



**UNIVERSIDAD DE TALCA  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**Prototipo de una Red Geosocial para crear,  
compartir y visualizar mapas con información  
personalizada: MyGIS**

**FRANCISCO JAVIER GARRAO MUÑOZ**

Profesor Guía: PABLO IGNACIO ROJAS VALDÉS

Memoria para optar al título de  
Ingeniero Civil en Computación

Curicó – Chile  
Diciembre, 2015

## CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Curicó, 2019

*Dedicado a mi familia, especialmente a mis padres María Soledad y Sergio, quienes siempre lo han dado todo por sus hijos. Gracias a su apoyo pude terminar mi carrera, la que me da una importante herramienta para cumplir mis objetivos como persona y profesional. Espero usar las habilidades adquiridas durante este tiempo para ser un aporte para la sociedad, como ellos me lo inculcaron.*

## AGRADECIMIENTOS

A través de estas líneas quiero expresar mi más sincero agradecimiento a todas las personas que con su apoyo técnico y humano han colaborado en la realización de este trabajo.

En primer lugar agradecer a la Universidad de Talca, a sus profesores, funcionarios y alumnos, por haber sido parte fundamental de una importante etapa de la vida y haber aportado en mi formación profesional y humana.

Agradezco a mis amigos y familiares, que siempre estuvieron apoyándome y dándome ánimos para poder finalizar de forma exitosa los desafíos y dificultades que se me presentaron en este periodo.

Agradezco a mis hermanos, por el apoyo y confianza que han depositado en mí como hermano mayor.

Finalmente agradezco a mis padres, porque son el principal apoyo y gracias a ellos y su gran esfuerzo puedo dar fin de forma satisfactoria a este trabajo y esta etapa de la vida.

## TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VII
Índice de Tablas	VIII
Resumen	IX
<b>1. Introducción</b>	<b>10</b>
1.1. Descripción del Problema . . . . .	12
1.2. Objetivos . . . . .	14
1.2.1. Objetivos Generales . . . . .	14
1.2.2. Objetivos Específicos . . . . .	14
1.3. Alcance . . . . .	16
1.3.1. Restricciones . . . . .	18
<b>2. Marco Teórico</b>	<b>19</b>
2.1. La información . . . . .	19
2.2. Comunicación . . . . .	19
2.2.1. Evolución mediante la comunicación . . . . .	19
2.2.2. Comunicación mediante la computación . . . . .	21
2.2.3. Tecnologías de la Información y la Comunicación (TIC) . . . . .	21
2.3. Sistemas de Información (SI) . . . . .	22
2.3.1. Actividades que realiza un SI . . . . .	22
2.3.2. Caracterización de un SI . . . . .	24
2.4. Sistemas de Información Geográfico (SIG) . . . . .	24
2.4.1. Funcionamiento de un SIG . . . . .	25
2.4.2. La creación de datos . . . . .	25

2.4.3.	La representación de los datos . . . . .	26
2.5.	Dispositivo móvil . . . . .	29
2.5.1.	Características de los dispositivos móviles . . . . .	29
2.5.2.	Sistemas Operativos Móviles . . . . .	30
2.6.	Sistema de Posicionamiento Global (GPS) . . . . .	31
2.7.	Redes Sociales . . . . .	31
2.8.	Arquitectura de software . . . . .	32
2.8.1.	Arquitectura Cliente-Servidor (C/S) . . . . .	32
2.8.2.	Arquitecturas multicapas . . . . .	33
2.8.3.	Arquitectura Modelo Vista Controlador (MVC) . . . . .	34
2.8.4.	Arquitectura Orientada a Servicios (SOA) . . . . .	37
2.9.	Lenguajes, librerías y frameworks . . . . .	39
2.9.1.	HTML5 . . . . .	39
2.9.2.	AJAX . . . . .	41
2.9.3.	JQuery . . . . .	42
2.9.4.	OpenStreetMap . . . . .	44
2.9.5.	OpenLayers . . . . .	45
2.9.6.	TinyMCE . . . . .	46
2.9.7.	HelloJS . . . . .	46
2.10.	Metodología de desarrollo de software . . . . .	46
2.10.1.	Metodología en espiral . . . . .	47
2.10.2.	Metodología basada en prototipos . . . . .	49
<b>3.</b>	<b>Metodología de desarrollo.</b>	<b>52</b>
3.1.	Puntos importantes para la elección. . . . .	52
3.2.	Metodología de desarrollo de software. . . . .	53
3.3.	Descripción de la metodología. . . . .	54
3.4.	Fases del ciclo de vida del proyecto. . . . .	54
3.4.1.	Análisis. . . . .	54
3.4.2.	Diseño. . . . .	55
3.4.3.	Implementación. . . . .	55
3.4.4.	Evaluación. . . . .	57
3.5.	Diagrama de la metodología a utilizar. . . . .	57

<b>4. Análisis</b>	<b>58</b>
4.1. Especificación de los requisitos . . . . .	58
4.1.1. Estado del arte . . . . .	59
4.1.2. Conclusiones del análisis . . . . .	60
4.2. Requisitos . . . . .	62
4.2.1. Funcionales . . . . .	62
4.2.2. No Funcionales . . . . .	67
<b>5. Diseño</b>	<b>69</b>
5.1. Aspectos importantes a considerar . . . . .	69
5.2. Diseño arquitectónico . . . . .	70
5.2.1. Elección de la arquitectura . . . . .	70
5.2.2. Diagrama de la arquitectura . . . . .	71
5.3. Diseño del modelo lógico . . . . .	71
5.3.1. Diseño Base de Datos . . . . .	71
5.3.2. Diagrama de clases . . . . .	73
5.4. Diseño vista . . . . .	73
5.4.1. Editor de Mapas . . . . .	74
5.4.2. Visualizador de Mapas . . . . .	74
5.5. Diseño Controlador . . . . .	74
5.5.1. Elección de plataforma de desarrollo . . . . .	75
5.6. Diagrama estructural final . . . . .	75
<b>6. Implementación</b>	<b>77</b>
6.1. Entorno de Desarrollo . . . . .	77
6.1.1. Hardware utilizado en el desarrollo . . . . .	77
6.1.2. Software utilizado en el desarrollo . . . . .	78
6.1.3. Base de Datos (BD) . . . . .	79
6.2. Prototipos . . . . .	79
6.2.1. Primera iteración . . . . .	80
6.2.2. Segunda iteración . . . . .	81
6.2.3. Tercera iteración . . . . .	83
6.2.4. Cuarta iteración . . . . .	85
6.3. Manejo de datos . . . . .	88

6.3.1. Comunicación con el servidor . . . . .	89
6.3.2. Conexión con la Base de Datos . . . . .	89
6.3.3. Servicios Web . . . . .	89
6.4. Manual de usuario . . . . .	92
6.4.1. Descripción del manual . . . . .	93
6.5. Capturas del producto final . . . . .	94
<b>7. Evaluación</b>	<b>95</b>
7.1. Pruebas de usuarios . . . . .	95
7.1.1. Cuestionario . . . . .	96
7.1.2. Resultado de las pruebas de usuario . . . . .	100
7.2. Resultados final de la evaluación . . . . .	102
<b>8. Conclusiones</b>	<b>104</b>
8.1. Trabajo futuro . . . . .	105
<b>Glosario</b>	<b>107</b>
<b>Bibliografía</b>	<b>108</b>

## ÍNDICE DE FIGURAS

	página
2.1. Pasos para convertir datos en sabiduría . . . . .	20
2.2. Funcionamiento de un SI . . . . .	24
2.3. Raster y vectorial . . . . .	27
2.4. Funcionamiento de un Sistema Operativo . . . . .	31
2.5. Diagrama arquitectura C/S . . . . .	33
2.6. Flujo MVC . . . . .	35
2.7. Tipos de aplicaciones SOA . . . . .	37
2.8. Como trabaja AJAX . . . . .	42
2.9. Metodología en espiral . . . . .	48
2.10. Metodología basada en prototipos . . . . .	50
3.1. Metodología de desarrollo de software . . . . .	57
5.1. Diagrama de clases de MyGIS . . . . .	73
5.2. Diagrama arquitectura de MyGIS . . . . .	76
6.1. Captura del producto de la primera iteración . . . . .	81
6.2. Captura del producto de la segunda iteración . . . . .	83
6.3. Captura del producto de la tercera iteración . . . . .	85
6.4. Captura del producto de la cuarta iteración: Vista Editor . . . . .	87
6.5. Captura del producto de la cuarta iteración: Vista Visor . . . . .	87

## ÍNDICE DE TABLAS

	página
7.1. Usuarios con poca experiencia y teléfono móvil. . . . .	100
7.2. Usuarios con poca experiencia y computadora personal. . . . .	100
7.3. Usuarios con experiencia media y teléfono móvil. . . . .	100
7.4. Usuarios con experiencia media y computadora personal. . . . .	100
7.5. Usuarios experimentados y teléfono móvil. . . . .	100
7.6. Usuarios experimentados y computadora personal. . . . .	100
7.7. Promedios de la evaluación de usuario. . . . .	101

## RESUMEN

El siguiente trabajo de memoria, trata sobre cómo se diseñó e implementó un sistema de información geográfica libre y multiplataforma, con la capacidad de crear, visualizar y compartir mapas con información georeferenciada a través de redes sociales.

Con la idea de generar una herramienta computacional que entregue la capacidad de comunicar información variada, de una forma eficiente y novedosa. Se pensó en elaboración de un software que integre las tecnologías más modernas y utilizadas hoy en día. Es así como surgió el propósito de este proyecto de crear el prototipo de una red social con funcionalidades georeferenciales.

En este documento contiene todos los aspectos que se estudiaron y todas las decisiones que se tomaron para el desarrollo de este sistema. Desde la elección de una metodología de trabajo, hasta la evaluación y conclusiones del sistema, una vez terminado.

Los puntos más importantes referente al desarrollo del sistema, que se pueden encontrar en este documento son los siguientes:

- Presentación del problema y objetivos.
- Investigación de conceptos para el desarrollo del sistema.
- Elección de la metodología de trabajo para el desarrollo del sistema.
- Definición de los requisitos del sistema.
- Diseño arquitectónico y diseño gráfico del sistema.
- Aspectos importantes de la implementación del sistema.
- Evaluación de los resultados del desarrollo del sistema.
- Conclusiones obtenidas del trabajo realizado.

# 1. Introducción

---

“Todos los hombres tienen por naturaleza el deseo de saber”

(Aristóteles).

Esta frase de Aristóteles refleja muy bien la naturaleza del hombre de satisfacer la propia curiosidad por saber y conocer el mundo que le rodea. Es por esto que a lo largo de la historia el ser humano ha realizado un gran esfuerzo por prolongar el saber. A lo largo del tiempo, el humano ha creado muchos mecanismos para perpetuar y transferir el conocimiento. Esto mediante la representación de datos a través de diversos medios físicos. Estos mecanismos han ido evolucionando a lo largo del tiempo, para generar medios por los cuales almacenar y/o transmitir un mayor volumen de información y de una forma más efectiva.

La búsqueda del conocimiento y los medios para conseguirlo, ha llevado a que en la actualidad sea muy fácil el acceso a la información. Esta facilidad de acceso se debe principalmente a las herramientas entregadas por los avances tecnológicos actuales. Tanta es la facilidad de acceso y la sobreexposición a datos de distintas índoles, que el período actual que estamos viviendo se ha bautizado como *“La era de la información”*. Dicha era se caracteriza por la facilidad y rapidez con la que se puede acceder y transmitir grandes volúmenes de datos e información, a los cuales hace algunos años no era tan simple de acceder, incluso, en algunos casos imposible.

Este fenómeno de sobre exposición de datos comenzó con la aparición del internet y se ha ido desarrollando cada vez más. Un fenómeno importante en el desarrollo del internet es la aparición de los dispositivos móviles, que aprovechan las características

que entrega esta red de comunicación mundial. Además estos dispositivos han ido incorporando nuevas características para que sea más sencilla y rápida la captura y transmisión de esos datos.

Estos dispositivos son muy comunes hoy en día y cada vez se hacen más populares. Una característica importante en la cual se centra este trabajo, es que la mayoría de esos dispositivos móviles traen incorporados un sistema de posicionamiento global (GPS).

En la actualidad es muy común que las personas lleven consigo alguno de esos dispositivos móviles, que tienen acceso a internet e incorporan un sistema GPS. Es por esto que se puede determinar en cualquier momento y lugar, la posición del dispositivo, por ende, de la persona que lo lleva consigo. Gracias a software especializados para estos dispositivos, se pueden determinar los lugares y elementos que se encuentran en las cercanías a éste.

Otro fenómeno que ha tenido gran impacto en la sociedad (que ha surgido gracias a internet) son las redes sociales. Las redes sociales ayudan al usuario a relacionarse con otros en función de los datos que él mismo proporciona. Dado todo esto se hace natural la relación entre las redes sociales y los dispositivos móviles.

## 1.1. Descripción del Problema

La clave del éxito del ser humano sobre las demás especies vivientes, es su capacidad de aprender y adaptarse a su entorno. Con esta capacidad fue creando y desarrollando cultura. La cultura se define como el conocimiento y costumbres que se van transmitiendo entre individuos y durante generaciones. La cultura va definiendo el modo de vivir del ser humano, y constituye la base con la cual se enfrentará al mundo. Para transmitir la cultura y cualquier otro conocimiento, éste comenzó a desarrollar distintas técnicas y herramientas a lo largo de su historia.

La comunicación del ser humano ha ido evolucionando a lo largo del tiempo. Desde una forma de comunicación más amplia y abstracta, hasta la comunicación de elementos y conocimientos más específicos y detallados. Así fue como a lo largo de la historia de la humanidad se utilizaron diversos mecanismos de comunicación. En un comienzo eran sonidos guturales y señas corporales, luego la comunicación humana fue evolucionando. Pasando por un lenguaje más sofisticado, con vocalización de palabras y la utilización de herramientas para transmitir y retener información.

Las primeras herramientas que se utilizaron con el fin de comunicar algo, fueron elementos que entregaba la naturaleza. Eran objetos que pertenecían al paisaje, como rocas y madera, en las que podían trazar símbolos e imágenes, con los cuales representaban lo que deseaban comunicar. Luego surgió la escritura y medios más efectivos, en los cuales escribir o representar la información (como el papel y tintes más sofisticados), que luego se convirtieron en papiros y libros. Así fue evolucionando la forma de comunicación y los medios por los cuales era posible transmitir información. Creando medios de comunicación a distancias y velocidades cada vez mayores.

En la actualidad, estos mecanismos de comunicación han evolucionado a tal nivel que se tiene la posibilidad de comunicarse a cualquier lugar del mundo (y fuera de él), en forma casi instantánea. Existe la posibilidad de conocer y experimentar a través de diversos sentidos, acontecimientos sucedidos en diferentes coordenadas espaciales y temporales, lo que ha hecho que la comunicación sea más efectiva.

