que utilicemos herramientas multiplataforma y software libre, seleccionamos el sistema operativo Ubuntu Linux por la facilidad de uso.

## 4.2 Herramienta y Lenguaje de desarrollo.

Las herramientas a seleccionar para implementar desde el lado del servidor debían ser software libre y multiplataforma.

Debido a las características del proyecto, la herramienta también debía poseer soporte para Servicios Web (11), tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

El intercambio de datos mediante Servicios Web posee la ventaja de aportar interoperabilidad entre aplicaciones de software, independientemente de sus propiedades o de los sistemas operativos sobre los que se instalen, además de brindar independencia entre la aplicación que usa el servicio web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no afectarían al otro.

Luego de analizar diferentes opciones y de descartar aquellas que no cumplían con los prerrequisitos, las alternativas J2EE (12) y PHP (13) fueron las preseleccionadas para realizar una investigación más en profundidad.

Una tercera alternativa, Ruby On Rails (14), tomaría fuerza, luego de consultar con el profesor de las materias .Net y J2EE, Nicolás Zuasti, el cual nos comentó que poseía experiencia en ambas tecnologías, y que en su opinión Ruby On Rails era la opción más práctica y robusta para el tipo de proyecto a implementar.

Fue así que basamos nuestra investigación en estas tres tecnologías apuntando principalmente a Ruby, que era la que menos conocíamos.

En esa etapa de investigación, buscamos información sobre las ventajas y desventajas de estas tecnologías en foros y documentación online, para así poder estar al tanto de la experiencia que otros desarrolladores hubieran tenido y compararlas para decidir cuál sería la más conveniente a utilizar en nuestro proyecto.

La información encontrada fue bastante relativa y a veces contradictoria. Los gustos personales y la experiencia en una u otra tecnología influían mucho en las opiniones de los desarrolladores, así que, luego de algunas pruebas básicas que realizamos, elaboramos una tabla donde el puntaje máximo era 10, para poder determinar más claramente los pros y contras del uso de estas herramientas en nuestro proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
|  | PHP | JAVA | Ruby On Rails |
| Soporte para Servicios Web | 10 | 10 | 10 |
| Mantenibilidad | 9 | 9 | 10 |
| Experiencia previa y conocimientos de la herramienta por parte del grupo | 10 | 7 | 0 |
| Orientación a Objetos | 9 | 10 | 10 |
| Ventajas del uso del lenguaje y de las herramientas de reconocimiento de errores | 9 | 10 | 10 |
| Velocidad de instalación , sencillez y agilidad de las herramientas a utilizar | 10 | 9 | 9 |
| Ventajas en aplicaciones empresariales de gran porte | 9 | 10 | 10 |
|  |  |  |  |
| **Total** | **66** | **65** | **59** |

Al finalizar el análisis, Ruby On Rails nos pareció una tecnología muy moderna, potente y práctica (11). Posiblemente si hubiéramos tenido experiencia previa en esta tecnología hubiésemos optado por la misma, pero dado de que los tiempos de investigación eran bastante limitados, ya que teníamos que ir a la par del otro grupo cuyo desarrollo en la interfaz de usuario dependía mucho de lo desarrollado en el servidor, y de que los riesgos de implementar en una tecnología en la cual no teníamos experiencia previa eran considerables, decidimos descartarla para este proyecto. De todas formas Ruby On Rails nos pareció muy interesante de aprender más adelante, para así tenerla en cuenta en futuros proyectos.

PHP fue entonces la herramienta seleccionada. Los puntos más fuertes que determinaron la elección de la misma fueron la experiencia con la que los miembros del equipo contamos en el desarrollo con esta tecnología y la sencillez de su uso. Creemos que estos puntos inciden en la velocidad de desarrollo, y que si bien JAVA es la opción que generalmente se utiliza en ámbitos empresariales y desarrollos de gran porte, no existen desventajas considerables para la utilización del mismo en este proyecto.

## 4.3 Base de datos.

Al igual que para el resto de las tecnologías, unos de los prerrequisitos era que el motor de base de datos debía poseer licencia libre y además tener un buen rendimiento, funcionar en diferentes plataformas y ser fácil de configurar e instalar.

De los motores de base de datos que investigamos, sólo dos (MySQL y Postgress) cumplían con esas especificaciones. La decisión unánime del equipo fue el uso de MySQL (15), dado que ya habíamos trabajado anteriormente con esta herramienta y que incluye WampServer y Xampp, herramientas que facilitan la instalación y configuración tanto de Mysql como de PHP y Apache.

## 4.4 Servidor HTTP

Una vez seleccionada la herramienta de desarrollo PHP, el motor de base datos MySQL y el servidor Apache fueron rápidamente seleccionados, teniendo en cuenta las ventajas de lo que comúnmente se llama un servidor LAMP (16) (Linux + Apache + MySQL + PHP).

4.5 Transferencia de archivos**.**

Dadas las desventajas que encontramos en la transferencia de archivos mediante Servicios Web (ver sección 7.3), optamos instalar un servidor FTP (17), y mediante ese protocolo realizar las transferencias de archivos.

Por la popularidad, las prestaciones y la amigable interfaz de usuario, Filezilla Server fue la opción seleccionada.

## 4.6 Sistema de control de versiones

La herramienta para el control de versiones seleccionada fue GIT (18), para poder organizarnos y controlar versiones y documentación de cambios.

## 4.7 Diagramación

Para el diseño de diagramas de esta documentación se utilizó Star UML (19).

# 5 Análisis

5.1 Esquema de Casos de Usos

El diagrama de casos de uso representa la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). (20)

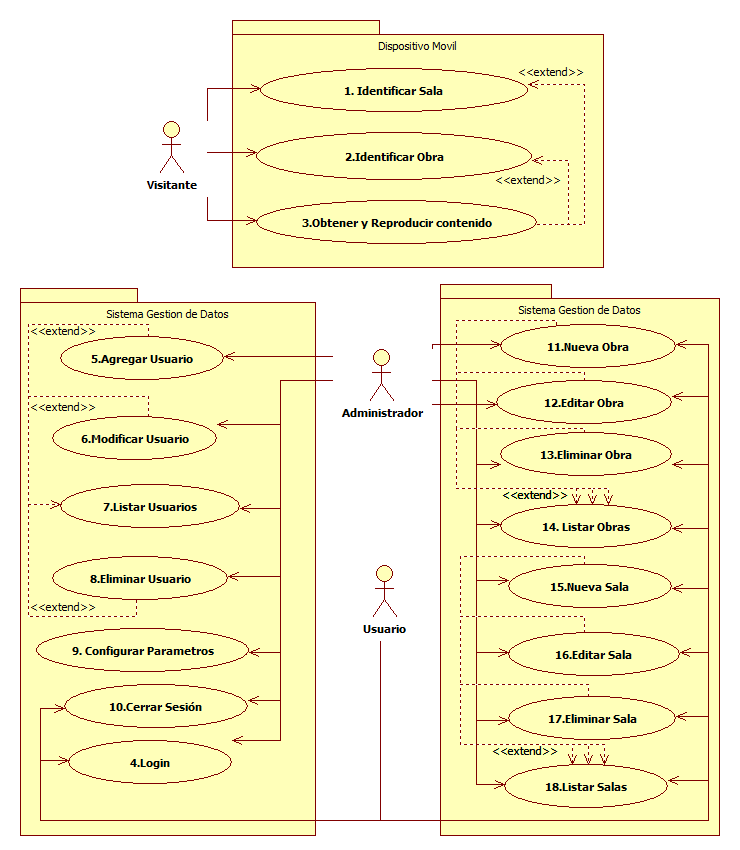
****

Figura - Esquema de casos de uso

5.2 Actores y Metas.

### 5.2.1 Empleado del museo

Realiza la gestión de las salas y obras del museo.

### 5.2.2 Administrador

Es un empleado más y puede realizar todas las acciones de un empleado y además gestiona los usuarios y los parámetros del sistema.

### 5.2.3 Visitante

Interactúa con los datos del sistema, ingresados previamente por los usuarios.

## 5.3 Requerimientos Funcionales.

### 5.3.1 Identificación de la sala en que se encuentra el visitante.

Este caso de uso permite identificar la sala del museo en la que se encuentra el visitante. Esta persona saca una foto al código QR de la misma y el sistema busca en qué sala se encuentra, retornando una identificación de la sala correspondiente. En caso de que no se encuentre una sala se retornará un error.

### 5.3.2 Identificación de una obra.

Luego de que el visitante fotografía una obra, la aplicación instalada en el dispositivo móvil envía la imagen al servidor donde mediante un algoritmo de detección de características de imágenes llamado SIFT se generara el descriptor correspondiente. Es un algoritmo basado únicamente en el contenido de la imagen para identificar la obra. Se verá en profundidad en el apartado 1.

Una vez generado el descriptor, el sistema busca en la base de datos la obra que más se aproxime a la que el visitante fotografió, teniendo en cuenta un rango de aproximación. Si la búsqueda encuentra una obra dentro del rango, se retorna la identificación de la obra que más se aproxime; en caso de que no se encuentre una obra en ese rango se retorna un mensaje de error.