Con el afán de seguir desarrollando tecnologías y formas de comunicación. Con

las cuales se pueda transmitir y almacenar información de forma más efectiva y novedosa, se ha ideado el desarrollo de este trabajo de memoria. El proyecto consiste en el diseño y desarrollo de una herramienta computacional efectiva y novedosa, la cual aproveche los últimos avances tecnológicos. Se pretende crear una herramienta útil, que ofrezca nuevas formas de transmitir información.

Esto se pretende lograr mediante el diseño e implementación de una herramienta computacional, la cual permitirá transmitir y almacenar información de forma efectiva y novedosa para generar un aporte en la forma de comunicar una información. En un constructor y un visualizador de mapas. Estas herramientas tienen la capacidad de poder crear y visualizar distintas capas interactivas. Con las cuales se puede representar y transmitir información diversa, vinculada un lugar o zona física específica. Esto es posible gracias a las tecnologías existentes en la actualidad, las cuales por otro lado son bien aceptadas y atractivas para la mayoría de las personas.

El constructor de mapas está dirigido a cualquier persona que desee transmitir información sobre un lugar u objeto físico, que se pueda coordinar espacialmente. El software permitirá crear múltiples capas sobre un mapa e ingresar toda la información que se desee. En cada capa se podrán trazar zonas, rutas u objetos, a los cuales se les podrá asignar información y datos diversos.

El visualizador de mapas va dirigido a una persona que desee recepcionar la información creada con el constructor de mapas. El acceso a la información será de una manera fácil e interactiva, para que así sea más fácil la comprensión de la información que se desea transmitir. Se aprovechará la interfaz táctil de los dispositivos móviles, con la cual se permitirá seleccionar cualquier zona, ruta u objeto que haya sido georreferenciado previamente con el constructor de mapas.

## **1.2. Objetivos**

### **1.2.1. Objetivos Generales**

El presente trabajo de memoria tiene como principal objetivo generar un prototipo de una red social, la cual permitirá crear, compartir y visualizar mapas con información personalizada. Con el propósito de generar una herramienta tecnológica que permita compartir conocimientos de forma eficaz.

### **1.2.2. Objetivos Específicos**

Para poder lograr un aporte en la forma de comunicar información, se hará uso de las últimas y más populares herramientas de comunicación que existen en la actualidad. Es por eso que se pretende construir un prototipo de una red social. La cual permitirá crear, modificar, compartir y visualizar información diversa, la cual se podrá vincular a un lugar geo-espacial.

Requerimientos generales:

- La aplicación debe permitir crear múltiples capas de información, independientes unas de otras, solapadas a un mapa base que servirá para guiarse geoespacialmente.
- La aplicación debe permitir compartir los mapas mediante redes sociales.
- La aplicación debe permitir ingresar información de forma libre y variada (a través de textos con formato, imágenes, vídeos, tablas, esquemas, etc...).
- La aplicación debe ser web y con una interfaz gráfica compatible con diversos dispositivos móviles.

Puntualmente los objetivos específicos son:

- Diseñar una aplicación web que cumpla con los requerimientos anteriormente mencionados.
- Implementar la aplicación web diseñada.
- Testear la aplicación web implementada.

### 1.3. Alcance

El presente proyecto, el cual pretende generar un prototipo de una red social, la que entregue una nueva y efectiva forma de compartir información geo-espacial, se implementará mediante la creación de una Red Geosocial.

El sistema será una red geosocial, esto quiere decir que tiene las características de una red social, al poder compartir información entre las personas, e incluye funcionalidades relacionadas con la georreferenciación. Esta red geosocial contará con dos vistas distintas. Esto permitirá separar la interfaz gráfica de los usuarios que deseen ingresar información al sistema, con la de los usuarios que desean obtener esa información. De esa manera una interfaz contará con herramientas especializadas en trazar las capas con sus elementos e información respectivos, y la otra se centrará más en la navegación dentro del mapa y la entrega de información. Ambas vistas deben tener una interfaz compatible con dispositivos móviles. Además, el sistema debe ser multiplataforma, por lo que la implementara como una aplicación web, para ser accesible desde cualquier dispositivo que cuente con internet y un browser que soporte HTML5.

#### **El editor de mapas podrá:**

- Crear múltiples mapas y guardarlos en una base de datos.
- Cargar y editar mapas creados con anterioridad.
- Crear múltiples capas dentro de un mapa, de forma dinámica.
- Importar y editar capas creadas con anterioridad.
- En las capas se podrán trazar distintos elementos, que representan distintos objetos.
- Los elementos que se podrán trazar en una capa son: polígonos (que representaran zonas), líneas (que representaran rutas) y puntos (que representaran objetos específicos).

- A los elementos que se trazan en las capas, se les podrá introducir distintos tipos de datos tales como: texto, imágenes, vídeos, hipervínculos, estadísticas, etc.
- Se podrán incluir un servidor de mapas, el cual servirán como guía para construir las capas.

**El visor de mapas podrá:**

- Ser utilizado en distintos dispositivos móviles.
- Cargar mapas creados previamente con el constructor.
- Visualizar las distintas capas que contenga el mapa y los elementos georeferenciados contenidos por las capas.
- Visualizar la información que contengan los elementos georeferenciados.
- Subir comentarios, fotografías o videos, para transmitir la experiencia vivida en un lugar determinado.
- Visualizar comentarios, fotografías o videos, subidos por otros usuarios, para transmitir la experiencia vivida en un lugar determinado.
- Obtener información de la red social Foursquare.
- Visualizar información de la red social Foursquare.

### 1.3.1. Restricciones

Como toda aplicación, el sistema tiene ciertos requerimientos para poder funcionar en un determinado dispositivo

#### Requisitos de sistema:

- Internet: Como el sistema es una aplicación web, para ejecutarlo en un dispositivo, es necesario que este cuente con conexión a internet.
- Navegador Web: El sistema está construido con la más reciente tecnología de desarrollo web, por lo que necesita de un navegador moderno, que tenga soporte para HTML5.
- Pantalla: el sistema está diseñado con una interfaz compatible con dispositivos móviles, por lo que es funcional en pantallas de poca resolución. Sin embargo la considerable cantidad de opciones que presenta la vista del Editor de Mapas?, hace mas recomendable ejecutar la aplicación en una pantalla con una resolución mínima de 480 pixeles de ancho.
- Periférico de entrada: El sistema necesita controles de entrada tanto para navegar por el mapa, como para ingresar datos en los formularios e información en los elementos. El sistema necesita un teclado y un mouse o bien una pantalla táctil para poder interactuar con el sistema.
- GPS: Aunque para utilizar la mayoría de las funciones del sistema no es indispensable el GPS. Este sirve para darle mayor precisión a la función de localización que tiene la aplicación.

## 2. Marco Teórico

---

### 2.1. La información

La información se distingue de los datos por sus características de utilidad. Un dato describe un hecho o un objeto de una manera objetiva y según unos parámetros aceptados. La información contiene datos que explicados en un contexto tienen una utilidad para poder generar más información, para poder tomar decisiones o incluso, para poder razonar.[2]

### 2.2. Comunicación

#### 2.2.1. Evolución mediante la comunicación

Si evolucionamos es porque nos comunicamos. Los seres humanos evolucionan porque se comunican. Por ejemplo, después de haber descubierto el fuego y de haberlo compartido, ya no hacía falta redescubrirlo: sólo había que comunicarlo. Un ejemplo más moderno sería el descubrimiento de la estructura helicoidal del ADN, moléculas que transportan información genética de una generación a la siguiente. Una vez que el artículo de James Watson y Francis Crick apareció en una publicación científica en abril de 1953, las disciplinas de la medicina y la genética pudieron tomar esta información y avanzar desde allí a pasos agigantados. Este principio de compartir información y aprovechar los descubrimientos se puede comprender mejor si se analiza la manera en que los seres humanos procesan los datos (ver Figura 2.1).



Figura 2.1: Pasos para convertir datos en sabiduría

Desde la base hasta la cúspide, las capas de la pirámide incluyen datos, información, conocimiento y sabiduría. Los datos representan la materia prima que se procesa para obtener información. Los datos individuales por sí mismos no son muy útiles, pero en volúmenes permiten identificar tendencias y patrones. Esta y otras fuentes de información se unen para conformar el conocimiento. En su sentido más básico, el conocimiento es la información de la que alguien es consciente. Luego, la sabiduría nace de la combinación de conocimiento y experiencia. En tanto que el conocimiento cambia con el tiempo, la sabiduría es atemporal, y todo comienza con la adquisición de datos.

También resulta importante destacar que existe una correlación directa entre la entrada (datos) y la salida (sabiduría). Cuántos más datos se generan, más conocimiento y sabiduría pueden obtener las personas. También mientras más diversa la manera de representar los datos, es más fácil que el receptor pueda captar la información de manera más efectiva.

### 2.2.2. Comunicación mediante la computación

La comunicación por computadora es el intercambio de la información entre las personas por las redes de ordenadores, puede ser todo tipo de información, por ejemplo texto, imágenes, audio, y vídeo. El intercambio de la información puede ser en tiempo real o síncrono, esto significa que la gente se está comunicando de forma simultánea.

Otra forma de comunicación es la asíncrona, en donde las personas se comunican a diversas horas. Pueden enviar y recibir sus mensajes en cualquier momento. Un ejemplo muy claro de esto es el correo electrónico.[1]

### Medios de la comunicación mediante la computación

Existen cinco clases de medios: texto, gráficos, imágenes, audio y video (Walters 1995, Agnew y Kellerman, 1996):

- el texto consiste en letras, números, puntuación, caracteres especiales, y controles.
- los gráficos son líneas, círculos, rectángulos, sombreado, los colores, etc.
- las imágenes son cuadros inmóviles, expresados en pixeles, éstos pueden ser fotografías o pinturas.
- el audio consiste, voz, música, y efectos especiales.
- el video consiste en cuadros sucesivos presentados a gran velocidad, para dar el aspecto del movimiento.

Los medios que la comunicación por computadora utiliza son importantes en el proceso de aprendizaje, porque definen la manera en que la información será transmitida.

### 2.2.3. Tecnologías de la Información y la Comunicación (TIC)

Tecnologías de la Información y de la Comunicación se entiende como un término para designar lo referente a la informática conectada a Internet, y especialmente el

aspecto social de estos. Las nuevas Tecnologías de la Información y Comunicación eligen a la vez un conjunto de innovaciones tecnológicas pero también las herramientas que permiten una redefinición radical del funcionamiento de la sociedad.

De una forma más estricta, las TIC son aquellas herramientas computacionales e informáticas que procesan, almacenan, resumen, recuperan y presentan información representada de la más variada forma.

Las Tecnologías de la Información y la Comunicación (la unión de los computadores y las comunicaciones) desataron una explosión sin antecedentes de formas de comunicarse al comienzo de los años 90. A partir de ahí, la Internet pasó de ser un instrumento experto de la comunidad científica, a ser una red de fácil uso que modificó las pautas de interacción social.

### 2.3. Sistemas de Información (SI)

Un Sistema de Información, es aquél que permite recopilar, administrar y manipular un conjunto de datos que conforman la información necesaria para que los estamentos ejecutivos de una organización puedan realizar una toma de decisiones informadamente. En resumen, es aquél conjunto ordenado de elementos (no necesariamente computacionales) que permiten manipular toda aquella información necesaria para implementar aspectos específicos de la toma de decisiones.

#### 2.3.1. Actividades que realiza un SI

Un Sistema de Información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

**Entrada de Información:** Proceso en el cual el sistema toma los datos que requiere para procesar la información, por medio de estaciones de trabajo, teclado, mouse, sensores, código de barras, etc. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.

**Almacenamiento de información:** el almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. Algunas de las unidades típicas de almacenamiento son los discos magnéticos o discos duros, las unidades de estado sólido, pendrives, los discos compactos, etc.

**Procesamiento de Información:** es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.

**Salida de Información:** la salida es la capacidad de un Sistema de Información para presentar la información procesada o bien datos de entrada al exterior. Algunas de las unidades típicas de salida son las impresoras, pantallas, almacenamiento USB, sonidos (mediante los parlantes), entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a salida.

Se puede apreciar cómo interactúan estas actividades en la siguiente imagen (Figura 2.2)



Figura 2.2: Funcionamiento de un SI

### 2.3.2. Caracterización de un SI

Los Sistemas de Información difieren de cualquier otro software por dos razones principales:

- Almacenan gran cantidad de Información.
- Realizan un bajo grado de procesamiento sobre la información, y este es, fundamentalmente, de tipo estadístico.

## 2.4. Sistemas de Información Geográfico (SIG)

Un Sistema de Información Geográfica (SIG) en español o Geographic Information System (GIS) en inglés, es un conjunto de herramientas que integra y relaciona diversos componentes (usuarios, hardware, software, procesos). Estas herramientas forman un sistema específico que permite a los usuarios crear consultas interactivas, integrar, analizar y representar de una forma eficiente cualquier tipo de información geográfica referenciada asociada a un territorio, conectando mapas con bases de datos. En el sentido más estricto, es cualquier sistema de información capaz de integrar,

almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada.

El uso de este tipo de sistemas facilita la visualización de los datos obtenidos en un mapa con el fin de reflejar y relacionar fenómenos geográficos de cualquier tipo, desde mapas de carreteras hasta sistemas de identificación de parcelas agrícolas o de densidad de población. Además, permiten realizar las consultas y representar los resultados en entornos web y dispositivos móviles de un modo ágil e intuitivo, con el fin de resolver problemas complejos de planificación y gestión, conformándose como un valioso apoyo en la toma de decisiones.

Por ser tan versátiles, el campo de aplicación de los Sistemas de Información Geográfica es muy amplio, pudiendo utilizarse en la mayoría de las actividades con un componente espacial.

#### **2.4.1. Funcionamiento de un SIG**

El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía.

La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, y facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no se podría obtener de otra forma.

#### **2.4.2. La creación de datos**

Las modernas tecnologías SIG trabajan con información digital, para la cual existen varios métodos utilizados en la creación de datos digitales. El método más utilizado es la digitalización, donde a partir de un mapa impreso o con información

tomada en campo se transfiere a un medio digital por el empleo de un programa de Diseño Asistido por Ordenador (DAO o CAD) con capacidades de georreferenciación.

Dada la amplia disponibilidad de imágenes orto-rectificadas (tanto de satélites, como aéreas), la digitalización por esta vía se está convirtiendo en la principal fuente de extracción de datos geográficos. Esta forma de digitalización implica la búsqueda de datos geográficos directamente en las imágenes aéreas en lugar del método tradicional de la localización de formas geográficas sobre un tablero de digitalización.

### **2.4.3. La representación de los datos**

Los datos SIG representan los objetos del mundo real (carreteras, el uso del suelo, altitudes). Los objetos del mundo real se pueden dividir en dos abstracciones: objetos discretos (una casa) y continuos (cantidad de lluvia caída, una elevación). Existen dos formas de almacenar los datos en un SIG:

1. Raster.
2. Vectorial.

A continuación se muestra una imagen con estas dos formas de representación de datos (Figura 2.3).

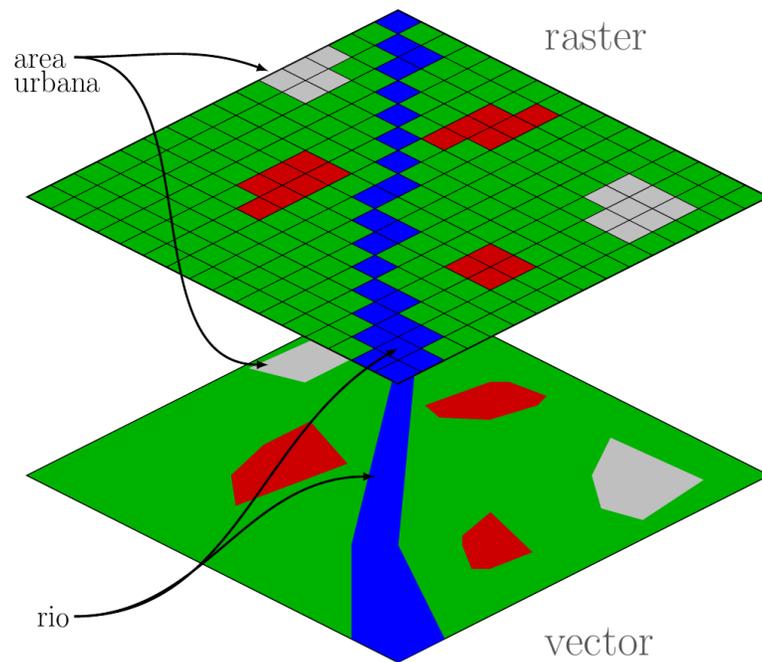


Figura 2.3: Raster y vectorial

Los SIG que se centran en el manejo de datos en formato vectorial son más populares en el mercado. No obstante, los SIG raster son muy utilizados en estudios que requieran la generación de capas continuas, necesarias en fenómenos no discretos; también en estudios medioambientales donde no se requiere una excesiva precisión espacial (contaminación atmosférica, distribución de temperaturas, localización de especies marinas, análisis geológicos, etc.).

### Raster

Un tipo de datos raster es, en esencia, cualquier tipo de imagen digital representada en mallas. El modelo de SIG raster o de retícula se centra en las propiedades del espacio más que en la precisión de la localización. Divide el espacio en celdas regulares donde cada una de ellas representa un único valor. Se trata de un modelo de datos muy adecuado para la representación de variables continuas en el espacio.

## Vectorial

En un SIG, las características geográficas se expresan con frecuencia como vectores, manteniendo las características geométricas de las figuras.

Representación de curvas de nivel sobre una superficie tridimensional generada por una malla TIN, la cual es una red de triángulos irregulares. En los datos vectoriales, el interés de las representaciones se centra en la precisión de la localización de los elementos geográficos sobre el espacio y donde los fenómenos a representar son discretos, es decir, de límites definidos.

Para modelar digitalmente las entidades del mundo real se utilizan tres elementos geométricos: el punto, la línea y el polígono.

**Punto:** Los puntos se utilizan para las entidades geográficas que mejor pueden ser expresadas por un único punto de referencia. En otras palabras: la simple ubicación. Por ejemplo, las localizaciones de los pozos, picos de elevaciones o puntos de interés. Los puntos transmiten la menor cantidad de información de estos tipos de archivo y no son posibles las mediciones. También se pueden utilizar para representar zonas a una escala pequeña. Por ejemplo, las ciudades en un mapa del mundo estarán representadas por puntos en lugar de polígonos.

**Líneas o polilíneas:** Las líneas unidimensionales o polilíneas son usadas para rasgos lineales como ríos, caminos, ferrocarriles, rastros, líneas topográficas o curvas de nivel. De igual forma que en las entidades puntuales, en pequeñas escalas pueden ser utilizados para representar polígonos. En los elementos lineales puede medirse la distancia.

**Polígonos:** Los polígonos bidimensionales se utilizan para representar elementos geográficos que cubren un área particular de la superficie de la Tierra. Estas entidades pueden representar lagos, límites de parques naturales, edificios, provincias, o los usos del suelo, por ejemplo. Los polígonos transmiten la mayor cantidad de información en archivos con datos vectoriales y en ellos se pueden medir el perímetro y el área.

## 2.5. Dispositivo móvil

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales.

### 2.5.1. Características de los dispositivos móviles

A continuación se detallan las características que generalmente tienen los dispositivos móviles:

- Son aparatos pequeños.
- La mayoría de estos aparatos se pueden transportar en el bolsillo del propietario o en un pequeño bolso.
- Tienen capacidad de procesamiento.
- Tienen conexión permanente o intermitente a una red.
- Tienen memoria (RAM, tarjetas MicroSD, flash, etc.).
- Normalmente se asocian al uso individual de una persona, tanto en posesión como en operación, la cual puede adaptarlos a su gusto.
- Tienen una alta capacidad de interacción mediante la pantalla o el teclado.

En la mayoría de los casos, un dispositivo móvil puede definirse con cuatro características esenciales que lo diferencian de otros dispositivos que, aunque pudieran parecer similares, carecen de algunas de las características de los verdaderos dispositivos móviles. Estas cuatro características son:

**Movilidad:** Se entiende por movilidad la cualidad de un dispositivo para ser transportado o movido con frecuencia y facilidad. Por tanto, el concepto de movilidad es una característica básica. Los dispositivos móviles son aquellos que son lo suficientemente pequeños como para ser transportados y utilizados durante su transporte.

**Tamaño reducido :** Se entiende por tamaño reducido la cualidad de un dispositivo móvil de ser fácilmente usado con una o dos manos sin necesidad de ninguna ayuda o soporte externo. El tamaño reducido también permite transportar el dispositivo cómodamente por parte de una persona

**Comunicación inalámbrica:** Otro concepto importante es el término inalámbrico. Por comunicación inalámbrica se entiende la capacidad que tiene un dispositivo de enviar o recibir datos sin la necesidad de un enlace cableado.

**Interacción con las personas:** Se entiende por interacción el proceso de uso que establece un usuario con un dispositivo. Entre otros factores, en el diseño de la interacción intervienen disciplinas como la usabilidad y la ergonomía.

### 2.5.2. Sistemas Operativos Móviles

Un Sistema Operativo (SO), es un programa o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes (aunque puede que parte de él se ejecute en espacio de usuario).

A continuación se muestra una imagen del funcionamiento de un Sistema Operativo (Figura 2.4).



Figura 2.4: Funcionamiento de un Sistema Operativo

Como la mayoría de aparatos electrónicos que utilizan microprocesadores, los dispositivos móviles necesitan de un Sistema Operativo para funcionar.

## 2.6. Sistema de Posicionamiento Global (GPS)

Una de las características importantes que presentan la mayoría de los dispositivos móviles modernos, es la capacidad de conocer nuestra posición geográfica en cualquier momento. Esto es posible gracias a la incorporación de GPS en los dispositivos y a otras técnicas que se utilizan para conocer la ubicación de los dispositivos.

## 2.7. Redes Sociales

Red, un término que procede del latín rete, hace mención a la estructura que tiene un patrón característico. Esta definición permite que el concepto se aplique en diversos ámbitos, como la informática (donde una red es un conjunto de equipos interconectados que comparten información).

Social, por su parte, es aquello perteneciente o relativo a la sociedad (el conjunto de individuos que interactúan entre sí para formar una comunidad). Lo social suele implicar un sentido de pertenencia.

La noción de red social, por lo tanto, está vinculada a la estructura donde un grupo de personas mantienen algún tipo de vínculo.

En la actualidad, el concepto se ha actualizado para señalar a un tipo de sitio de Internet o aplicación web, que favorece la creación de comunidades virtuales, en las cuales es posible acceder a servicios que permiten formar grupos según los intereses de los usuarios, compartiendo fotografías, videos e información en general.

## **2.8. Arquitectura de software**

### **2.8.1. Arquitectura Cliente-Servidor (C/S)**

La Arquitectura Cliente-Servidor (C/S) 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. Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta.

Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado, proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura.

A continuación se muestra una imagen de la arquitectura (Figura 2.5).

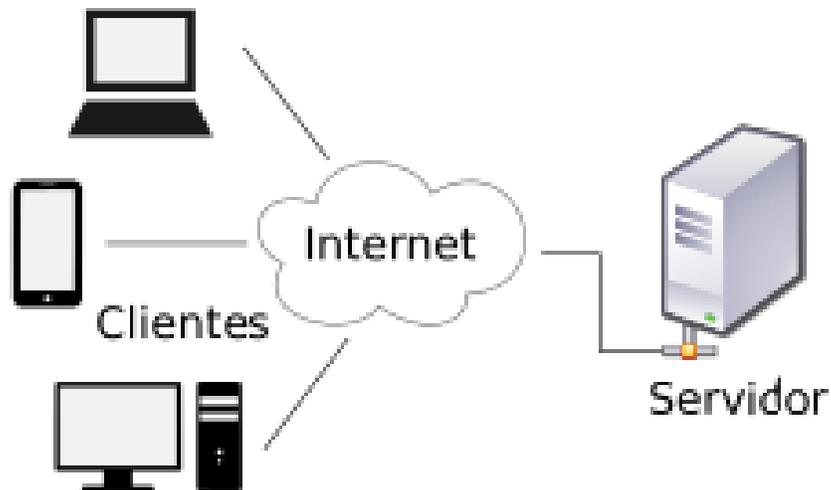


Figura 2.5: Diagrama arquitectura C/S

### 2.8.2. Arquitecturas multicapas

La arquitectura cliente/servidor genérica tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas de dos niveles o dos capas.

Algunas redes disponen de tres tipos de nodos:

1. Clientes que interactúan con los usuarios finales.
2. Servidores de aplicación que procesan los datos para los clientes.
3. Servidores de la base de datos que almacenan los datos para los servidores de aplicación. Esta configuración se llama una arquitectura de tres-capas.

#### Ventajas

La ventaja fundamental de una arquitectura n-capas comparado con una arquitectura de dos niveles (o una tres-capas con una de dos niveles), es que separa hacia fuera el proceso. Eso ocurre para mejorar el balance la carga en los diversos servidores; es más escalable.

### **Desventajas**

Sin embargo, debido a una mayor cantidad de tráfico de la red, es mucho más difícil programar y probar el software que en arquitectura de dos niveles porque tienen que comunicarse más dispositivos para terminar la transacción de un usuario.

### **2.8.3. Arquitectura Modelo Vista Controlador (MVC)**

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software basado en el modelo multicapas (de tres capas). Esta arquitectura separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, los cuales forman las tres distintas capas.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

El modelo, al igual que el controlador y la vista, maneja todos los datos que se relacionen consigo (sólo es el proceso medio de la separación por capas que ofrece la arquitectura MVC). Y sólo la vista, puede demostrar dicha información. Con lo cual ya se ha generado la jerarquía del programa: Controlador, Modelo y Vista.

A continuación se muestra una imagen donde se puede apreciar el flujo de la arquitectura MVC (Figura 2.6).

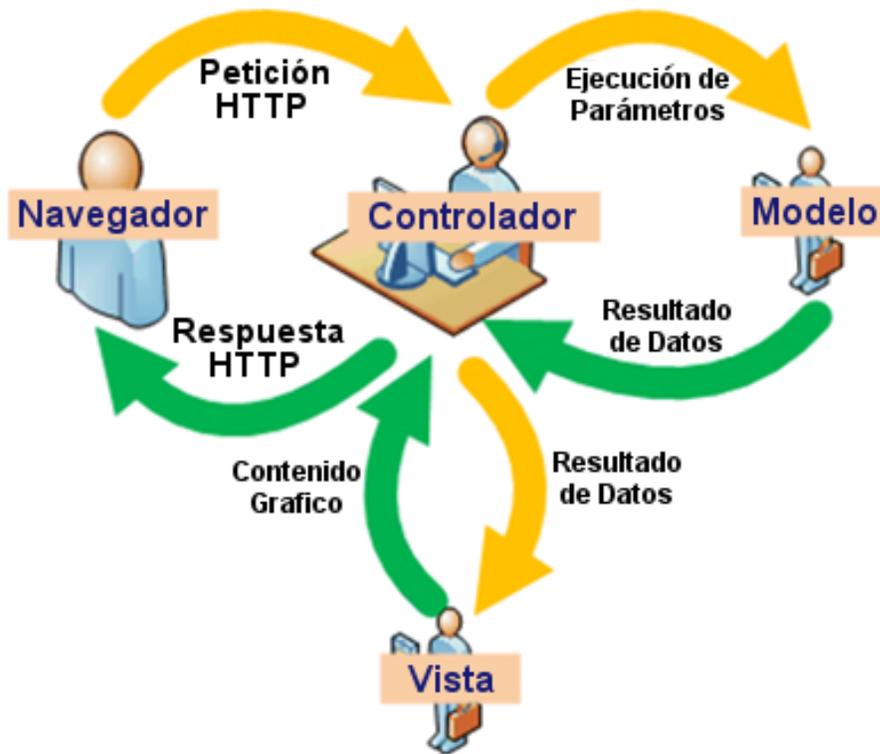


Figura 2.6: Flujo MVC

## Modelo

El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

La capa modelo se encarga de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: “Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor”.
- Lleva un registro de las vistas y controladores del sistema.

- Se está ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

## Vista

La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

La capa vista se encarga de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo “SI Evento Z, entonces Acción W”.
- Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método “Actualizar()”. Una petición al modelo puede ser “Obtener\_tiempo\_de\_entrega (nueva\_orden\_de\_venta)”.

## Controlador

El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

La capa controlador se encarga de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de “Actualización()”, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

### 2.8.4. Arquitectura Orientada a Servicios (SOA)

La Arquitectura Orientada a Servicios (SOA, siglas del inglés Service Oriented Architecture) es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos. SOA se basa en definir la utilización de servicios para dar soporte a ciertos requisitos del negocio.

Existen diversas maneras de implementar la arquitectura SOA en una aplicación. En la siguiente imagen se puede apreciar un diagrama con variadas formas de implementar SOA (Figura 2.7).