### 5.3.3 Obtener y Reproducir Contenido de una obra.

Luego de identificarse la obra, se muestra al usuario una breve descripción de la misma y la opción de ver diferentes contenidos uno a la vez.

El usuario le comunica al sistema que quiere obtener un contenido en particular de la obra actual; para ello se envía el identificador de la obra al servidor. Los contenidos pueden ser videos, audio, texto y animaciones en 3D.

### 5.3.4 Inicio de sesión de los usuarios en el sistema.

Un usuario administrador o un empleado podrán iniciar sesión en el sistema mediante el nombre de usuario y contraseña. El sistema verifica que los datos sean válidos y permite el inicio de sesión o retorna un mensaje de error informando que los datos no son correctos.

### 5.3.5 Ingreso de un nuevo usuario al sistema

El administrador podrá ingresar nuevos usuarios al sistema completando datos obligatorios y otros opcionales de los mismos.

### 5.3.6 Modificar datos de un usuario

El administrador podrá seleccionar un usuario de la lista de usuarios, previamente ingresado en el sistema, y modificar los datos del mismo.

### 5.3.7 Listar usuarios

El sistema lista todos los usuarios ingresados en el sistema; luego el administrador selecciona uno y el sistema muestra todos los datos correspondientes al mismo.

### 5.3.8 Eliminar usuario

El usuario administrador podrá eliminar un usuario del sistema, pudiendo ser este un empleado u otro administrador.

### 5.3.9 Configuración de parámetros

Permite al administrador configurar la calidad del video y la velocidad de la transferencia de datos.

### 5.3.10 Cierre de Sesión

Permite a los usuarios cerrar sesión en el sistema.

### 5.3.11 Ingresar una nueva obra al sistema.

Posteriormente al inicio de sesión, el usuario podrá agregar una nueva obra ingresando el nombre de la obra y su imagen.

Los datos se envían al servidor, donde se genera un descriptor sobre la imagen ingresada y se verifica que no exista una obra registrada con los mismos datos.

Una vez finalizada la verificación, el usuario podrá elegir, de una lista de contenidos, qué tipos de contenidos (audios, textos, videos o animaciones en 3D) desea agregar a la obra.

### 5.3.12 Modificar datos de una obra

Permite a un usuario con sesión previamente iniciada seleccionar una obra de la lista de obras previamente ingresadas al sistema, y  modificar los datos de la misma incluyendo sus contenidos (audio, texto y video).

### 5.3.13 Listar obras

Permite a un usuario que previamente haya iniciado sesión, listar todas las obras ingresadas en el sistema. Luego, el usuario selecciona la obra deseada y el sistema muestra todos los datos de dicha obra.

### 5.3.14 Eliminar una obra

Permite a un usuario que previamente haya iniciado sesión, seleccionar una obra de la lista de obras previamente ingresadas al sistema y eliminarla de la base de datos.

### 5.3.15 Agregar nueva sala

Permite a un usuario agregar una nueva sala al sistema ingresando los datos de la misma. Al confirmar el ingreso, los datos se envían al servidor y se crea una nueva sala en el sistema.

### 5.3.16 Modificar datos de una sala

Permite a un usuario seleccionar una sala existente de una lista de salas, y modificar los datos de la misma. Al confirmar la modificación, los datos se envían al servidor modificando la sala en el sistema.

### 5.3.17 Listar las salas del sistema

El sistema lista todas las salas ingresadas; si el usuario selecciona una sala, el sistema muestra todos los datos correspondientes a la misma.

### 5.3.18 Eliminar salas del sistema

Permite a un usuario con sesión previamente iniciada seleccionar una sala y eliminarla del sistema. Para eso la sala no debe tener obras. Al confirmar la acción se envían los datos al servidor donde se procede a eliminarla.

# 6 Solución del Problema

## 6.1 Organización de los archivos en el servidor

Con la intención de organizar los archivos se optó por distribuirlos en una estructura de directorios que sea fácil de comprender y de modificar.

Esta estructura consta de un directorio raíz y dentro de estos tres directorios (“obras”, “salas” y “zonas”) y que a su vez contienen los recursos organizados en subdirectorios con nombre la identificación única e irrepetible de la sala, zona u obra en cuestión.

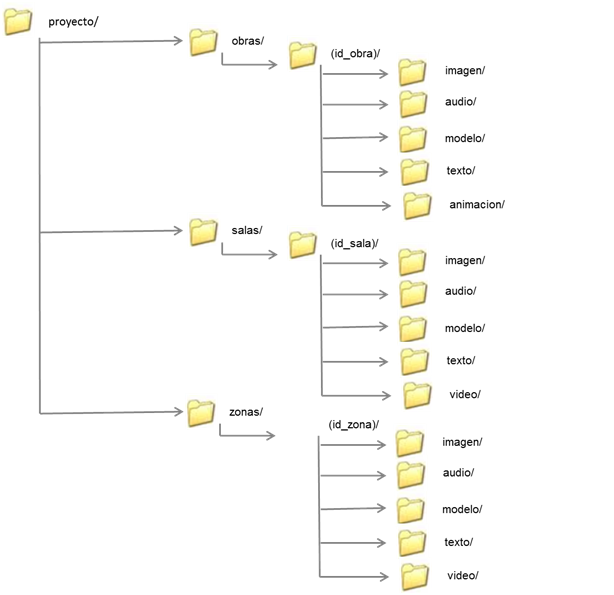
****

Figura - Jerarquía de archivos dentro del servidor.

## 6.2 Descriptores

La implementación de la funcionalidad descripta en la sección 1.2.2.1 se puede dividir en dos partes.

1. Cuando se da de alta una obra, los descriptores son generados y almacenados en el servidor mediante la función SIFT de la librería VLFEAT, la cual está desarrollada en C y transforma la imagen en una lista de números llamados descriptores.
2. La identificación de una obra se realiza mediante la comparación de los descriptores generados de la fotografía de la obra a identificar enviada al servidor con los descriptores ya almacenados en el servidor que fueron generados en el momento en que se dieron de altas las obras en el sistema.

Cuando comenzamos con el proyecto, el algoritmo utilizado por los compañeros de ingeniería guardaba en archivos de texto el conjunto de descriptores de una obra, en un directorio fijo.

La identificación de la obra, se realizaba comparando la aproximación de los descriptores generados por el algoritmo de la imagen enviada al servidor, con los descriptores almacenados en archivos de textos.

Dado que la búsqueda era estática la primera corrección realizada a este algoritmo, consistió en extender la comparación a todos los archivos de un directorio y no solo a una lista de archivos previamente definidos, logrando así una búsqueda dinámica.

Al solucionar el problema anterior, nos encontramos con que el algoritmo de detección de imágenes demoraba un tiempo considerable en la comparación de los datos almacenados en los archivos de textos.

Teniendo en cuenta que estas pruebas en un principio las realizamos con pocos archivos y que un museo puede tener una cantidad significante de obras, este es un problema muy importante de solucionar antes de implantar el sistema en un museo.

Luego de llevar a cabo una investigación para encontrar las causas de este problema, se determino que la demora era causada por el algoritmo de comparación y no por la forma en que se accedía a los datos.

Dado que la solución de este problema no formaba parte del alcance del proyecto, y que las dimensiones del desarrollo de esta solución son comparables a las dimensiones de un proyecto similar al presente, este problema formara parte de los aspectos a mejorar en el futuro.

Una de las últimas correcciones realizadas consistió en cambiar la forma en la que se almacenaban los descriptores, guardándolos ahora en una tabla de la base de datos en lugar de archivos de texto, en busca de mejorar la accesibilidad y seguridad en los datos.

## 6.3 Transferencia de archivos

En un principio se intentó enviar y recibir los archivos mediante los servicios web publicados en el servidor, pero cuando enviábamos archivos multimedia de tamaño considerable, nos encontramos con que los archivos no llegaban a destino, debido a que los servicios web no están diseñados principalmente como mecanismo de transferencia de archivos. Se realizó una investigación para encontrar una alternativa la cual resulto en la determinación del uso de un servidor ftp para la transferencia de archivos, tecnología que no solo permitía él envió de archivos sin importar su tamaño, sino que además facilita el control del acceso a los directorios, al poder asignar a los usuarios permisos a determinados directorios.

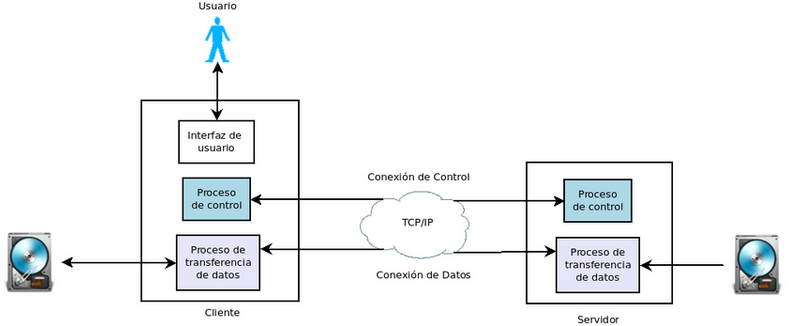


Figura - Esquema de transferencia de archivos.