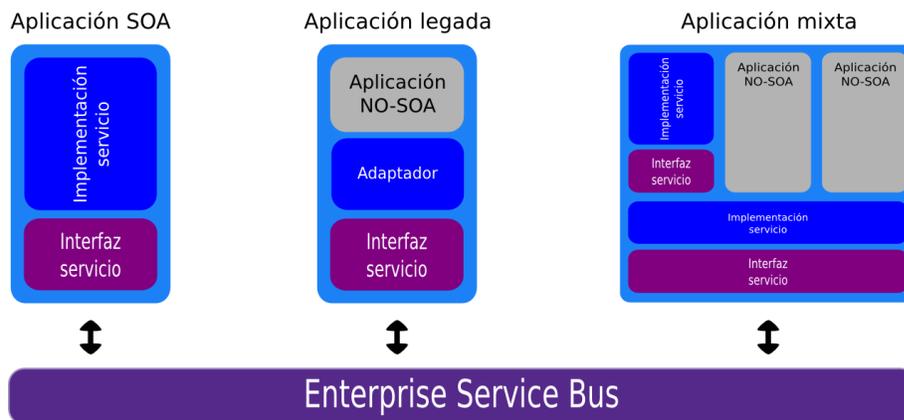


Figura 2.7: Tipos de aplicaciones SOA

Las ventajas de la arquitectura SOA son múltiples y la mayoría de ellas deriva de dos factores. En primer lugar, este enfoque hace posible la reutilización a gran escala del software, con lo que ello implica sobre la calidad, los costos y la facilidad de despliegue. SOA independiza las aplicaciones de la infraestructura y la plataforma tecnológica.

#### Ventajas

La Arquitectura Orientada a Servicios – SOA Service Oriented Architecture – es una manera de diseñar e implementar los procesos de negocios, obteniendo una gran flexibilidad en su mantención y Evolución. Se basa en una forma natural de pensar los procesos, atomizando el conocimiento sin adicionar conceptos propiamente informáticos, hasta el momento de su implementación física. Una de las principales

características de SOA es que resuelve los problemas de conectividad y permite una real reusabilidad masiva y una gran independencia de las plataformas, rentabilizando las ya existentes.

- Reduce el nivel de acoplamiento.
- Clara definición de roles de desarrollo.
- Definición de seguridad más clara.
- Fácil testeo.
- Mejora la mantención.
- Favorece la reutilización.
- Favorece el desarrollo en paralelo.
- Permite fácil escalabilidad.
- Permite un mapeo directo entre los procesos y los sistemas.
- Permite un monitoreo preciso.
- Permite la interoperabilidad.

### **Desventajas**

- SOA depende de la implementación de estándares. Sin estándares, la comunicación entre aplicaciones requiere de mucho tiempo y código.
- SOA no es para: aplicaciones con alto nivel de transferencia de datos, aplicaciones que no requieren de implementación del tipo request/response y para aplicaciones que tienen un corto periodo de vida.
- Incrementalmente se hace difícil y costoso el ser capaz de cumplir con los protocolos y hablar con un servicio.
- Implica conocer los procesos del negocio, clasificarlos, extraer las funciones que son comunes a ellos, estandarizarlas y formar con ellas capas de servicios que serán requeridas por cualquier proceso de negocio.

- En la medida en que un servicio de negocio, vaya siendo incorporado en la definición de los procesos de negocio, dicho servicio aumentará su nivel de criticidad. Con lo cual cada vez que se requiera efectuar una actualización en dicho servicio (por ejemplo, un cambio en el código, una interfaz nueva, etc.), deberá evaluarse previamente el impacto y tener mucho cuidado con su implementación. Sin embargo, parte de la problemática anterior, puede ser solventada en virtud a un buen diseño del servicio.

### **Servicio Web**

Se podría decir que un servicio web es un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.

## **2.9. Lenguajes, librerías y frameworks**

### **2.9.1. HTML5**

HTML5 (HyperText Markup Language, versión 5) es la quinta y más actualizada versión del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como HTML5, y una variante XHTML conocida como sintaxis XHTML5 que deberá servirse con sintaxis XML (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

HTML5 representa dos conceptos diferentes:

1. Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
2. Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5.

Las nuevas características y funciones que ofrece HTML5 son:

**Semántica:** Permite describir con mayor precisión cuál es su contenido.

**Conectividad:** Permite comunicarse con el servidor de formas nuevas e innovadoras.

**Sin conexión y almacenamiento:** Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente.

**Multimedia:** Nos otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.

**Gráficos y efectos 2D/3D:** Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.

**Rendimiento e Integración:** Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.

**Acceso al dispositivo:** Proporciona APIs para el uso de varios componentes internos de entrada y salida de nuestro dispositivo.

**CSS3:** Nos ofrece una nueva gran variedad de opciones para hacer diseños más sofisticados.

## 12 grandes ventajas del HTML5

- Es nativo, y por tanto independiente de plugins de terceros. Es decir, no pertenece a nadie, es open source.
- Es más semántico, con etiquetas que permiten clasificar y ordenar en distintos niveles y estructuras el contenido. Además, incorpora metadatos de manera más formal, favoreciendo el posicionamiento SEO y la accesibilidad.

- El código es más simple lo que permite hacer páginas más ligeras que se cargan más rápidamente favoreciendo la usabilidad y la indexación en buscadores.
- Ofrece una compatibilidad mayor con los navegadores de dispositivos móviles.
- Incluye la etiqueta de dibujo canvas, que ofrece más efectos visuales.
- Ofrece soporte a codecs específicos.
- Posibilita la inserción de vídeos y audio de forma masiva.
- Permite la geolocalización del usuario. Algo muy útil para el marketing móvil.
- Tiene la capacidad de ejecutar páginas sin estar conectado a internet.
- Incorpora nuevas capacidades Javascript que aumentan la capacidad de almacenamiento. Frente a las cookies que dejaban almacenar algunos kilobytes, ahora se puede conseguir el almacenamiento de entre 5 y 10 megas, dependiendo de la plataforma. Además, se permiten múltiples Javascripts corriendo en paralelo en una misma página.
- Dispone de nuevas capacidades CSS3 como posibilidad de usar cualquier fuente o tipografía en HTML, columnas de texto, opacidad, transparencia, canales alpha, contraste, saturación, brillo, animaciones de transición y transformación, bordes redondeados, gradientes, sombras, etc.
- Permite realizar diseños adaptables a distintos dispositivos (web, tablets, smartphones...)

### 2.9.2. AJAX

Aunque AJAX no es en sí un lenguaje de programación, este cuenta con estándares, lo que al momento de usarlo se asemeja a un lenguaje de programación.

AJAX es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (**R**ich **I**nternet **A**pplications). AJAX es el acrónimo de (**A**synchronous **J**avaScript **A**nd **X**ML), funciona de parte del cliente y es una tecnología asíncrona, ya que los datos se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

A continuación se muestra una imagen de como trabaja AJAX (Figura 2.8).

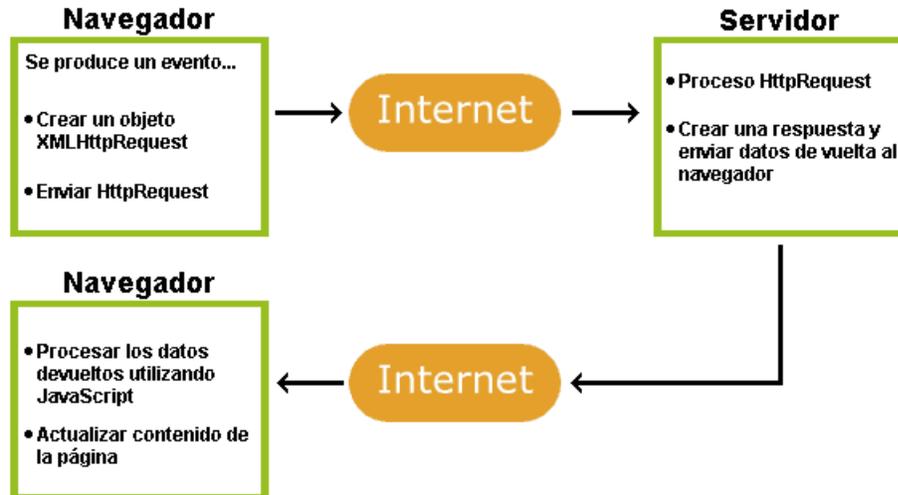


Figura 2.8: Como trabaja AJAX

**Estándares base:** AJAX funciona a través de diversos estándares de Internet, y utiliza una combinación de:

- objeto XMLHttpRequest (para el intercambio de datos de forma asíncrona con el servidor)
- JavaScript/DOM (para visualizar / interactuar con la información)
- CSS (para darle estilo a los datos)
- XML (a menudo utilizado como formato para la transferencia de datos)

### 2.9.3. JQuery

jQuery es una librería de JavaScript liviana, rápida y abundante en funciones. Permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con AJAX a páginas web. jQuery ha cambiado la forma en que millones de personas escriben JavaScript, haciéndolo mucho más simple con una API fácil de usar que funciona a través de una multitud de navegadores. Con una combinación de versatilidad y extensibilidad.

### Características de JQuery

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.

### JQuery Mobile

JQuery Mobile es un Framework optimizado para dispositivos táctiles (también conocido como Framework móvil) que está siendo desarrollado actualmente por el equipo de proyectos de jQuery. El desarrollo se centra en la creación de un Framework compatible con la gran variedad de dispositivos móviles, algo necesario en el creciente y heterogéneo mercado de los dispositivos móviles.

Todos los proyectos que utilizan jQuery Mobile utilizan más o menos el mismo código. Es importante enlazar las librerías JavaScript de jQuery y jQuery Mobile, así como sus hojas de estilo.

#### 2.9.4. OpenStreetMap

OpenStreetMap (también conocido como OSM) es un proyecto colaborativo para crear mapas libres y editables, esto quiere decir que OpenStreetMap lo crea una gran comunidad de colaboradores que con sus contribuciones al mapa añaden y mantienen datos sobre caminos, senderos, cafeterías, estaciones de ferrocarril y muchas cosas más a lo largo de todo el mundo.

Los mapas se crean utilizando información geográfica capturada con dispositivos GPS móviles, ortofotografías y otras fuentes libres. Esta cartografía, tanto las imágenes creadas como los datos vectoriales almacenados en su base de datos, se distribuye bajo licencia abierta Licencia Abierta de Bases de Datos (en inglés ODbL).

##### Característica de OSM

**Conocimiento local:** OpenStreetMap valora mucho el conocimiento local. Los colaboradores utilizan imágenes aéreas, dispositivos GPS, mapas y otras fuentes de datos libres para verificar que los datos de OSM sean precisos y estén actualizados.

**Impulsado por la comunidad:** La comunidad de OpenStreetMap es diversa, apasionada y creciente día a día. Estos colaboradores incluyen a mapeadores entusiastas, profesionales de los SIG, ingenieros que mantienen los servidores de OSM, personas con inquietudes humanitarias que se encargan de cartografiar zonas afectadas por desastres y muchos más.

**Datos abiertos:** OpenStreetMap puede usarse libremente para cualquier propósito, siempre y cuando se de crédito a OpenStreetMap y a sus colaboradores. Si se basa o se alteran los datos en casos determinados, se debe distribuir el resultado únicamente bajo la misma licencia.

**Legal:** El sitio y los muchos servicios relacionados son manejados por la Fundación OpenStreetMap (OSMF) en nombre de la comunidad.

**Socios:** El alojamiento de los servidores está respaldado por el centro VR de la UCL, Imperial College de Londres y Bytemark Hosting, y otros socios.

### 2.9.5. OpenLayers

OpenLayers es una librería SIG desarrollado en JavaScript y de código abierto, para acceder, manipular y mostrar mapas web dinámicos.

OpenLayers funciona bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores web. OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red.

#### Funcionamiento

Programando en Javascript es posible adoptar un enfoque orientado a objetos. De esta forma, mediante el mecanismo de ir extendiendo las clases según una jerarquía determinada, se van obteniendo objetos adaptados a ciertas necesidades particulares que reducirán el tiempo de los desarrollos futuros.

La API de OpenLayers tiene dos conceptos que es importante entender:

**Mapa (Map):** Un mapa de OpenLayers almacena información sobre la proyección por defecto, la extensión, unidades y demás propiedades del mapa. Dentro del mapa, los datos se muestran a través de las Capas.

**Capa (Layer):** Una Capa es una fuente de datos-información sobre cómo OpenLayers debe pedir los datos y representarlos.

#### OpenLayers Editor (OLE)

OpenLayers Editor (OLE) ofrece un conjunto de controles para la edición extendida de datos espaciales. Añade a un mapa de OpenLayers herramientas de edición de la geometría en el navegador sin necesidad de utilizar plugins.

OLE se puede enlazar a un rango de back-ends usando JavaScript personalizado o simplemente estar unido mediante WFS-T.

### 2.9.6. TinyMCE

TinyMCE es un editor de texto WYSIWYG para HTML de código abierto que funciona completamente en JavaScript y se distribuye gratuitamente bajo licencia LGPL.

Al ser basado en JavaScript TinyMCE es independiente de la plataforma y se ejecuta en el navegador de internet. Tiene la habilidad de convertir un campo del tipo textarea u otros elementos de html en instancias del editor. Se puede integrar fácilmente a cualquier CMS. Incluye el idioma español.

### 2.9.7. HelloJS

HelloJS es una librería JavaScript SDK para autenticar servicios web y consultar APIs REST. HelloJS estandariza caminos y respuestas a las API comunes como Google Data Services, Facebook Graph y Windows Live Connect. Es modular, de modo que la lista está creciendo.

## 2.10. Metodología de desarrollo de software

Una metodología de desarrollo de software es una perspectiva de las actividades que ocurren durante el diseño y el desarrollo del software, se pretende determinar el orden de las etapas implicadas en el sistema y los criterios de transición asociadas entre estas etapas. En las funciones de una metodología de desarrollo de software se incluye:

- Describe las etapas primordiales del desarrollo de software.
- Define las etapas primarias esperadas para ser aplicadas durante esas etapas.
- Ayuda a administrar el progreso del desarrollo
- Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

### 2.10.1. Metodología en espiral

El modelo espiral en el desarrollo del software es un modelo meta del ciclo de vida del software donde el esfuerzo del desarrollo es iterativo, tan pronto culmina un esfuerzo del desarrollo por ahí mismo comienza otro. En cada ejecución del desarrollo se sigue cuatro etapas principales.

#### Etapas

**1. Análisis de requerimientos:** Durante esta etapa se estudia detalladamente los requerimientos que cada objetivo conlleva. Aquí establecen todos los detalles funcionales deseados.

**2. Diseño del sistema:** Con los datos de la etapa anterior, se diseña el sistema. Se realiza el diseño de la base de datos (en caso de ser aplicable), interfaz de usuario y entorno.

**3. Etapa de construcción:** La etapa de construcción comprende básicamente la codificación y test de unidades. Esta etapa es un trabajo de programación pura.

**4. Evaluación:** En esta etapa se realiza un test del módulo completo así como su evaluación frente al estudio de requerimientos. En muchos casos en esta etapa los usuarios finales participan de manera activa aportando información decisiva para la usabilidad del sistema.

En la siguiente imagen se muestra un diagrama con el ciclo de vida del modelo en espiral (Figura 2.9).