Para cumplir con la organización de directorios en el servidor planteada en el capítulo anterior, es necesario que previo al envío de un archivo, se haya consumido el servicio web encargado de crear el directorio donde se alojara el archivo.

Se crearon tres usuarios en el servidor ftp, cada uno de ellos con permisos para un directorio en particular.

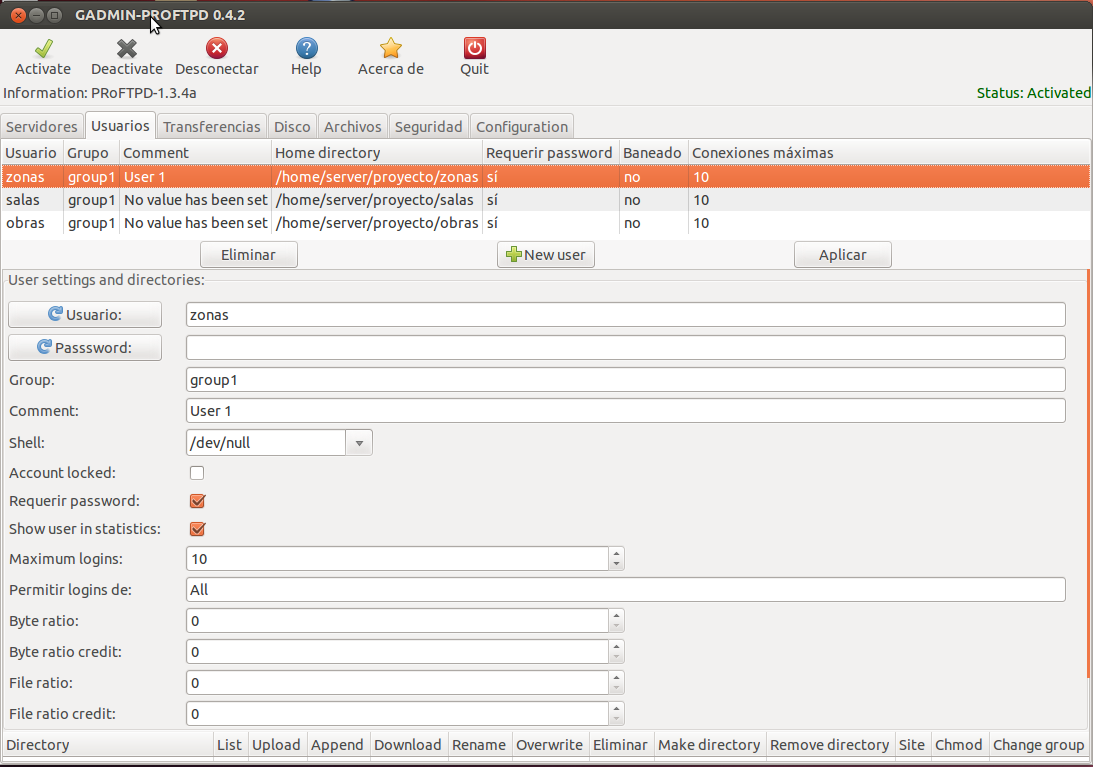


Figura - Captura de pantalla del servidor ftp visto desde Gadmin Pro FTP.

* Usuario con acceso al directorio salas
  + Usuario : salas
  + Contraseña: 12345678
* Usuario con acceso al directorio obras
  + Usuario : obras
  + Contraseña: 12345678
* Usuario con acceso al directorio zonas
  + Usuario : zonas
  + Contraseña: 12345678

## 6.4 Organización de funciones y métodos públicos.

Si bien son muy conocidas por el grupo las ventajas de la orientación a objetos, en nuestra solución no otorgaba grandes beneficios ya que cada uno de los servicios web publicados realizan una operación en concreto en la base de datos y que cuando se necesita traer un conjunto de datos de una o varias tablas, se retorna una lista en formato similar a JSON (21).

También es importante destacar que en caso de que hubiésemos optado por utilizar orientación a objetos en el servidor solo tendrían una utilidad local, ya que las instancias de los objetos no podrían ser retornadas a las aplicaciones clientes que consumen los servicios web.

De todas formas se decidió agrupar las funciones en los siguientes archivos:

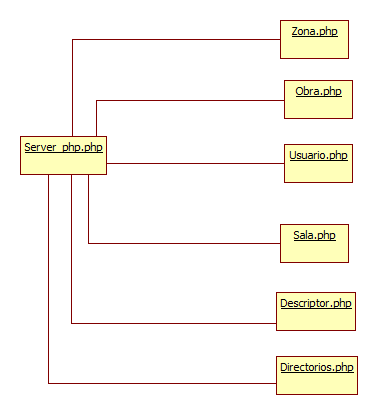


Figura - Organización de las funciones dentro del servidor web.

**Server\_php.php:**

Es el archivo que publica todos los servicios web del servidor, establece la conexión a la base de datos e invoca las funciones existentes en el resto de los archivos.

**Usuario.php:**

Contiene las funcionalidades relacionadas con el alta, baja,  modificación,  y control de ingreso de los usuarios del sistema.

**Sala.php:**

Contiene las funcionalidades relacionadas con el alta, baja, y  modificación de las salas del sistema; así también como las funcionalidades para obtener datos de las salas y sus contenidos.

**Obra.php:**

Contiene las funcionalidades relacionadas con el alta, baja, y  modificación de las obras del sistema; así también como las funcionalidades para obtener datos de las obras y sus contenidos.

**Zona.php:**

Contiene las funcionalidades relacionadas con el alta, baja, y  modificación de las zonas de interés de las obras del sistema; así también como las funcionalidades para obtener datos de las zonas y sus contenidos.

**Descriptor.php:**

Contiene dos funciones que utilizan librerías en C, la primera de ellas es responsable de generar los descriptores a partir de la imagen de una obra y almacenarlos en la base de datos.

La segunda función, se utiliza para identificar una obra a partir de su imagen, comparando los descriptores de la obra enviada por parámetro con los descriptores de las obras ya existentes en el sistema.

**Directorios.php:**

Posee dos funciones auxiliares para eliminar archivos y directorios.

## 6.5 Esquema de la base de datos

Son muchas las ventajas que presentan los gestores de base datos para almacenar información, las más importantes de ellas son la seguridad,  la accesibilidad  y la integración de los datos.

El sistema desarrollado por los compañeros de ingeniería, no contaba con una base de datos, ya que los únicos datos que se persistían en ese entonces, eran los descriptores y se los guardaba en archivos de texto.

Ante la necesidad de ampliar y mejorar ese sistema, también surgió la necesidad de mejorar la forma de persistir los datos, optamos por utilizar un SGBD (sistema gestor de base de datos)  para almacenar la información, y así disponer de esas ventajas comentadas anteriormente.

### 6.5.1 Lógica del problema

La estructura de la base de datos se realizó teniendo en cuenta la información de las salas, obras y zonas y sus respectivos contenidos y la información de los usuarios del sistema.

De las salas se conoce su identificación, nombre, descripción  e imagen del código QR que identifica la sala. En cada sala se exponen varias obras, de las cuales se conoce su identificador, nombre, descripción, autor y la imagen. Cada obra tiene descriptores y zonas de interés. De las zonas, interesa registrar su identificador, el nombre de la obra a la que pertenece, largo, ancho, y sus coordenadas X e Y, que indican su posición y dimensión dentro de la obra. De los descriptores  se conoce su identificador, el identificador de la obra a la cual pertenece y sus valores.

Cada sala, obra y zona tienen contenidos los cuales se conoce su identificador, el identificador a la sala, obra o zona a la cual pertenece, los contenidos pueden ser de tipo audio, video, texto, modelo 3D e imagen; además a las obras se le agrega el tipo de contenido animación.

Por otro lado están los empleados del museo de los cuales se registra su identificador, nombre, apellido, cédula, email, nick y contraseña. Algunos

### 6.5.2 Normalización de la base datos

La estructura de la base de datos se realizó en base a la lógica del problema y aplicándole a ella las técnicas de normalización de base de datos, para así crear una base de datos eficiente y simple de entender.

La aplicación de una serie de reglas a las relaciones obtenidas tras el paso del [modelo entidad-relación](http://es.wikipedia.org/wiki/Modelo_E-R) al [modelo relacional](http://es.wikipedia.org/wiki/Modelo_relacional), nos permitió obtener un esquema de base de datos que protege la integridad de los datos, no posee redundancia y evita problemas en la actualización de los datos.

### 6.5.3 Diagrama MER

En la figura 13 se encuentra el modelo MER realizado, a partir de la realidad del problema planteado anteriormente.

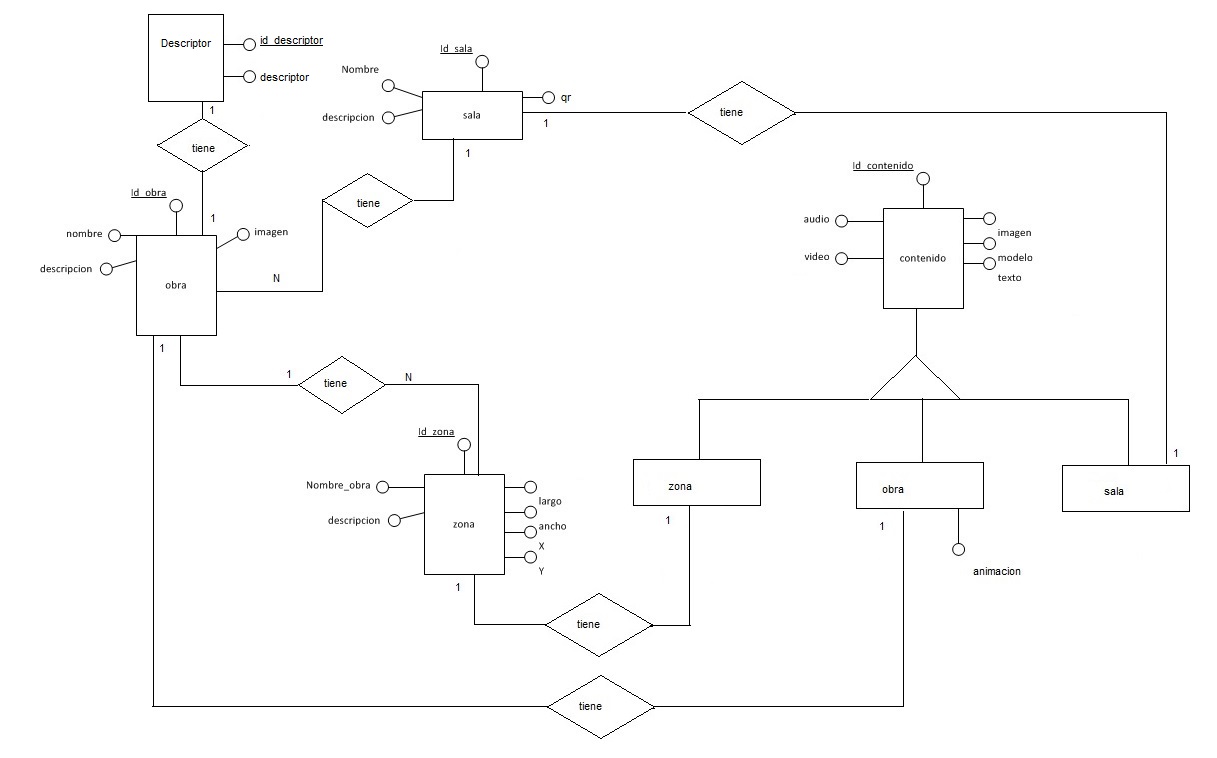


Figura - Diagrama MER.

|  |
| --- |
| Al modelo de la figura 13 podemos representarlo con las siguientes tablas:  Sala ( id\_sala, nombre, descripción, qr);  Obra ( id\_obra, nombre, descripción, imagen);  Zona ( id\_zona, nombre\_obra, descripción, largo, ancho, X, Y);  Contenido\_sala ( id\_contenido, id\_sala, audio, video, imagen, modelo, texto);    Contenido\_obra ( id\_contenido, id\_obra, audio, video, imagen, modelo, texto, animación);  Contenido\_zona ( id\_contenido, id\_zona, audio, video, imagen, modelo, texto);  Descriptor ( id\_descriptor, descriptor, id\_obra);  Utilizamos la tabla Usuario para guardar los datos de los usuarios, clasificar el tipo de usuario, y autentificar el usuario en el sistema.  Usuario( id\_usuario, nick, nombre, apellido, pass, cedula, email, tipoUs); |

## 6.6 Pagina de configuración del servidor

Con la intención de permitir la portabilidad de los datos del servidor y de facilitar la implantación y portabilidad del sistema, se realizo una página web encargada de configurar los datos más importantes del servidor.

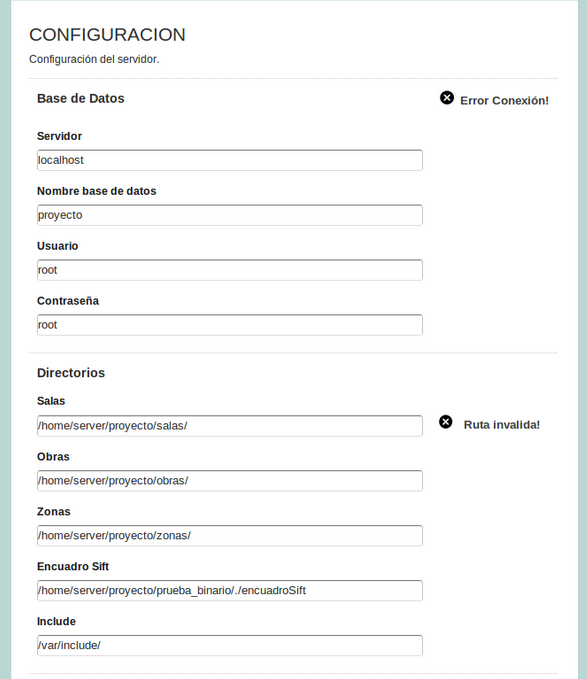


Figura - Configuración del servidor, parte 1

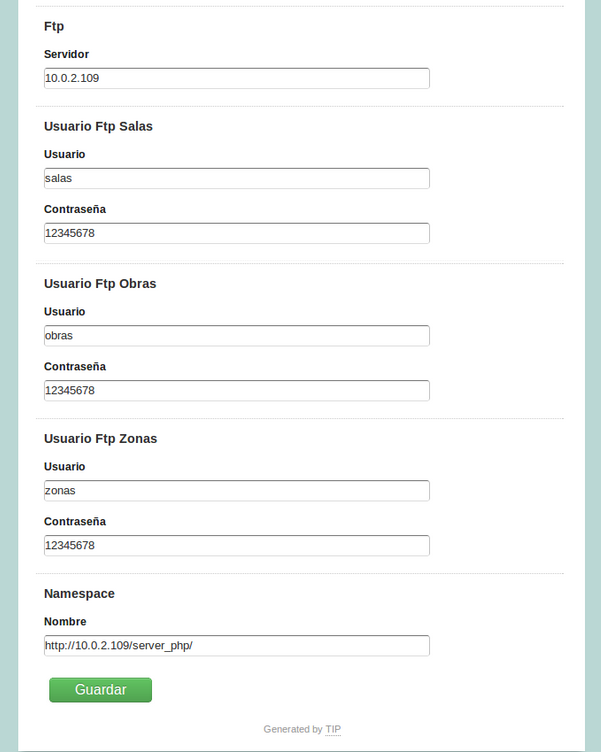


Figura - Configuración del servidor, parte 2.

## 

Figura - Configuración del servidor, parte 3.

# 7 Creación de una aplicación en Android

Android es un sistema operativo desarrollado en un primer momento para dispositivos móviles pero que ha ido evolucionando a lo largo del tiempo y en la actualidad es posible utilizarlo en diversos dispositivos que nada tienen que ver con un terminal móvil.

Así lo podemos ver en set-top-box para televisiones, en las propias televisiones, en radios, en vehículos, en lavadoras, en robots industriales, microondas, etc.

Es un Sistema Operativo para terminales móviles, se encarga de  recibir las órdenes de los usuarios, de gestionarlas y transmitirlas a los diferentes componentes (hardware) del terminal en donde esté instalado.

Las aplicaciones para Android se escriben y desarrollan en Java aunque con unas APIs propias por lo que las aplicaciones escritas en Java para PC y demás plataformas ya existentes no son compatibles con este sistema.

Si bien el iPad de Apple desde su aparición ha dominado el mercado, en los últimos meses han surgido una gran cantidad de tabletas con prestaciones similares y en ocasiones mejores.

La aplicación desarrollada por los compañeros de Montevideo fue desarrollada en el lenguaje  Objective-C solo para iPad de Apple que son las únicas que poseen el sistema operativo IOS.

Una de las solicitudes del cliente,  fue la de realizar una aplicación para Android que tenga  un comportamiento similar a la aplicación desarrollada por los compañeros de Montevideo.

Con excepción del iPad, el resto de las tabletas con mejores prestaciones existentes actualmente poseen el sistema operativo Android. En la siguiente tabla se puede apreciar una comparativa de las prestaciones del iPad y de otras tabletas con Android.