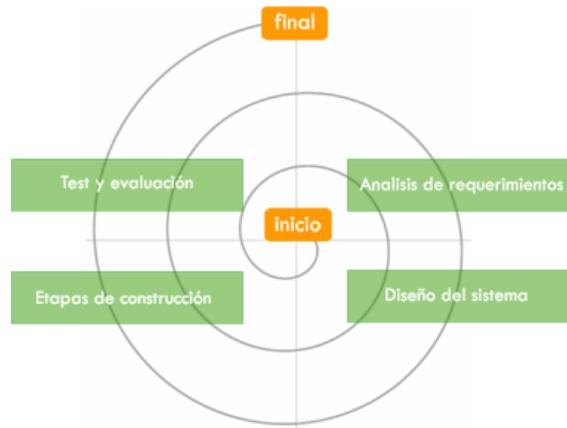


Figura 2.9: Metodología en espiral

### Ventajas

- Permite el desarrollo de proyectos en donde los objetivos finales están perfectamente definidos pero todos los detalles no pueden ser completamente establecidos al principio.
- Permite la especialización de los equipos de trabajo.
- Apela a una gestión de proyecto ordenada.
- Facilita la distribución de recursos de desarrollo.
- Permite conseguir funcionalidad en etapas tempranas.

### Desventajas

- Resulta difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.
- Debido a su elevada complejidad no se aconseja utilizarlo en pequeños sistemas.
- Genera mucho tiempo en el desarrollo del sistema.
- Modelo costoso.
- Requiere experiencia en la identificación de riesgos.

### 2.10.2. Metodología basada en prototipos

La metodología basada en prototipos, pertenece a los modelos de desarrollo evolutivo. Se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el producto entregable.

La construcción de prototipos se puede utilizar como un modelo del proceso independiente, se emplea más comúnmente como una técnica susceptible de implementarse dentro del contexto de cualquiera de los modelos del proceso de desarrollo de software.

#### **Etapas**

1. Recolección y refinamiento de requisitos
2. Modelado, diseño rápido.
3. Construcción del Prototipo.
4. Entrega y evaluación del cliente.
5. Refinamiento.
6. Entrega del desarrollo final.

En la siguiente imagen se muestra un diagrama con el ciclo de vida del modelo basado en prototipos (Figura 2.10).



Figura 2.10: Metodología basada en prototipos

### Ventajas

- Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.
- Se puede reutilizar el código.

### Desventajas

- El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final. A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte de los casos a reconstruirlo una vez que el prototipo ha cumplido su función. Es frecuente que el usuario se muestre reacción a ello y pida que sobre ese prototipo se construya

el sistema final, lo que lo convertiría en un prototipo evolutivo, pero partiendo de un estado poco recomendado.

- En aras de desarrollar rápidamente el prototipo, el desarrollador suele tomar algunas decisiones de implementación poco convenientes (por ejemplo, elegir un lenguaje de programación incorrecto porque proporciona un desarrollo más rápido). Con el paso del tiempo, el desarrollador puede olvidarse de la razón que le llevó a tomar tales decisiones, con lo que se corre el riesgo de que dichas elecciones pasen a formar parte del sistema final.

## 3. Metodología de desarrollo.

---

Lo primero antes de comenzar con un proyecto de desarrollo de software, es planificar metodológicamente el trabajo. Esto permitirá estructurar, planificar y controlar el proceso de desarrollo de software.

Según Sommerville (2005), el software no son solo programas de computadora, sino que también abarca todos aquellos documentos asociados a la configuración de datos que se necesitan para hacer que estos programas operen de manera adecuada. Para el diseño y desarrollo de proyectos de software se aplican metodologías, modelos y técnicas que permiten resolver los problemas.

Con el fin de obtener un producto final de calidad y que satisfaga los objetivos del proyecto de forma íntegra, se hace necesario adaptar una metodología de desarrollo de software, para que sea compatible con el entorno de trabajo requerido.

### 3.1. Puntos importantes para la elección.

Dadas las características del proyecto, hay ciertos puntos importantes a tomar en cuenta en la elección de la metodología a utilizar para el desarrollo del software. Los puntos más importantes son:

- El trabajo realizado tendrá como propósito satisfacer las competencias para el desarrollo del proyecto de titulación de la carrera de Ingeniería Civil en Computación.
- El desarrollo no estará a cargo de un equipo, sino que de un solo desarrollador.

- El software no busca satisfacer las necesidades de un usuario en específico, sino que busca resolver un problema más general, el cual involucra a diversos usuarios distintos.
- Es necesario tener claros los requisitos generales del sistema, antes de iniciar con el proyecto.
- El software tiene el carácter de prototipo, por lo que tiene que ser altamente modular y escalable.
- Es necesario investigar y adquirir nuevos conocimientos durante el proceso de desarrollo del proyecto.

### 3.2. Metodología de desarrollo de software.

Considerando y analizando los puntos señalados para la elección de una metodología de trabajo, se ha optado por tomar en cuenta las características de tres metodologías en conjunto y adaptarlas para satisfacer las necesidades de este proyecto. Las metodologías en las cuales se basa el marco de trabajo son:

**Metodología cascada:** Se ha optado por la utilización de esta metodología, ya que el trabajo, al ser parte del proyecto de titulación, se requiere generar gran cantidad de documentación. Por eso es que se utilizará como base un enfoque más tradicional.

**Metodología espiral:** Al carecer en un comienzo de algunos conocimientos necesario para el desarrollo del proyecto y como tiene un carácter de prototipo. El proyecto es muy susceptible a cambios durante el periodo de diseño y desarrollo. Es por eso que aunque tenga un enfoque de fondo más tradicional, es necesario que también el proyecto cuente con cierta flexibilidad para enfrentar imprevistos y poder rediseñar algunas soluciones.

**Metodología basada en prototipos:** Al ser un proyecto susceptible a cambios, este marco de trabajo entrega al proyecto una buena forma de ir apreciando los resultados parciales del proyecto. Estos resultados (versiones de prototipo) permiten generar una buena instancia para apreciar y analizar de forma temprana los riesgos que vayan surgiendo a medida que avanza el proyecto.

### 3.3. Descripción de la metodología.

Al unir y adaptar al proyecto estas tres metodologías mencionadas, se tiene un modelo más acorde con las características del proyecto. El nuevo modelo se puede dividir en 4 fases básicas de proyecto (basadas en una metodología en cascada), la cual corresponde a las principales fases del ciclo de vida de un proyecto de desarrollo de software:

1. Análisis.
2. Diseño.
3. Implementación.
4. Evaluación.

Estas fases siguen un esquema tradicional y lineal, en las cuales se generará la documentación necesaria para desarrollar un informe sobre el trabajo realizado en este proyecto.

A su vez, el proyecto requiere de cierta flexibilidad al momento de implementar los requisitos y la estructura resultante del diseño. Es por esto que en la fase de *Implementación*, se aplicarán técnicas de las metodologías de desarrollo en espiral y basada en prototipos. Estas metodologías pretenden darle un carácter más ágil al período de desarrollo. De esta manera nos permitirá enfrentar de mejor forma contratiempos fortuitos que vayan surgiendo durante la implementación.

### 3.4. Fases del ciclo de vida del proyecto.

A continuación se presentarán y describirán las fases y subfases del ciclo de vida del proyecto, los cuales conforman el marco de trabajo o metodología de desarrollo a seguir.

#### 3.4.1. Análisis.

El análisis consisten principalmente en estudiar la mejor forma de cumplir con los objetivos del proyecto. Esto permite establecer las funcionalidades que deberá poseer el sistema. Esta fase tiene un carácter tradicional, en la que se definirán requisitos

de funcionalidades puntuales que deba tener el sistema, obteniendo como resultado la especificación y definición de los requisitos del sistema, tanto los funcionales como los no funcionales.

El proceso de análisis y especificación de los requisitos, debe ser documentado de manera formal y aunque en etapas posteriores (fase de implementación) es posible que cambien algunos requisitos, no deberían ser cambios muy significativos.

#### **3.4.2. Diseño.**

Una vez concluida la fase de análisis, viene la fase de diseño. Esta etapa consiste en que a partir del modelo de análisis, se definen las estructuras de datos, los módulos en los que se descompone el sistema, la interfaz de usuario y cómo se relacionan entre sí estos componentes. Como resultado de la fase de diseño se obtiene la estructura detallada que tendrá el sistema.

Al igual que la fase de análisis, esta fase de diseño sigue un enfoque tradicional, en la que, aunque se permite modificaciones en etapas futuras, se espera que no tenga mayores variaciones durante el resto del ciclo de vida del proyecto.

#### **3.4.3. Implementación.**

La fase de implementación es donde se genera el código para la construcción del sistema. Esta etapa comienza una vez finalizada la fase de diseño. La implementación se basa en lo establecido en las fases anteriores de análisis y diseño, con lo cual se genera código ejecutable y funcional. La fase de implementación va dando forma y genera como resultado el prototipo final del sistema.

Esta fase difiere bastante del enfoque tradicional que poseen las otras fases principales del modelo. Dadas las características del proyecto, es necesario que este proceso sea más dinámico y flexible, permitiendo analizar de mejor forma la evolución del sistema. Esto nos permite apreciar de forma más clara los riesgos que vayan surgiendo, facilitándonos así el tomar las decisiones correspondientes al proceso de implementación.

Para el trabajo que se realice en esta fase, se seguirá una metodología más ágil que en las fases anteriores. La documentación de esta fase no irá tan enfocada en el proceso, sino que irá más dirigido a exponer los resultados de la implementación. Las metodologías que mejor satisfacen las necesidades de esta etapa, son las metodologías

en espiral y basada en prototipos. De esta forma obtenemos un modelo iterativo e incremental, el cual permite la generación de productos concretos (prototipos) al final de cada iteración de la espiral.

La fase de implementación está integrada en una escala en espiral que tiene su centro y enfoque principal en la creación del prototipo. Se generan prototipos, los cuales van evolucionando con cada iteración. Las iteraciones se realizarán hasta llegar a un prototipo final que satisfaga la totalidad de los requerimientos definidos. Cada iteración se divide en cuatro subfases.

### **Análisis de riesgo.**

Durante esta etapa, se estudiará el resultado de la iteración previa (de existir) y se examina detalladamente los requerimientos que cada objetivo conlleva para la nueva iteración. Es decir, se establecen todos los detalles funcionales que se implementarán en esta iteración. El fin de esta fase es disminuir al máximo los posibles conflictos que se puedan generar entre requisitos o causas externas. De ser necesario, con el fin de solucionar problemas que surjan durante el análisis, es posible hacer cambios en los requisitos o en el modelo establecidos en las etapas principales.

### **Planificación del prototipo.**

En esta etapa se planea de qué manera se implementarán los requisitos analizados en la etapa anterior y sus interacciones con el resto del sistema. Esto conlleva seleccionar las herramientas que se utilizarán, para satisfacer de la mejor forma posible las funcionalidades planteadas para implementarlas en la iteración vigente.

### **Construcción del prototipo.**

La etapa de construcción comprende básicamente la codificación y test de unidades. Esta etapa es un trabajo de programación pura y se genera el prototipo funcional que implica la iteración.

### **Pruebas del prototipo.**

En esta etapa se realiza un test del prototipo resultante de la iteración, así como su evaluación frente al estudio de requerimientos.

#### 3.4.4. Evaluación.

Una vez concluida todas las iteraciones de la fase de implementación y se ha obtenido un sistema que satisfaga todos los requisitos planteados. Se dará inicio a la fase de evaluación, donde se validará el sistema resultante de todas las etapas anteriores. Finalmente, concluirá a qué nivel se cumplieron los objetivos propuestos para este proyecto.

#### 3.5. Diagrama de la metodología a utilizar.

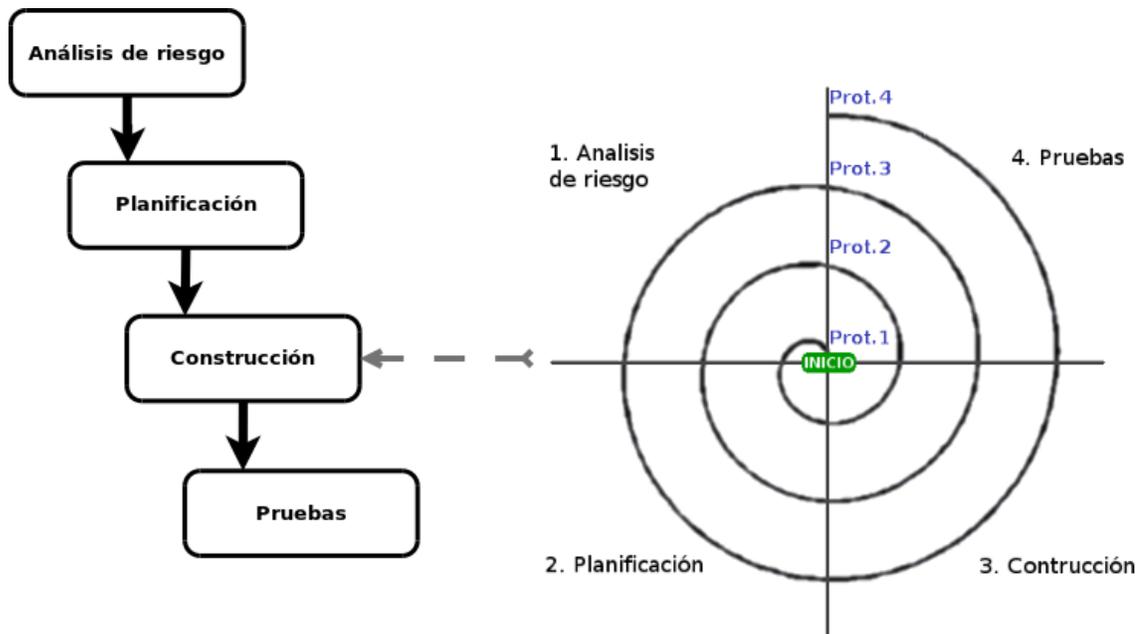


Figura 3.1: Metodología de desarrollo de software

## 4. Análisis

---

La etapa de análisis, es la primera etapa definida para la implementación del proyecto. Esta comprende principalmente al proceso de la definición de los requisitos que tendrá el sistema.

Un requisito se podría definir como:

- Condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo (IEEE).
- Condición o capacidad que debe exhibir o poseer un sistema para satisfacer un contrato, estándar, especificación, u otra documentación formalmente impuesta (IEEE).
- Una condición o capacidad que debe ser conformada por el sistema (RUP).
- Algo que el sistema debe hacer o una cualidad que el sistema debe poseer (Robertson - Robertson).

### 4.1. Especificación de los requisitos

Para la especificación de requisitos se utilizarán tres criterios principales, los cuales son:

1. La experiencia personal, obtenida tanto como desarrollador y como usuario de aplicaciones web y móviles, redes sociales y georeferenciación.
2. Investigación sobre aplicaciones web y móviles, redes sociales y georeferenciación.

3. Análisis de herramientas existentes en la actualidad, las cuales sean acordes con los objetivos del proyecto.

#### 4.1.1. Estado del arte

Con el afán de crear una herramienta útil y novedosa, para la especificación de los requisitos, se han analizado sistemas SIG y redes sociales vinculadas con mapas y georreferenciación existentes en la actualidad. Con este análisis se pretende conocer los puntos fuertes y débiles de cada una de estas herramientas, para tener en cuenta al momento de implementar funciones en el sistema.

##### Facebook Places

Esta es una función de la conocida red social Facebook.

**Funcionalidades:** Facebook Places permite hacer entradas o check-ins en un sitio específico con un dispositivo móvil.

##### Waze

Waze es una red social SIG que se especializa en las condiciones del tránsito y la optimización de rutas.

**Funcionalidades:** Waze funciona a base de puntos coordenados, los cuales se muestran en el mapa mediante distintos iconos. Estos iconos representan distintos eventos relacionados con el tránsito (ej: congestión vehicular, control policial, etc.). Esta información que se muestra en el mapa, es proporcionada por los mismos usuarios.

##### Foursquare

Foursquare es una red social SIG que presenta lugares de interés en una determinada zona.

**Funcionalidades:** Foursquare permite marcar puntos «check-in» en lugares específicos donde uno se encuentra, e ir ganando puntos por «descubrir» nuevos lugares. De esta manera Foursquare muestra lugares de interés de distintas zonas del

mundo, basado en los «check-in» y comentarios del lugar. Además Foursquare separa los lugares por categorías (ej: compras, comida, etc.), facilitando la tarea de búsqueda de un lugar para una determinada actividad.

### Google My Maps

Google My Maps, es una red social, basada en google maps, lanzada recientemente por la conocida compañía Google.

**Funcionalidades** Google My Maps crea mapas personalizados para compartirlos y publicarlos en línea. Puede crear mapas en los que se muestren diferentes tipos de información.

Con My Maps, podrás hacer lo siguiente:

- Crear un mapa.
- Dibujar líneas, formas o marcadores en el mapa.
- Importar datos geográficos específicos, como direcciones, nombres de lugares o coordenadas de latitud y longitud.
- Organizar el mapa con capas para ocultar o mostrar distintos tipos de contenido.
- Aplicar estilos al contenido con diferentes colores, iconos de marcadores y anchos de línea.

#### 4.1.2. Conclusiones del análisis

De todas las aplicaciones analizadas, Google My Maps es la que más se asemeja y satisface los objetivos del proyecto. Es por esto que será una guía en la cual basarse a la hora de especificar los requisitos y también diseñar y validar una interfaz gráfica funcional.

Aunque Google My Maps satisfaga de cierta forma los objetivos del proyecto, la aplicación de Google tiene otro enfoque, por lo que aún carece de funcionalidades para satisfacer de forma más íntegra estos objetivos. Es por esto que aunque será de

gran aporte en la especificación de requisitos e interfaz gráfica, se necesitaran tomar en cuenta más funcionalidades al momento de implementar este proyecto.

A continuación se señalaran las características de las aplicaciones analizadas, las cuales proporcionarán una base para la especificación de los requisitos de este proyecto:

**Facebook Places:** La única característica que presenta que puede aportar al proyecto es el la función de geolocalización.

**Waze:** La característica interesante que presenta, es la manera cómo los usuarios generan los datos y la información. Además de que hace esta presentación de datos a través de puntos coordenados en el mapa.

**Foursquare:** Al igual que Waze, la información es generada por los usuarios y funciona a través de puntos coordenados en un mapa. Pero además cuenta con un buen sistema de capas, para filtrar de mejor manera la información desplegada. También permite hacer comentarios sobre los distintos lugares.

**Google My Maps:** Como se mencionó es la herramienta que presenta más funcionalidades acorde con los objetivos del proyecto. Además de aportar como guía para la especificación de los requisitos, Google My Maps es una importante pauta para la interfaz y usabilidad del sistema que se va a desarrollar.

Las funcionalidades a considerar de Google My Maps son:

- A grandes rasgos permite crear, editar y visualizar mapas con información personalizada.
- Presenta una completa estructura para la generación de mapas, la cual está compuesta por Mapa  $\Rightarrow$  Capas  $\Rightarrow$  Elemento.
- Permite la creación de 3 tipos de elementos distintos: punto, línea y polígono.
- Permite modificar el estilo de los elementos creados.
- Puede publicar los mapas construidos en redes sociales.

### Funcionalidades extra

- El sistema debe permitir la generación de información de la forma más completa posible, por lo que debe trabajar con un generador HTML lo que permite ingresar texto con formato, imágenes, vídeos, tablas, estadísticas y cualquier cosa que se pueda representar mediante HTML.
- El sistema debe tener más herramientas para crear y modificar los elementos de forma gráfica en el mapa.

## 4.2. Requisitos

### 4.2.1. Funcionales

**Crear usuario** Se podrá registrar una cuenta de usuario en el sistema. Para crear una cuenta se solicitará:

- Correo electrónico.
- Contraseña.
- Confirmación de contraseña.
- Nombre (opcional).

**Editar usuario** Un usuario registrado podrá modificar o agregar datos a su cuenta. Los datos que se podrán agregar/modificar serán:

- Correo electrónico.
- Contraseña.
- Confirmación de contraseña.
- Nombre (opcional).

**Eliminar usuario** Un usuario registrado, se podrá dar de baja del sistema al eliminar su cuenta de usuario.

**Iniciar sesión con cuentas de redes sociales** El usuario podrá iniciar sesión si es que está registrado en el sistema o mediante su cuenta de facebook.

**Cerrar sesión** Un usuario logueado en el sistema, podrá cerrar su sesión para que su información no sea accesible por otras personas.

**Listar mapas** Se desplegará una lista con los nombres de todos los mapas que existen en el sistema.

**Visualizar servidor de mapas** Se podrá conectar con un servidor de mapas, el que servirá como fondo para guiarse visualmente en el mapa y poder crear capas y elementos en sobre el. El servidor de mapas por defecto que utilizará el sistema, será OpenStreetMap.

**Controles del mapa** Se podrá navegar a través del mapa de forma sencilla, mediante los siguientes controles:

- desplazamiento: se podrá desplazar a través del mapa presionando y arrastrando el puntero o el dedo por el interior de este.
- zoom: se podrá acercar y alejar el mapa, mediante el cambio de proximidad de dos dedos presionados en el mapa o moviendo el scroll hacia arriba o abajo.
- escala: se mostrará la escala del mapa, tanto en sistema internacional, como el sistema anglosajón de unidades.
- coordenadas: se mostrarán las coordenadas espaciales del puntero o dedo, dentro del mapa.

**Crear mapa** Se podrá crear una nueva instancia de mapa en el sistema. Los parámetros para crear un mapa seran:

- nombre del mapa.
- código de edición (opcional).

**Guardar mapa** Se podrá guardar un mapa en el sistema, se guardarán todos los datos ingresados (a nivel de capas y elementos), así como también la vista actual del mapa (coordenada central y nivel de zoom).

**Cargar mapa** Se podrá acceder un mapa anteriormente guardado en el sistema, tanto para su edición como para la visualización de los datos que tenga ingresados.

**Eliminar mapa** Un usuario que pueda acceder al mapa, podrá eliminarlo. Esta acción eliminará junto con el mapa, todos los datos que se encuentren contenidos en este.

**Asignar viewport inicial de mapa** Se podrá asignar el centro y zoom inicial de un mapa. Esta función se realizará al momento de guardar un mapa.

**Editar parametros del mapa** Se podrá cambiar los distintos parametros que tiene un mapa, los parámetros que se pueden editar son:

- nombre del mapa.
- código de edición (opcional).

**Código de ingreso para editar mapas** Se podrá establecer un código alfanumérico al mapa, para restringir el acceso a la edición de este. De esta manera, solo podrán editar un mapa los usuarios que sepan el código.

**Crear capa** Se podrá crear una nueva instancia de capa en un mapa. Los parametros para crear una capa serán:

- nombre de la capa.
- color de la capa.
- su estado de visibilidad inicial (visible/oculto).

**Editar parámetros de la capa** Se podrá cambiar los distintos parámetros que tiene una capa, los parámetros que se pueden editar son:

- nombre de la capa.
- color de la capa (elementos que se dibujen en la capa).
- estado de visibilidad inicial (visible/oculto).

**Importar capa** Se podrá importar una capa de otro mapa al que se está editando actualmente. (Esta función aunque implementada, no está accesible mediante la interfaz de usuario).

**Listar capas** Se desplegará una lista con los nombres de las distintas capas que existen en el actual mapa.

**Seleccionar capa** Se podrá seleccionar una capa del mapa, para poder trabajar en ella.

**Eliminar capa** Se podrá eliminar una capa seleccionada.

**Cambiar orden de capas** Se podrá cambiar el orden de las capas en el eje Z, de esta manera se podrá controlar la visualización de una capa sobre otra (esta función aunque implementada, no está accesible mediante la interfaz de usuario).

**Opción ver/ocultar capa** Se podrá controlar la opción de ver/ocultar una capa, para de esta manera tener una mejor visualización de las demás capas.

**Crear elemento** Se podrá crear un nuevo elemento dentro de una capa seleccionada. Un elemento consiste tanto en su parte gráfica y una parte lógica (atributos). Gráficamente se podrán crear tres clases de elementos distintos:

- Polígonos: se podrá construir cualquier polígono de forma libre.
- Líneas: se podrá construir una línea.

- Puntos: se podrá construir un punto que a diferencia de los otros dos elementos, tiene la particularidad de conservar un tamaño constante, independiente del número o nivel de zoom.

Los atributos lógicos para crear un elemento se generarán automáticamente al crear su parte gráfica, obteniendo un nombre genérico el cual podrá ser editado posteriormente. Los atributos de un elemento son:

- Nombre del elemento.
- Información asignada.

**Listar elementos** Se desplegará una lista con los nombres de los distintos elementos que existen en la capa seleccionada.

**Seleccionar elemento de la lista** Se podrá seleccionar un nombre de la lista de elementos, tanto para visualizarlo en el mapa, como para editar sus parámetros.

**Seleccionar geometría de elementos del mapa** Se podrá seleccionar uno o múltiples elementos visibles en el mapa, para poder editarlo gráficamente.

**Editar parámetros del elemento** Se podrán cambiar los distintos parámetros que tiene un elemento, los parámetros que se pueden editar son:

- nombre del elemento.
- información asignada.

**Editar gráficos del elemento** Se podrá modificar gráficamente un elemento dibujado en el mapa. Las opciones de modificación serán:

- modificar la geometría: permite cambiar la posición y la cantidad de los vértices.
- dibuja agujero: permite cortar un agujero (en forma de polígono), al interior de un polígono dibujado en el mapa.
- desplazar geometría: permite mover o desplazar el elemento por el mapa.

**Ingresar información a un elemento** Se podrá ingresar o modificar datos en un elemento creado. Los tipos de datos que se podrá ingresar/modificar será:

- texto con formato.
- caracteres especiales.
- links.
- imágenes.
- video.
- tablas.

**Eliminar elemento** Se podrá eliminar un elemento. Esta acción eliminará junto con el elemento, todos los datos que se encuentren contenidos en este.

**Compartir en redes sociales** Se podrá compartir un mapa en Facebook, permitiendo ingresar al mapa haciendo click en la publicación de Facebook.

**Comentarios** Los usuarios que abran un mapa con el visualizador de mapas, podrán dejar un comentario acerca de ese mapa.

**Localización** Los usuarios que abran un mapa con el visualizador de mapas, podrán acceder a su ubicación actual en el mapa. Aunque no es imprescindible, esta función es más exacta si el dispositivo en el cual se ejecuta el sistema, cuenta con un GPS incorporado.

#### 4.2.2. No Funcionales

**Fácil de usar** Para facilitar el uso, se disminuirá al máximo la sobrecarga de información y opciones al usuario. De esta manera el usuario podrá acceder y realizar las tareas de forma sencilla sin tener que preocuparse de opciones complejas difíciles de utilizar. El usuario no necesitará grandes conocimientos, ya que la aplicación será intuitiva e interactiva ya que tendrá una interacción similar a un editor de imágenes simple (como Paint).

**Flexible** El sistema será lo suficientemente flexible como para poder crear distintos tipos de mapas sin problema para diversos usos.

**Seguro** La seguridad será gestionada mediante cuentas de usuario y privilegios para los mapas. Además se usará una arquitectura modelo-vista-controlador, para manejar de mejor forma la información delicada en el servidor. Gracias a esto, antes de que se cargue el controlador se filtran los datos enviados para que estos puedan resultar fiables.

**Rápido** La creación de mapas se podrá hacer de forma rápida ya que contendrá pocas opciones para el usuario, con las que podrá realizar todo lo que necesite. Esto disminuirá el tiempo que emplea el usuario para realizar cualquier acción en el sistema. Además se disminuirá al máximo la interacción con el servidor para agilizar el tiempo de respuesta de las consultas, esto último se hará dándole más carga al cliente, aprovechando el alto nivel de procesamiento con el que vienen los dispositivos móviles hoy en día.

## 5. Diseño

---

La etapa de diseño, es la segunda etapa definida para la implementación del proyecto.

“El milagro más común de la ingeniería de software es la transición del análisis al diseño y del diseño al código”

(Richard Due).

El diseño del software se encuentra en el núcleo técnico del desarrollo de software. Una vez que se analizan y especifican los requisitos, el diseño del software es la última acción de la ingeniería realizada dentro de la actividad del modelado, la cual establece una plataforma para la construcción (generación de código y prueba).

### 5.1. Aspectos importantes a considerar

Para el diseño de este proyecto, es fundamental tomar en cuenta ciertos aspectos esenciales:

**El sistema será un prototipo:** Esta primera instancia del proyecto, se trata de la creación de un prototipo, por lo que debe ser diseñado para que sea altamente escalable y adaptarse fácilmente a los cambios.

**El sistema debe ser multiplataforma:** El sistema necesita ser funcional en una gran cantidad de dispositivos distintos. A diferencia de antes que existía poca variedad de gadget, Sistemas Operativos y plataformas distintas, cada vez aumentan más

las alternativas. Es por esto que no se puede atar a alguna plataforma en específico, si no que la aplicación debe poder correr en la mayor cantidad de plataformas distintas.

**El sistema debe ser efectivo en la transmisión de información:** Una de las metas fundamentales del proyecto, es generar una herramienta para transmitir información de forma efectiva y novedosa. Es por esto que es muy importante la forma como lleva a cabo esta función. Además una de las particularidades de MyGIS es que será una red social, por lo que debe permitir un fácil acceso a sus funcionalidades a todo tipo de usuario y entregas un buen método para difundir la información deseada.