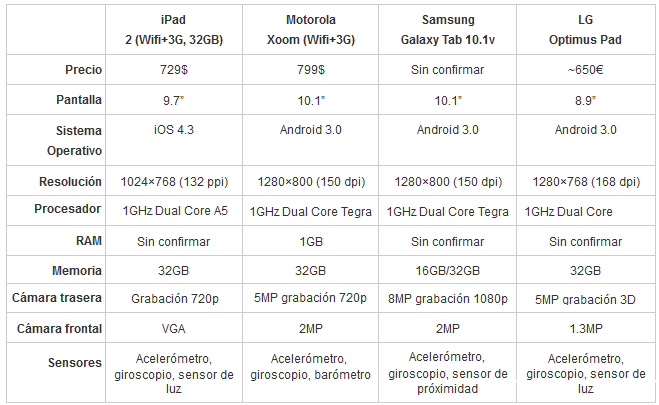


Figura - Tabla comparativa de dispositivos móviles.

## 7.1 Requerimientos

### 7.1.1 Obtener información de una sala a partir de la lectura de un código QR.

Se debe realizar una aplicación que a partir de un código QR se obtenga la información de la sala en la cual se encuentra el visitante del museo  y que utilice el sistema operativo Android.

### 7.1.2 Procesamiento de imágenes

Se debe realizar una aplicación donde se procese una imagen desde Android utilizando algún algoritmo escrito en C.

El algoritmo utilizado para la aplicación fue LSD (22): a Line Segment Detector, este fue el elegido por el cliente.

El algoritmo LSD incluye fórmulas matemáticas, geometría y lógica entre otras, que procesan una imagen, la cual puede ser utilizada de distintas maneras. En resumen lo que hace el algoritmo es detectar e identificar las los rasgos más significativos de la imagen y dibujar una nueva imagen dejando en ella solo “segmento de recta”.

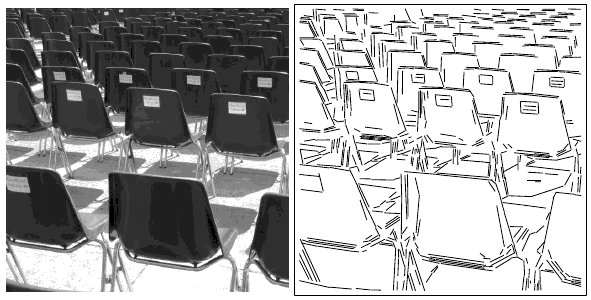


Figura – A la izquierda se muestra la imagen de entrada, a la derecha imagen de salida del algoritmo LSD.

Esto nos resulta de mucha utilidad al momento de identificar un cuadro dentro del museo, es mucho más fácil “escanear” un grupo de líneas que una imagen en bruto, con brillo y sombras que pueden interferir en el proceso

## 7.2 Solución del problema

### 7.2.1 Obtener información de una sala a partir de la lectura de un código QR.

El modo que optamos para resolver el problema fue fotografiar un código QR para obtener la identificación de la sala y enviarla al servidor para que este nos brinde toda la información de la sala.

Para solucionar este problema se dividió en dos partes, una de estas fue el manejo de códigos QR,  y otra la comunicación con el servidor.

Para resolver las parte del problema se utilizaron dos librerías, ksoap2-android para enviar y recibir información del el servidor y ZXing para escanear  y decodificar códigos QR.

KSOAP2-android (23) es una librería de cliente SOAP eficiente y ligera para la plataforma Android, con la cual se puede hacer peticiones de forma muy sencilla.

Esta librería forma parte de KSOAP2 que en su mayoría se utiliza en la plataforma de Android, pero también funciona en otras plataformas que utilizan bibliotecas de Java.

[ZXing](http://code.google.com/p/zxing/) (24) es un proyecto de código abierto implementado en Java, que centra su capacidad en el uso de la cámara de los dispositivos para escanear y decodificar códigos de barras y QR directamente en el dispositivo sin necesidad de comunicarse con el servidor.

Esta librería es la más popular en este ramo y nos facilitó mucho integrar la lectura de códigos a través de las llamadas a los compontes de esta.

La aplicación consiste en que por medio de un botón se invoca al lector QR de la librería ZXing el cual detecta y recupera el contenido del código QR en forma texto.



Figura – Primera etapa de la aplicación.

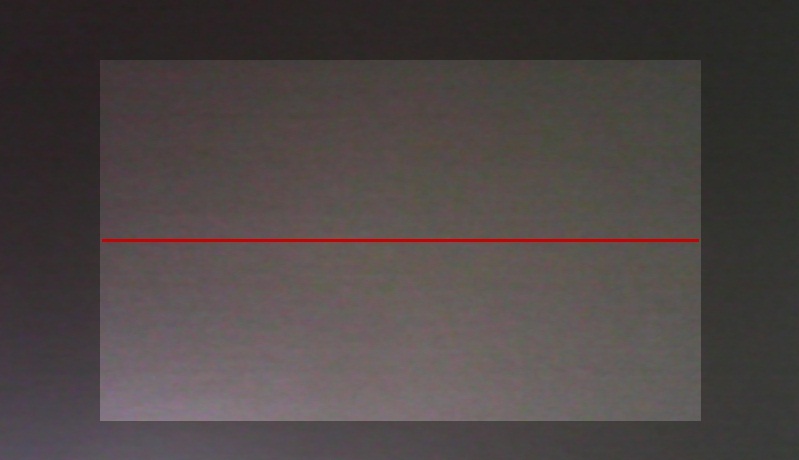


Figura - Lector QR de la librería ZXing.

Este texto es utilizado mediante la librería KSOAP2-android en la petición al servicio web definido en el servidor y este nos retorna la información de la sala correspondiente si el contenido del código QR es correcto, de lo contrario nos devuelve un mensaje de error. Estos valores tanto la información de la sala o el mensaje de error se imprimen en pantalla.

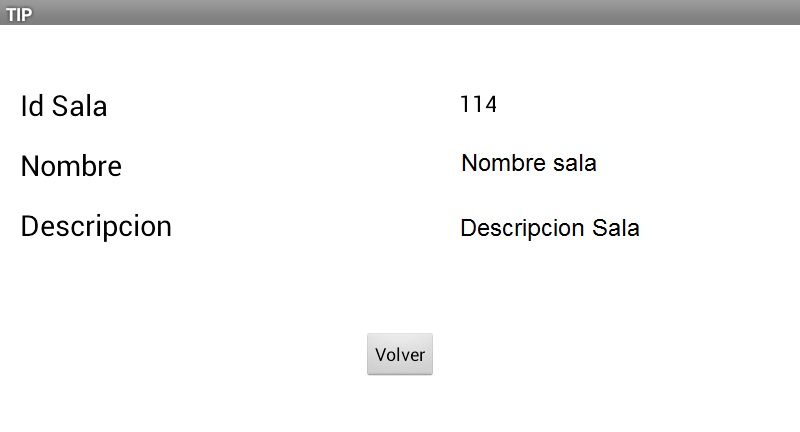


Figura - Información de la sala correspondiente al código QR fotografiado.

Esta aplicación está diseñada para soportar desde la versión de Android 2.2 en adelante.

### 7.2.2 Procesamiento de imágenes

Esta quizás sea la etapa que más dificultades presentó dentro de Android, el algoritmo seleccionado LSD para procesar las imágenes proviene de un grupo de desarrolladores en el cual uno de ellos es uruguayo, y no tuvo ningún inconveniente en ayudarnos ante cualquier duda o problema que se nos presentó, como por ejemplo la implementación de este algoritmo codificado en un lenguaje diferente al que se utiliza en Android y también con los formatos de las imágenes que este genera.

Para cumplir con la tarea decidimos plantearnos ciertos pasos para que realice la aplicación, los cuales serían la obtención de un imagen, y el procesamiento de ésta para utilizarla desde C (25) y el manejo del resultado obtenido desde C.

Para resolver este problema comenzamos separando a este en varias partes, unas de las partes fue como utilizar funciones en C desde Android, como procesar imágenes y obtenerlas también desde Android, cómo funciona y como utilizar el resultado del código de LSD.

Dentro de la investigación para la primera parte planteada de como ejecutar funciones en C desde Android nos encontramos con un conjunto de herramientas llamado NDK que permite integrar código de las aplicaciones escritas en Java que han sido compiladas con el Android SDK (26) y código nativo escrito en C/C++ (27). Para lograrlo se utiliza JNI (28) que es una interfaz que nos permite hacer llamadas a código escrito en C/C++ desde Java mediante un sencillo sistema de nombrar los métodos.

El NDK (29) es utilizado con el fin de reutilizar código, para mejorar el rendimiento de ciertas partes de las aplicaciones y para cargar librerías que no se puede utilizar directamente en Java.

Para obtener imágenes desde Android se utilizó un código que llama a la aplicación de la cámara.

El algoritmo utilizado fue LSD, el cual detecta segmentos de líneas a partir de una imagen, para comprender más sobre esta función nos pusimos en contacto con Rafael Grompone uno de los creadores de esta función y nos ayudó a comprender un poco más sobre esta.