## 5.2. Diseño arquitectónico

La arquitectura del software es el diseño de más alto nivel de la estructura de un sistema. Esta consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco de trabajo.

La arquitectura del software se selecciona y diseña con base en objetivos (requerimientos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información.

Aunque este proyecto tenga requisitos definidos, es un prototipo para el estudio de un mercado. El que sea un prototipo, lo hace muy susceptible a cambios de requisitos a distintos niveles, es por esto que la flexibilidad y modularidad son aspectos importantes al momento de seleccionar una arquitectura. Por otro lado tiene la cualidad de ser una red social, por lo que también adquiere gran relevancia el tema de la conectividad, el acceso y la transmisión de los datos.

### 5.2.1. Elección de la arquitectura

Después de analizar las cualidades de las distintas arquitecturas de software existentes, y contrastándolas con los objetivos y requerimientos de este proyecto. Se ha

decidido que implementar una única arquitectura no satisficará de buena manera los objetivos de este proyecto, es por esto que se ha optado por usar la mezcla de dos modelos arquitectónicos muy utilizados en la actualidad, los cuales a su vez son la evolución y mezcla de otros populares modelos arquitectónicos.

Los modelos arquitectónicos elegidos para la construcción de MyGIS son:

1. Arquitectura Modelo Vista Controlador (MVC).
2. Arquitectura Orientada a Servicios.

### **5.2.2. Diagrama de la arquitectura**

## **5.3. Diseño del modelo lógico**

Una de las partes fundamentales de MVC es el modelo de lógica de negocio. Para el diseño del modelo es importante identificar las principales clases que interactúan y participan en el sistema.

Las clases que se logran identificar son:

- Mapa
- Capa
- Elemento
- Usuario
- Comentario

### **5.3.1. Diseño Base de Datos**

La Base de Datos es de los aspectos fundamentales en el desarrollo de un Sistema de Información como lo será MyGIS, por lo que es de suma importancia una buena elección de esta.

## Elección de Base de Datos

Dado que no se cuentan con los recursos para la adquisición de licencias pagadas de motores de Bases de Datos, además de tomar en cuenta que el producto final de este proyecto será un prototipo y no una versión final, ni comercial de MyGIS. Las alternativas de elección de una Base de Datos, se limitan a opciones que sean libres.

Tomando en consideración la limitante anteriormente señalada, se estudiará la implementación de uno de los dos motores de Base de Datos libres más populares:

**PostgreSQL:** Se caracteriza por enfocarse tradicionalmente en la fiabilidad, integridad de datos y características integradas enfocadas al desarrollador. Además una característica interesante para el desarrollo del proyectos es un mejor soporte de coordenadas espaciales.

**MySQL:** Por otra parte, se ha enfocado tradicionalmente en aplicaciones web de lectura mayormente, usualmente escritas en PHP, donde la principal preocupación es la optimización de consultas sencillas.

Analizando las principales características de estos dos motores de Base de Datos libres. Se ha decidido implementar MySQL como Base de Datos en MyGIS. Esta elección se debe a tres razones principales, las cuales se nombraran a continuación:

- Se utilizarán servicios web, por lo que el manejo de los datos será procesado a nivel de servicios y no de Base de Datos. Utilizando la Base de Datos solo para el almacenamiento y recuperación de los datos a través de consultas sencillas. Esta función es mejor soportada por MySQL.
- Dado que una de las características principales de MyGIS es su conectividad y el traspaso de información via web. MySQL ofrece un mejor soporte para estas funciones.
- Se tiene mayor experiencia con el uso de MySQL como motor de Base de Datos, por lo que la curva de aprendizaje será mucho menor que al utilizar PostgreSQL.

### 5.3.2. Diagrama de clases

A continuación se muestra un diagrama con las clases anteriormente señaladas, además de sus atributos y relaciones. A partir de estas clases se construirán las tablas que darán forma a la Base de Datos:

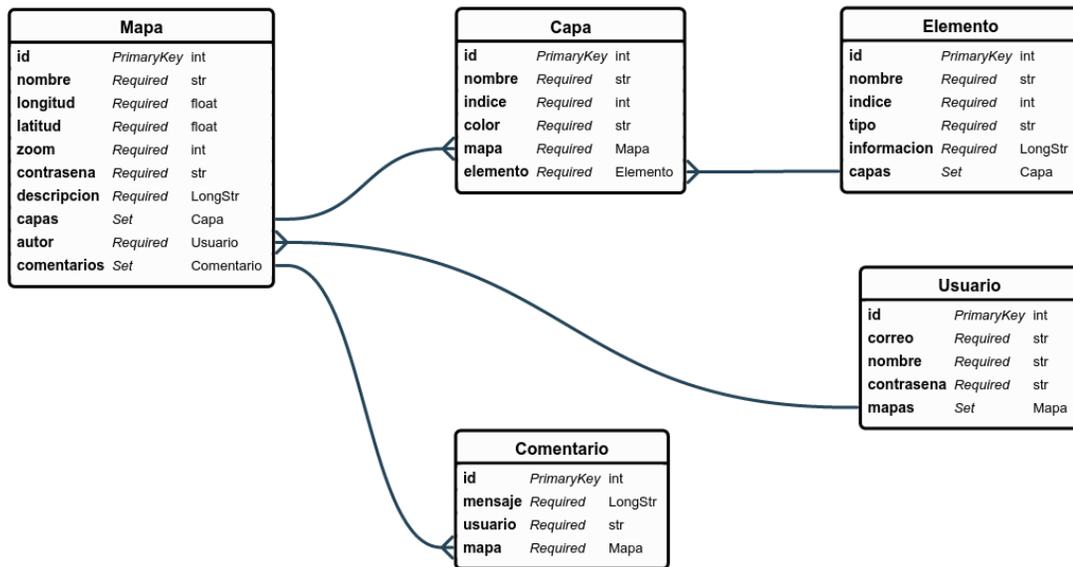


Figura 5.1: Diagrama de clases de MyGIS

## 5.4. Diseño vista

Varios de los requisitos de MyGIS, tienen que ver con la vista y como se muestra esta al usuario. Dos características importantes que hay que tomar en cuenta a la hora de diseñar la vista son:

- MyGIS requiere ser una herramienta fácil de utilizar, por lo que es importante que tenga una interfaz amigable con el usuario.
- Otra característica importante de MyGIS es que sea visible en la mayor cantidad de dispositivos posibles. Esto incluye dispositivos móviles con pantallas

reducidas, los cuales son los que entregan mayor dificultad a la hora de diseñar una vista adecuada.

Para satisfacer los puntos anteriormente mencionados, se tomó la decisión de dividir el sistema en dos vistas distintas. Esta decisión se tomó considerando que lo siguiente:

- La visualización de los mapas para la obtención de los datos que contiene (lectura de datos), se hará con mucha mayor frecuencia que la creación de los mapas y sus datos (escritura de datos).
- Las opciones y conocimientos requeridas para la creación de mapas son mucho mayores y de mayor complejidad que las requeridas para la visualización de un mapa.

Es por esto que con la intención de simplificar la vista a la hora de visualizar un mapa, se desido en entregas vistas distintas para la creación (y edición) y la visualización de un mapa.

#### **5.4.1. Editor de Mapas**

Será un vista con más funcionalidades y opciones, la cual entregará todas las herramientas necesarias para creación, administración y difusión de un mapa y todas las cosas que esto implica.

#### **5.4.2. Visualizador de Mapas**

Será una vista mucho más sencilla de utilizar que la vista de edición de mapas, ya que tendrá muchas menos opciones y se enfocara más en la navegación del mapa y la entrega de datos.

### **5.5. Diseño Controlador**

En la Arquitectura MVC, el controlador es el encargado de comunicar la vitas con el modelo y la Base de Datos. Se podría decir que el controlador es el encargado de administrar el flujo de datos. Esta función está ligada a la plataforma de desarrollo que se utilizara, es por esto que a continuación se definirá la plataforma de desarrollo a utilizar para la implementación de MyGIS.

### 5.5.1. Elección de plataforma de desarrollo

Para la elección de la plataforma de desarrollo se considerarán los siguientes puntos:

- El sistema requiere de servicios web externos, tales como el servidor de mapas o la integración con redes sociales.
- El sistema requiere un alto grado de conectividad y traspaso de datos via web.
- El sistema requiere ser utilizado en distintos dispositivos, tanto en pcs de escritorio como en dispositivos móviles como teléfonos inteligentes. Por lo que requiere ser multiplataforma.
- La Base de Datos que se utilizará (MySQL) se enfoca en aplicaciones web.

Considerando los puntos mencionados, resulta evidente que la plataforma a utilizar será una plataforma web. Una aplicación web satisficaría los requerimientos definidos para el sistema de buena manera.

Teniendo como objetivo el desarrollo de una aplicación web. Dada la experiencia en desarrollo de este clase de aplicaciones y deseando profundizar más en dichos conocimientos, se ha decidido por montar una plataforma de desarrollo basado en HTML5, JavaScript y CSS3.

## 5.6. Diagrama estructural final

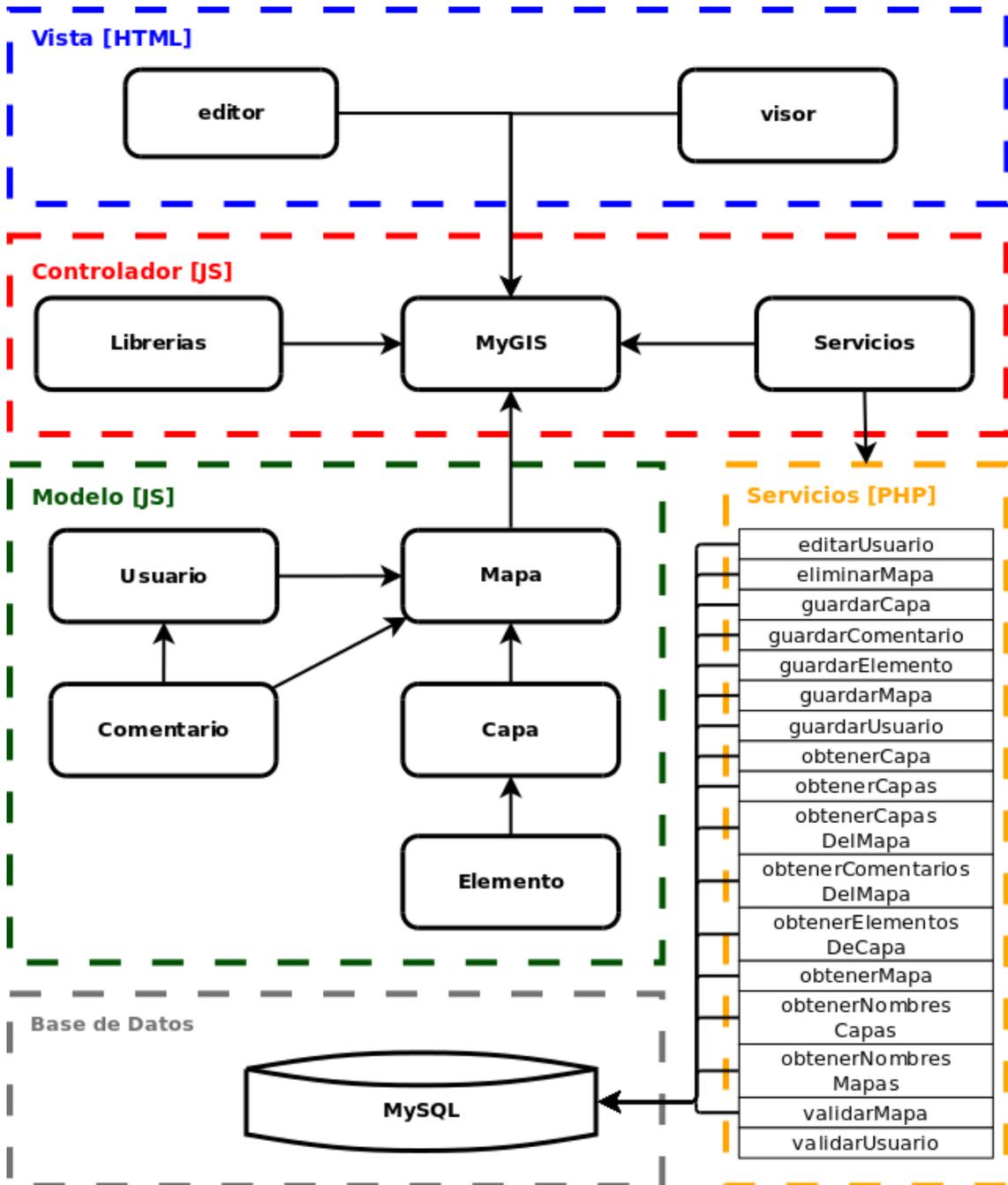


Figura 5.2: Diagrama arquitectura de MyGIS

## 6. Implementación

---

La etapa de implementación, es la tercera etapa definida para la implementación del proyecto.

En esta etapa del proyecto, toma un carácter más ágil, por lo que la documentación no es tan extensa y más que nada se basa en los productos obtenidos (prototipos).

### 6.1. Entorno de Desarrollo

En esta sección, se expondrán todos los componentes utilizados para formar el entorno de desarrollo, con el cual se construyó el sistema MyGIS. Estos componentes incluyen tanto el software como el hardware.

#### 6.1.1. Hardware utilizado en el desarrollo

Para el desarrollo se utilizó un computador personal portatil o notebook de la marca Dell. Se eligió utilizar este computador ya que es de fácil acceso y cumple con los requisitos para poder desarrollar un sistema de las características de este proyecto.

#### Características de hardware

Las características de hardware del computador utilizado son las siguientes:

**Marca y modelo:** Dell Inspiron 15R (N5110)

**Procesador:** Intel® Core i7-2630QM 2.00 GHz, 6M Cache (2.9GHz Turbo Mode)

**Memoria:** 8GB 1333MHz DDR3

**Video:** NVIDIA® GeForce® GT 525M 1GB (Quad)

**Almacenamiento:** 120GB Corsair Force 3 SSD (1.3.3)

### 6.1.2. Software utilizado en el desarrollo

#### Sistema Operativo

Para el desarrollo del sistema se utilizó mayormente Linux, mediante su distribución Ubuntu 14.04.

#### Lenguajes de programación

Debido a que se necesita un sistema multi plataforma, se decidió desarrollar el sistema como una aplicación web. Los lenguajes con los que fue construida esta aplicación web por parte del cliente son:

- HTML5.
- CSS3.
- JavaScript.

Mientras que por parte del servidor se implementaron servicios utilizando PHP, el cual se comunica con la base de datos.

#### Servidor HTTP

Como servidor HTTP se utilizó Apache

#### Intérprete HTTP (navegador web)

Durante el desarrollo, como navegador web o browser se utilizó principalmente Google Chrome. Aunque como es un sistema multiplataforma y debe poder ser leído por múltiples navegadores. Para las pruebas del sistema, se utilizaron múltiples navegadores, tanto de dispositivos móviles como de computadoras de escritorio.