La función LSD funciona con imágenes de un solo canal o también a estas se las llaman imágenes en escala de grises. La escala de grises es la representación de una imagen en la que cada pixel se describe dentro de una serie limitada de valores que representa su luminancia, en una escala que se extiende entre blanco y negro.

Al llamar a LSD como función esta retorna una lista de números que corresponden a las coordenadas de los segmentos y otros datos, en la documentación del algoritmo LSD se encuentra explicado que es cada uno de los 7 números asociados.

Para mostrar el resultado de la función LSD utilizamos las coordenadas de los puntos de inicio y fin de cada segmento de línea encontrado, los cuales con una clase llamada Canvas (30) que representa una superficie donde podemos dibujar y proporciona métodos que nos permite representar líneas entre otras cosas.

La aplicación consiste en que a través de un botón se invoca una aplicación la cual maneja la cámara del dispositivo, como se muestra en la Figura 24.

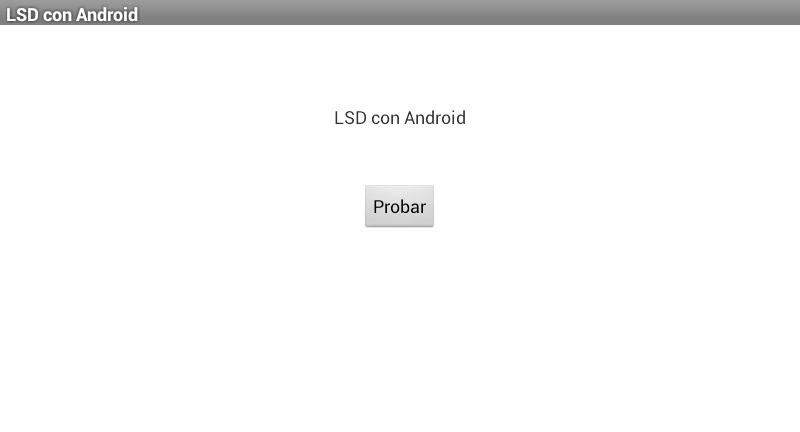


Figura – Probar

Esta aplicación nos permite sacar una foto la cual es enviada a nuestra aplicación para luego ser convertida a escalas de grises mediante una función (Figura 25).



Figura - Imagen tomada por la aplicación.

Esta imagen en escala de grises por medio de NDK y JNI es enviada a la función LSD donde nos retorna una lista de números, de la cual se obtienen las coordenadas de los segmentos. Estas coordenadas y la imagen obtenida desde la cámara son dibujadas a través de la superficie creada con la clase Canvas para ser mostradas al usuario.



Figura - Imagen resultado de la aplicación.

Esta aplicación está diseñada para soportar desde la versión de Android 2.2 en adelante.

## 7.3 Librería Codename One (31)

Codename One es una solución de código abierto (32) que permite crear rápidamente aplicaciones nativas para todos los dispositivos móviles usando Java y opcionalmente un constructor de interfaz gráfica de usuario. Este Framework proporciona acceso completo a la plataforma nativa subyacente sin dejar de lado la portabilidad.

Esta herramienta se utilizó en un principio y resultó ser una manera muy sencilla de crear aplicaciones para dispositivos móviles. A medida que intentábamos cubrir con todas las necesidades de la aplicación Android, nos encontramos con un problema, este framework es relativamente nuevo, aún se encuentra en una etapa temprana de desarrollo, y aunque la persona encargada de esta herramienta respondía casi instantáneamente a nuestras consultas, hay varias operaciones que aún no se pueden hacer con la sencillez que caracteriza a Codename One.

Debido a estas limitaciones, se optó por dejar de utilizarla, es una muy buena herramienta para el desarrollo de aplicaciones móviles, y se encuentra en constante crecimiento, esperemos en un futuro poder utilizarla en su totalidad.

# 8 Conclusión

## 8.1 Introducción

Teniendo en cuenta que nuestro proyecto final está contenido dentro de otro proyecto más grande, el cual engloba también el trabajo del otro grupo de Tecnólogo en Informática y el ya realizado por el grupo de Ingeniería Eléctrica de Montevideo es importante analizar el proyecto desde diferentes perspectivas.

Primero se visualiza la resolución desde una perspectiva local donde se analiza el cumplimiento o no de los objetivos planteados en una primera instancia, realizando una discusión técnica respecto a las elecciones realizadas y también comentaremos las dificultades que nos encontramos en la implementación del mismo.

Luego se analiza el proyecto en su conjunto, desde una perspectiva más global donde se evalúa como se llevó a cabo la comunicación entre los grupos de desarrollo y con el cliente; y como estas comunicaciones influyeron en el proyecto.

Finalmente, se plantea el trabajo que se podría hacer a futuro de manera de mejorar la aplicación final.

## 8.2 Conclusiones específicas

Cabe destacar, que para poder llegar a los resultados obtenidos, la etapa de análisis fue muy importante, ya que además de la investigación acerca de las herramientas a utilizar, tuvimos que dedicar también un tiempo considerable a la investigación de la lógica del problema, y de cómo la habían abordado los compañeros del grupo de Ingeniería Eléctrica.

### 8.2.1 Desarrollo de la solución en el servidor

La primer y principal parte de nuestro proyecto consistió en la implementación de una solución a un conjunto de requerimientos, en el servidor.

Las técnicas de proceso de normalización de [bases de datos](http://es.wikipedia.org/wiki/Base_de_datos) (33) nos permitieron diseñar un esquema de base de datos fácil de entender y que minimiza los problemas de lógica.

Para facilitar y organizar la persistencia de los recursos multimedia en el servidor, se diseñó una estructura de directorios, la cual es creada de forma dinámica a medida que se ingresan nuevas entidades. Luego de que un recurso multimedia es persistido en el servidor, también se guarda en la base de datos una referencia al mismo en la tabla correspondiente, para luego poder acceder a él.

Con respecto a la generación de los descriptores en el servidor, si bien no realizamos directamente modificaciones en el algoritmo que los genera, ni en la forma en que se comparan los descriptores, si cambiamos el sitio donde ellos son persistidos, pasándolos de un archivo de texto a una tabla de la base de datos.

Las ventajas que proporcionan el uso de un servidor ftp para la transferencia de archivos, y la publicación de servicios web en un servidor web, contribuyeron tecnológicamente al desarrollo de una eficiente solución de software al problema planteado.

Finalizando el proyecto surgió además la necesidad de facilitar y centralizar la configuración de los datos más relevantes del servidor, para lo cual se realizó una página web que persiste en un archivo de configuración toda la información brindada por el usuario del servidor.

### 8.2.2 Pautas para el desarrollo de una aplicación en Android

La segunda parte de nuestro proyecto, consistió, en el desarrollo de una funcionalidad en Android que permita la identificación de las salas del museo, a partir de la fotografía de un código Qr y en la investigación de librerías que permitan el procesamiento de imágenes en Android.

Con respecto al desarrollo de la funcionalidad en Android que permite identificar las sala del museo a partir del código Qr, si lo miramos desde un punto de vista técnico, el mayor aporte que deja no es solo la solución de esa funcionalidad en sí, sino que a través de ella abordamos puntos críticos, como lo son la comunicación desde un dispositivo Android con el servidor y las decodificación de códigos qr que sumados a la investigación ya documentada en la sección 8.2.2 sobre el procesamiento de imágenes en Android, nos permite concluir que es totalmente viable el desarrollo de una aplicación en Android con características similares a la desarrollada en Ipad y dejar las pautas dadas para que en un futuro, otro grupo continúe con el trabajo.

## 8.3 Conclusiones generales

Ya analizados los aspectos cualitativos del proyecto, también podemos afirmar que cumplimos con los objetivos establecidos en primera instancia desarrollando exitosamente todos los requerimientos acordados y detallados en el alcance del proyecto.

.

## 8.4 Dificultades del proceso.

Como parte natural de un proceso de desarrollo, nos encontramos con obstáculos a superar en las diferentes etapas del proyecto. Los más importantes son los descritos a continuación.

La selección del lenguaje a utilizar en el servidor, fue un reto importante para nuestro grupo, pues había uno de ellos “Ruby On Rails” que se nos había recomendado por parte de uno de los docentes y a su vez nos resultaba muy atractivo, pero también era el lenguaje menos conocido por parte del grupo de desarrollo y del cual en ese entonces no se contaba con la documentación debida para el uso que le queríamos dar.

Llegado el momento donde no se podía extender más el periodo de investigación de selección del lenguaje, se tomó la decisión de optar por PHP, lenguaje con el cual contábamos con más experiencia lo que implicaba una opción más segura para resolver rápidamente posibles problemas que pudiesen surgir durante el proyecto y así reducir riesgos tecnológicos que puedan comprometer el proyecto.

Dado que teníamos que ir a la par del otro grupo, para que pudiera realizar las pruebas de sus avances en la interfaz de usuario persistiendo los datos en el servidor, en un principio comenzamos el desarrollo en un servidor que contaba con sistema operativo Windows 7 y que entre otras herramientas le habíamos instalado Filezilla Server para las transferencias de archivos mediante FTP, pero siempre con la idea de luego migrar todo a un servidor con sistema operativo Ubuntu como lo había solicitado el cliente.

Cuando migramos el servidor de Windows a Ubuntu, nos encontramos con que el servidor de FTP Filezilla Server no estaba disponible para Linux y que solo la aplicación cliente FTP de Filezilla era multiplaforma. A esta dificultad la solucionamos instalando otro Servidor FTP “[Gadmin Pro FTP”.](http://www.google.com.uy/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CDsQFjAB&url=http%3A%2F%2Fubuntuforums.org%2Fshowthread.php%3Ft%3D1496943&ei=BftHUavWNc-y0QHo8YG4AQ&usg=AFQjCNEUR9A1GAJeJme7taKZFFWPjOtmbg&bvm=bv.43828540,d.dmQ)

Para la implementación del sistema en Android se utilizaron lenguajes y herramientas que no habían sido estudiadas durante la carrera, lo cual requirió de un esfuerzo extra en cuanto a tiempo de análisis e investigación.

## 8.5 Comunicación entre los grupos de desarrollo.

Se destaca que hubo una comunicación muy fluida entre ambos grupos encargados del desarrollo del sistema (el de la interfaz de usuario y el del servidor), la cual concluyó en una eficiente planificación de las etapas a desarrollar y permitió que ambos grupos fuesen implementando al mismo tiempo los mismos casos de uso elaborados en la etapa de análisis, pero desde diferentes perspectivas.

Esta sincronización, tanto en el diseño como en la implementación, favoreció significativamente a ambas partes, ya que permitió ir testeando las funcionalidades de forma unitaria y en conjunto a medida que estas se iban implementando. Además fomentó la participación de los integrantes de ambos grupos para llegar a un acuerdo en la decisión de qué características debían tener los servicios web a publicar en el servidor para que la interfaz de usuario los pueda consumir, para así cumplir con los objetivos detallados en el punto 1-3 de este documento.

Si bien ambos grupos operaron como uno sólo al definir claramente las interdependencias, las características de los Servicios Web permitieron trabajar bajo una arquitectura de software descentralizada donde el desarrollo en la interfaz de usuario, con respecto al desarrollo en el servidor poseía una considerable autonomía, bajo acoplamiento y alta cohesión.

Esta modularidad que establecen los Servicios Web, donde la interdependencia de ambos grupos residió casi únicamente en los servicios web pactados, redujo la complejidad del sistema ya que cada grupo tenía responsabilidades específicas y claramente limitadas.

Un ejemplo de esto es que el grupo de la interfaz de usuario no tenía necesidad de estar al tanto sobre la estructura de la base de datos, las consultas SQL del lado del servidor ni de los algoritmos que allí se utilicen.

Tampoco fue necesario que el grupo del servidor estuviera al tanto de la estructura de clases de la interfaz de usuario, de la navegación de las ventanas y de la forma en que los datos se muestren a los usuarios.

## 8.6 Comunicación con el cliente.

A pesar de que la especialidad del cliente es la Ingeniería eléctrica, posee amplios conocimientos en el desarrollo de software que permitieron que en las reuniones pactadas se emplearan términos técnicos.

Creemos que esos conocimientos por parte del cliente influyeron de manera muy positiva en el proyecto, ya que en los requerimientos planteados también nos daba sugerencias e información técnica de cómo quería que se resolviesen sus solicitudes.

La comunicación fue muy buena. El cliente supo expresar de una manera muy clara sus necesidades y expectativas con respecto a este proyecto, y el grupo pudo comprender e interpretar lo transmitido por el cliente.

Otro factor favorable al buen entendimiento de los requerimientos fue que si bien en los grupos de desarrollo se designaron delegados representantes de cada grupo, que eran encargados de acordar entrevistas con el cliente y la tutora del proyecto, no existió una jerarquía donde un líder del proyecto hablara únicamente con el cliente. Por el contrario todos los integrantes podían participar de las reuniones y emitir sus opiniones y dudas.

El cliente puso a disposición de los grupos su correo electrónico y mensajería online para responder consultas de forma inmediata, favoreciendo así a que no se tuviese que esperar a la próxima reunión para despejar las dudas que pudieran surgir, además de que aquellos integrantes que por razones de disponibilidad horaria no pudieran concurrir a una reunión, pudieran comunicarse con el cliente por este medio.

## 8.7 Trabajo a Futuro

### 8.7.1 Sobre el servidor

Si bien realizamos diferentes pruebas en búsqueda de errores, estas fueron realizadas a un producto en entorno de producción y con las limitaciones que eso conlleva.

Un trabajo a futuro importante es someter el sistema a prueba de estrés, simulando un entorno real, en donde serán muchos los dispositivos clientes que se conectaran simultáneamente al servidor, el cual manejara una cantidad considerable de datos.

### 8.7.2 Sobre la aplicación en Android

Las funcionalidades desarrolladas por nuestro grupo para Android, tenían como objetivó investigar la viabilidad de una futura aplicación que contemple el procesamiento de imágenes y la comunicación con los servicios web del servidor.  
  
Como trabajo a futuro, se podría unificar estas dos funcionalidades y agregarle otras con la finalidad de implementar una aplicación que pueda ser utilizada en los museos, haciendo uso de lo ya desarrollado en el servidor.

# Glosario

**ABM** - Se refiere a los términos de Alta Baja y Modificación de datos. Se dice así cuando un sistema te tiene que dar todas esas posibilidades. En inglés CRUD.

Alcance de un proyecto de software - Es una de las herramientas utilizadas en la administración de proyectos. Puede utilizarse para definir qué formato debe tener o qué aspectos debe incluir, sin embargo la finalidad es la misma: la definición y control de lo que está y no está incluido en el proyecto.

**Algoritmo** - es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad

**Android -** Es un sistema operativo basado en Linux, diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas.

**Apache** - es un servidor web de código abierto, para que implemente el protocolo HTTP. Es usado principalmente para enviar páginas web estáticas y dinámicas en Internet.

**Aplicación móvil** - Es un programa que de descarga y al que se puede acceder directamente desde su teléfono o desde algún otro aparato móvil – como por ejemplo una tableta.

**Array** - Conjunto de elementos ordenados en fila.

**Arquitectura de software descentralizada** – Es aquella que tiene varias o muchas aplicaciones funcionando en varios ordenadores con un flujo importante de comunicaciones. Tiene como objetivo global: obtener prestaciones razonables a un coste bajo

**Base de datos** - conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

**Objetive-C** (34) – Es un lenguaje de programación orientado a objetos.  
 Ejemplo de sintaxis de código Objetive-C:

int main( int argc, const char \*argv[] )

{

NSLog( @"Hola Mundo\n" );

return 0;

}

**Calidad de Software** - Es la aptitud de un producto o servicio para satisfacer las necesidades del usuario. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios.

**Caso de uso** – Es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. Es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

**Ciclo de vida de un proyecto** - Un proyecto se desarrolla mediante actividades que se agrupan en fases. A este conjunto de fases se les llama ciclo de vida, las cuales facilitan el control sobre los tiempos del proyecto y el control sobre el trabajo subcontratado por dicho proyecto.

**Clases** - En la programación, una clase es una construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo. El modelo describe el estado y el comportamiento que todos los objetos de la clase comparten.

**Cliente / Servidor** - La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.

**Código QR** - Es un módulo útil para almacenar información en una matriz de puntos o un código de barras bidimensional. Se caracteriza por los tres cuadrados que se encuentran en las esquinas y que permiten detectar la posición del código al lector.

Figura - Código QR

Figura 20– Código QR

**CRUD** - se refiere a los términos de Create Read Update & Delete, o en español, crear, leer, actualizar y borrar. Se dice así cuando un sistema te tiene que dar todas esas posibilidades.

**Filezilla** - es un cliente FTP que realiza transferencias de archivos entre sistemas conectados a una red.

**Framework** - Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, con base a la cual otro proyecto de *software* puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**FTP** - Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

**Gadmin Pro FTP** – es un cliente FTP que realiza transferencias de archivos entre sistemas conectados a una red.

**Giroscopio** - Es un dispositivo mecánico que sirve para medir, mantener o cambiar la orientación en el espacio de algún aparato o vehículo. Dispositivos móviles traen incluido este sensor para el uso de las aplicaciones.

**GIT** - Es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

**GUI** - La interfaz gráfica de usuario, conocida también como es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

**Ingeniería de software** - Es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software.

**Hardware** -Se refiere a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos. Son cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

**HTTP** - Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web) se lo conoce como agente del usuario. A la información transmitida se la llama. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

**Interfaz de comunicación** - Es la comunicación que establece la CPU del ordenador con cada uno de los dispositivos periféricos (teclado, ratón, teléfono móvil conectado,…) que se unen a ella a través de cables enlazados a sus respectivos puertos de conexión o conexiones inalámbricas con el fin de permitir el intercambio de información.

**Interoperabilidad** - H*abilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada*.

**iOS** - Es un sistema operativo móvil de la empresa Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV.

**J2EE** - Java Platform, Enterprise Edition es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java.

**JAVA** – Es un lenguaje de programación, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web.

**Lenguaje de programación** - Es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

**Librería** - Es un conjunto de subprogramas utilizados para desarrollar software. Las librerías contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos.

**Linux** - es un núcleo libre de sistema operativo basado en Unix. Es uno de los principales ejemplos de software libre, y está desarrollado por colaboradores de todo el mundo.

**Mantenibilidad** - Propiedad de un sistema que representa la cantidad de esfuerzo requerida para conservar su funcionamiento normal o para restituirlo una vez se ha presentado un evento de falla. Se dirá que un sistema es "Altamente mantenible" cuando el esfuerzo asociado a la restitución sea bajo.

**Metodología** - Hace referencia al conjunto de procedimientos racionales utilizados para alcanzar una gama de objetivos que rigen en una investigación científica, una exposición doctrinalo tareas que requieran habilidades, conocimientos o cuidados específicos.

**Módulo** - Es una porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso).

**Multimedia** - Se utiliza para referirse a cualquier objeto o sistema que utiliza múltiples medios de expresión físicos o digitales para presentar o comunicar información. De allí la expresión multimedios. Los medios pueden ser variados, desde texto e imágenes, hasta animación, sonido, video, etc.

**Multiplataforma** - es un atributo conferido a los programas informáticos o los métodos de cálculo y los conceptos que se ejecutan y operan en múltiples plataformas informáticas. Por ejemplo, una aplicación multiplataforma puede ejecutarse en Microsoft Windows, Linux y Mac OS.

**MySQL** - Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar modificar y analizar los datos.

**.NET** - Es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java y a los diversos framework de desarrollo web basados en PHP.

**PHP** - Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

**Persistencia de datos** - Es la representación residual de datos que han sido de alguna manera nominalmente borrados o eliminados.

**PostgreSQL** – Es un sistema de gestión de bases de datos que permite el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar modificar y analizar los datos.

**Programación orientada a objetos** - es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

**Procesamiento digital de imágenes** - Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

**Protocolo de comunicaciones o de red** - Es el conjunto de reglas normalizadas para la representación, señalización, autenticación y detección de errores necesario para enviar información a través de un canal de comunicación.

**RAM** - La memoria de acceso aleatorio se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo. Se denominan «*de acceso aleatorio*» porque se puede leer o escribir en una posición de memoria con un tiempo de espera igual para cualquier posición, no siendo necesario seguir un orden para acceder a la información de la manera más rápida posible.

**Realidad aumentada** - Es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente, es decir, añadir una parte sintética virtual a lo real.

**Riesgo de un proyecto** - es un evento incierto o condición incierta que si ocurre, tiene un efecto positivo o negativo sobre el proyecto.

**ROM** - La memoria de sólo lectura, es un medio de almacenamiento utilizado en ordenadores y dispositivos electrónicos, que permite sólo la lectura de la información y no su escritura, independientemente de la presencia o no de una fuente de energía.

**Ruby On Rails** - Es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby. Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración

**Sensor** - Es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

**Sistema global de navegación por satélite (GPS)** - es una constelación de satélites que transmite rangos de señales utilizados para el posicionamiento y localización en cualquier parte del globo terrestre, ya sea en tierra, mar o aire. Estos permiten determinar las coordenadas geográficas y la altitud de un punto dado como resultado de la recepción de señales provenientes de constelaciones de satélites artificiales de la Tierra para fines de navegación, transporte, geodésicos, hidrográficos, agrícolas, y otras actividades afines.

**Sistema Operativo** - Es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes.

**SOAP** - Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos. Es uno de los protocolos utilizados en los servicios Web.

**Software** - Se conoce como el equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.

**Software Libre** – Es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido el mismo puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, y estudiar el mismo, e incluso modificar el software y distribuirlo modificado.

**Star UML** - Es una herramienta para el modelado de software basado en los estándares

**Ubuntu -**  Es un sistema operativo la comunidad de desarrolladores. Utiliza un núcleo Linux. Ubuntu está orientado al usuario novel y promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia de usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto.

**Servicios Web** (*Web Services* en inglés) -Es una tecnología que utiliza un conjunto de protocolos y estándares que permiten a distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma intercambiar datos entre sí.

# Bibliografía

1. **CAILLARD, Leo.** Léo Caillard y el futuro de los museos. [En línea] http://eldadodelarte.blogspot.com/2011/02/leo-caillard-y-el-futuro-de-los-museos.html.

2. **BRAUN, juan, y otros.** *Tesis de ingeniería eléctrica Facultad de Ingeniería.* Montevideo : Documentación enCuadro, 2012.

3. GEEKNIZER. *GPS & Reliable Indoor Position System IPS.* [En línea] http://geeknizer.com/gps-reliable-indoor-position-system-ips/.

4. ILÍBERI Software. *Blog de ILÍBERI Software & Geografía.* [En línea] http://iliberi.com/blog/category/windows-phone/.

5. **KINDBERG, Tim.** Mobile Codes: Standards and Guidelines A discussion document. [En línea] http://www.mobilecodes.org/StandardsDiscussion.pdf.

6. **PASLAY, David.** Toyota Augmented Reality. [En línea] http://davidpaslay.com/Toyota-Augmented-Reality.

7. *Qué es HTTP.* [En línea] http://aprenderinternet.about.com/od/ConceptosBasico/a/Que-Es-Http.htm.

8. **RAMOS, Rafael.** Cliente FTP en JAVA. [En línea] http://www.compujuy.com.ar/postx.php?id=54.

9. VLFeat. *SIFT Tutorials.* [En línea] http://www.vlfeat.org/overview/sift.html.

10. **SILBERSCHATZ, Galvin.** *Fundamentos de Sistemas Operativos, 7ª edición.* s.l. : McGraw Hill, 2006.

11. **VOHRA, Deepak.** Create Web services with Ruby on Rails and Action Web Service. [En línea] http://www.ibm.com/developerworks/opensource/library/os-ws-rubyrails/.

12. **KEOGH, Jim.** *J2EE the Complete Reference.* s.l. : McGraw Hill, 2002.

13. **SKLAR, David.** *Learning PHP5.* s.l. : O'Reilly Media, 2004.

14. **TATE, Bruce y HIBBS, Curt.** *RUBY ON RAILS.* s.l. : Anaya Multimedia, 2007.

15. The world's most popular open source database. *MySQL.* [En línea] http://www.mysql.com/.

16. **LEE, James y WARE, Brent.** *Open Source Development with LAMP.* s.l. : Addison-Wesley Educational Publishers Inc., 2002.

17. *FTP (File Transfer Protocol).* [En línea] 18 de Mayo de 2013. http://www.efficienthosting.co.uk/support/ftp.html.

18. GIT. [En línea] http://git-scm.com/.

19. **CRAIG, Larman.** *UML y Patrones.* s.l. : PEARSON, 2006.

20. **PRESSMAN, Roger.** *Ingeniería del Software, 6ª edición.* s.l. : McGraw Hill, 2005.

21. *Introducing JSON.* [En línea] 18 de Mayo de 2013. http://www.json.org/.

22. **GROMPONE, Rafael.** LSD: a Line Segment Detector. [En línea] http://www.ipol.im/pub/art/2012/gjmr-lsd/.

23. KSOAP2-ANDROID PROJECT. *ksoap2-android.* [En línea] http://code.google.com/p/ksoap2-android/.

24. GOOGLE PROJECT HOSTING. *ZXing (Zebra Crossing).* [En línea] http://code.google.com/p/zxing/.

25. **Dennis M, Ritchie.** The Development of the C Language. [En línea] http://cm.bell-labs.com/cm/cs/who/dmr/chist.html.

26. Android SDK. [En línea] http://developer.android.com/sdk/index.html.

27. **Bjarne, Stroustrup.** *El lenguaje de programación C++.* Madrid : Addison Wesley, 1998.

28. JNI Java Native Interface. [En línea] http://docs.oracle.com/javase/6/docs/technotes/guides/jni/.

29. Android NDK. [En línea] http://developer.android.com/tools/sdk/ndk/index.html.

30. Tutorial de Canvas. [En línea] https://developer.mozilla.org/es/docs/Canvas\_tutorial.

31. CODENAME ONE. *Android Developer Introduction to Codename One.* [En línea] http://www.codenameone.com/android.html.

32. The Open Source Initiative. [En línea] http://opensource.org/.

33. **Elmasri, Ramez y Navathe, Shamakant.** *SISTEMAS DE BASE DE DATOS. Conceptos fundamentales. Tercera Edición.* s.l. : Addison – Wesley, 1997.

34. **CHISNALL, David.** *Objective-C Phrasebook.* s.l. : Addison-Wesley Professional, 2011.

35. GRAPHICSMAGIG GROUP. *GraphicsMagick API.* [En línea] http://www.graphicsmagick.org/1.1/www/api.html.