### **Comunicación cliente/servidor**

Para la comunicación asincrónica entre cliente, servidor y viceversa, se utilizó AJAX.

### **Editor de código fuente**

Como editor de código fuente se utilizó Sublime Text.

### **Librerías**

Para la creación de algunos componentes de MyGIS, se utilizaron algunas librerías escritas en JavaScript. La librerías utilizadas son:

- JQuery
- JQuery Mobile
- OpenLayers
- OLE
- TinyMCE
- Hello

#### **6.1.3. Base de Datos (BD)**

Como BD, se utilizó una BD relacional, gestionada por MySQL.

## **6.2. Prototipos**

Ya que para el desarrollo de MyGIS no existía un cliente concreto, la construcción de este no contó con prototipos propiamente como indica la metodología basada en prototipos. Sin embargo, la etapa de implementación se puede dividir en 4 hitos importantes, los cuales culminaron en productos concretos y funcionales. Estos hitos, surgieron tras cada iteración de la fase de implementación. Esto quiere decir que la fase de implementación estuvo constituido por 4 iteraciones hasta su culminación.

Estas iteraciones se podrían resumir en:

1. Generación de mapas y capas de forma dinámica.
2. Generación de elementos dinámicos y asignar información a los elementos.
3. Conexión base de datos, implementación de usuarios e integración con redes sociales.
4. Interfaz gráfica y visualizador de mapas con comentarios y localización.

### **6.2.1. Primera iteración**

#### **Análisis**

Como es la primera iteración de la etapa de implementación, esta no cuenta con análisis de iteraciones previas.

#### **Planificación**

Se planificaron los pasos que se llevaron a cabo en el proceso de construcción.

#### **Construcción**

Los pasos que se siguieron en el proceso de construcción de esta iteración son los siguientes:

1. Crear la estructura de clases e implementación de la arquitectura MVC en JavaScript.
2. Crear HTML inicial (index).
3. Implementar librerías JavaScript: JQuery, JQuery Mobile, OpenLayers.
4. Generar una interfaz gráfica simple, para visualizar y controlar eventos.
5. Generar las funciones necesarias para poder crear y editar los atributos gráficos y lógicos de un mapa. Esto de forma dinámica.
6. Generar las funciones necesarias para poder crear y editar los atributos gráficos y lógicos de las capas. Esto de forma dinámica.

## Pruebas

Se ejecutaron las pruebas unitarias y de integración correspondiente a las funcionalidades implementadas en la iteración.

## Producto obtenido

Esta primera iteración, dio como resultado un producto con una interfaz gráfica muy simple. Donde los controles consisten en botones y cajas de texto sencillas, los cuales daban la posibilidad de crear y editar los atributos de un mapa con sus capas y sus respectivos atributos.

## Captura del producto

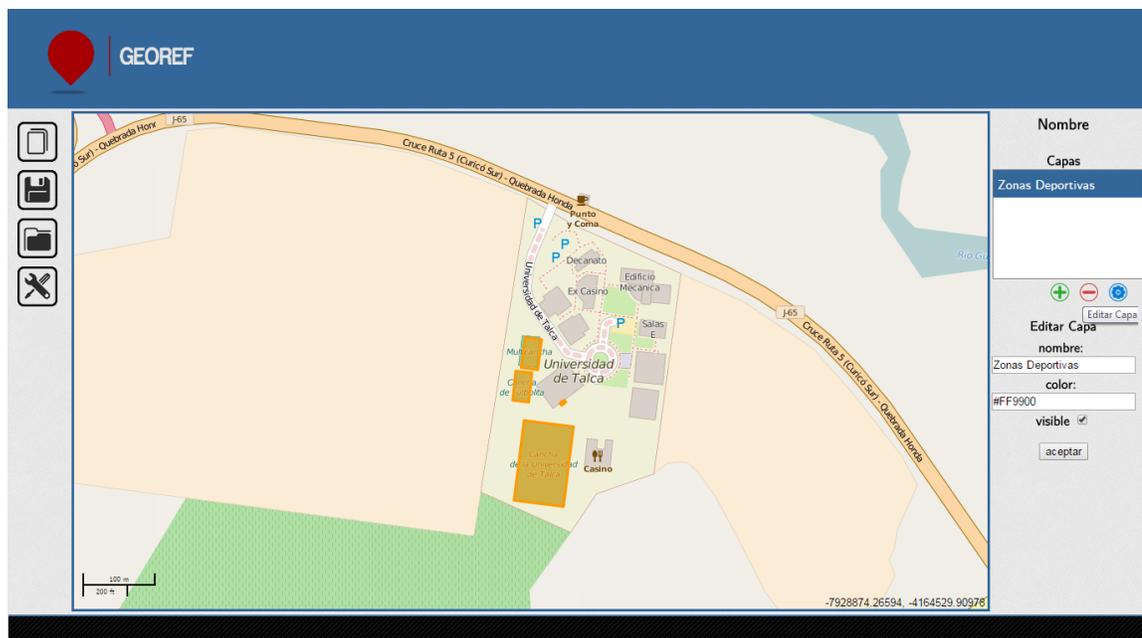


Figura 6.1: Captura del producto de la primera iteración

### 6.2.2. Segunda iteración

#### Análisis

El producto de la primera iteración fue satisfactorio. Sin embargo esta iteración tomó una cantidad importante de tiempo para su finalización. Esto debido prin-

principalmente al tiempo empleado en el aprendizaje del funcionamiento de la librería OpenLayers, ya que como se pretendía usar de forma dinámica, se requirió de una comprensión a fondo de dicha librería.

Antes de comenzar con la segunda iteración, se decidió obtener la mayor cantidad posible de documentación y ejemplos sobre OpenLayers, así como también sobre JQuery, JQuery Mobile y JavaScript. Para la segunda iteración, se decidió implementar los demás objetos del mapa.

### **Planificación**

Se planificaron los pasos que se llevaron a cabo en el proceso de construcción.

### **Construcción**

Los pasos que se siguieron en el proceso de construcción de esta iteración son los siguientes:

1. Implementar librerías JavaScript: OLE (OpenLayer Editor) y TinyMCE.
2. Generar las funciones necesarias para poder crear y editar los atributos gráficos y lógicos de los elementos. Esto de forma dinámica.
3. Generar las funciones necesarias para poder crear y editar la información contenida por los elementos.

### **Pruebas**

Se ejecutaron las pruebas unitarias y de integración correspondiente a las funcionalidades implementadas en la iteración.

### **Producto obtenido**

La segunda iteración, dio como resultado un producto que contaba con las funciones y la interfaz de usuario (aunque seguía siendo básica) necesarias para crear un mapa con todos sus componentes (mapa, capas y elementos con su información). Todo esto de forma dinámica.

## Captura del producto

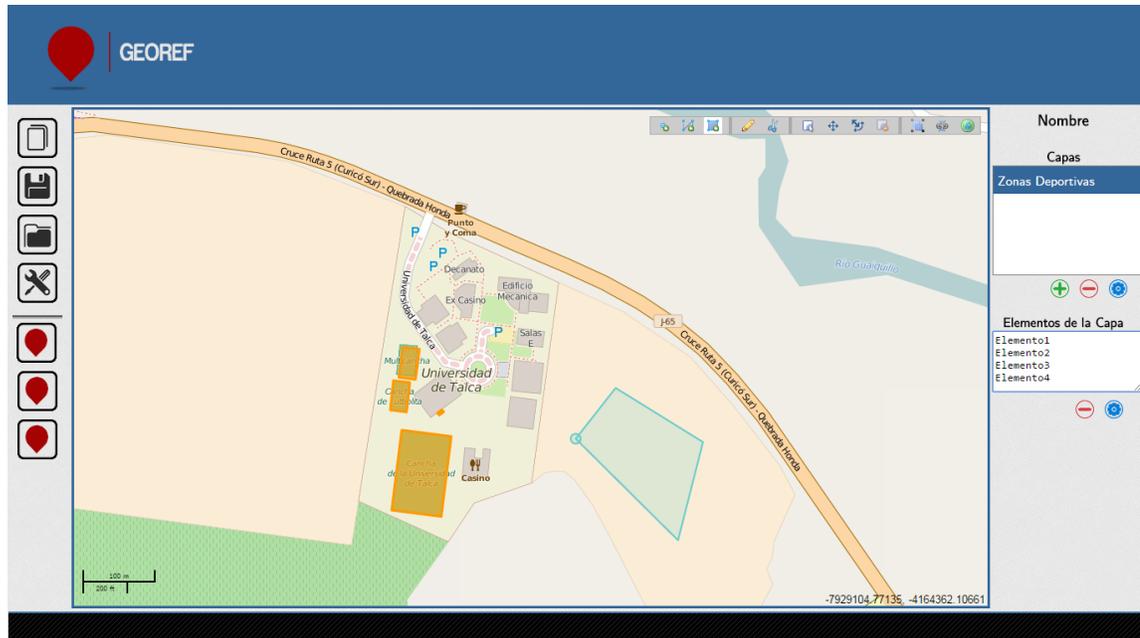


Figura 6.2: Captura del producto de la segunda iteración

### 6.2.3. Tercera iteración

#### Análisis

El producto de la segunda iteración también fue satisfactorio. Los único contra-tiempo que se presentó en la iteración anterior, fueron conflictos entre las librerías JQuery y OLE. Los conflictos que se generaron fueron:

- Alcance de nombres de variables entre las librerías JQuery y OLE.
- Conflicto con el foco del mapa entre las librerías OpenLayer y OLE.

Una vez identificados los conflictos, estos finalmente se solucionaron sin mayores problemas.

Dando que como producto resultante de la iteración anterior, se obtuvo un sistema con la posibilidad de crear mapas de forma íntegra. Para la tercera iteración se decidió implementar las funciones sociales y la persistencia de los datos (conexión con Base de Datos).

### **Planificación**

Se planificaron los pasos que se llevaron a cabo en el proceso de construcción.

### **Construcción**

Los pasos que se siguieron en el proceso de construcción de esta iteración son los siguientes:

1. Implementar la librería JavaScript: Hello.
2. Crear base de datos.
3. Generar servicios y conexión con la base de datos.
4. Generar las funciones necesarias para acceder a los servicios.
5. Generar las funciones necesarias para gestionar cuentas de usuarios.
6. Generar las funciones necesarias para ingresar con una cuenta de usuario de Facebook.
7. Generar las funciones necesarias para compartir en Facebook los mapas creados.

### **Pruebas**

Se ejecutaron las pruebas unitarias y de integración correspondiente a las funcionalidades implementadas en la iteración.

### **Producto obtenido**

La tercera iteración, dio como resultado un producto que contaba con las funciones e interfaz de usuario básicas, capaces de ingresar con una cuenta de usuario. Además de permitir crear, guardar/cargar y compartir los mapas generados.

### Captura del producto

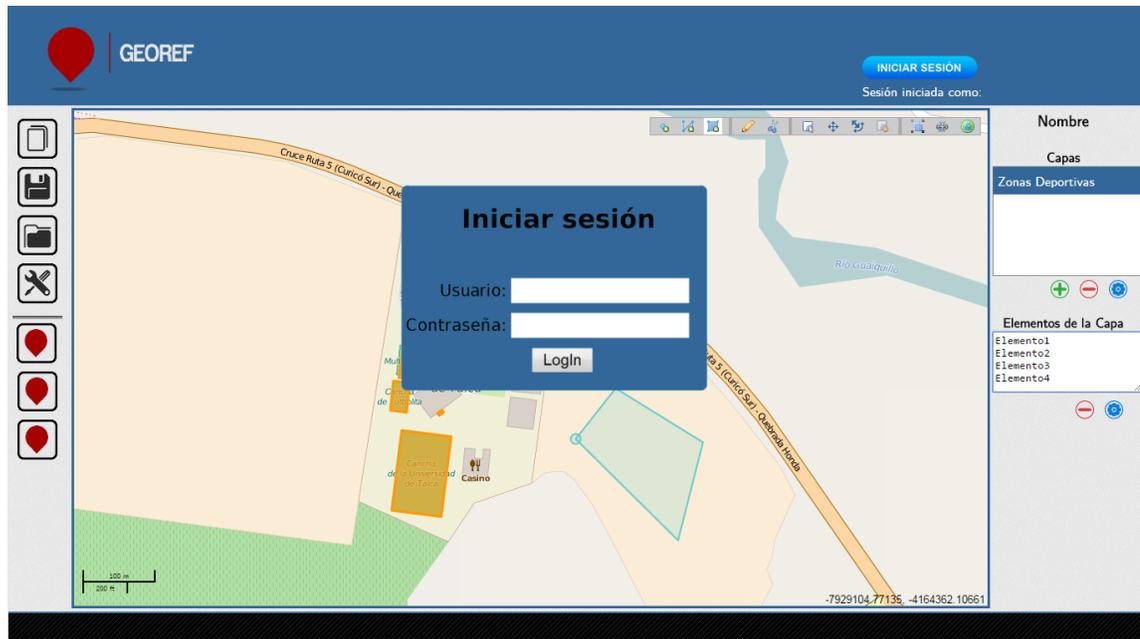


Figura 6.3: Captura del producto de la tercera iteración

#### 6.2.4. Cuarta iteración

##### Análisis

Al terminar la tercera iteración, ya se cuenta con un sistema bastante completo en cuanto a la satisfacción de los requisitos propuestos. Con este producto los requisitos funcionales mas significativos, asi como la arquitectura y la lógica del sistema están satisfechos.

Teniendo satisfecho la .obra gruesa.º las funcionalidades más significativas del sistema. Lo siguiente es enfocarse en la apariencia y usabilidad del sistema, es en esto en lo que se centrará esta cuarta y última iteración de la etapa de implementación. .

##### Planificación

Se planificaron los pasos que se llevaron a cabo en el proceso de construcción.

### **Construcción**

Los pasos que se siguieron en el proceso de construcción de esta iteración son los siguientes:

1. Creación de la vista del visor de mapas.
2. Generar las funciones necesarias para el visor de mapas.
3. Generar las funciones necesarias para crear comentarios en un mapa.
4. Rediseño de la interfaz gráfica, para que sea compatible con dispositivos móviles.

La cuarta y última iteración fue el rediseño de la interfaz gráfica, para que sea compatible con dispositivos móviles, además se separaron las vistas entre el editor y el visor de mapas.

### **Pruebas**

Se ejecutaron las pruebas unitarias y de integración correspondiente a las funcionalidades implementadas en la iteración.

### **Producto obtenido**

Culminando la cuarta y última iteración de la etapa de implementación del software, tenemos como resultado el sistema final, el cual será presentado como proyecto de título. Este producto satisface todos los requisitos propuestos de forma íntegra, por lo que se puede determinar cómo finalizado exitosamente la etapa de implementación.

Captura del producto

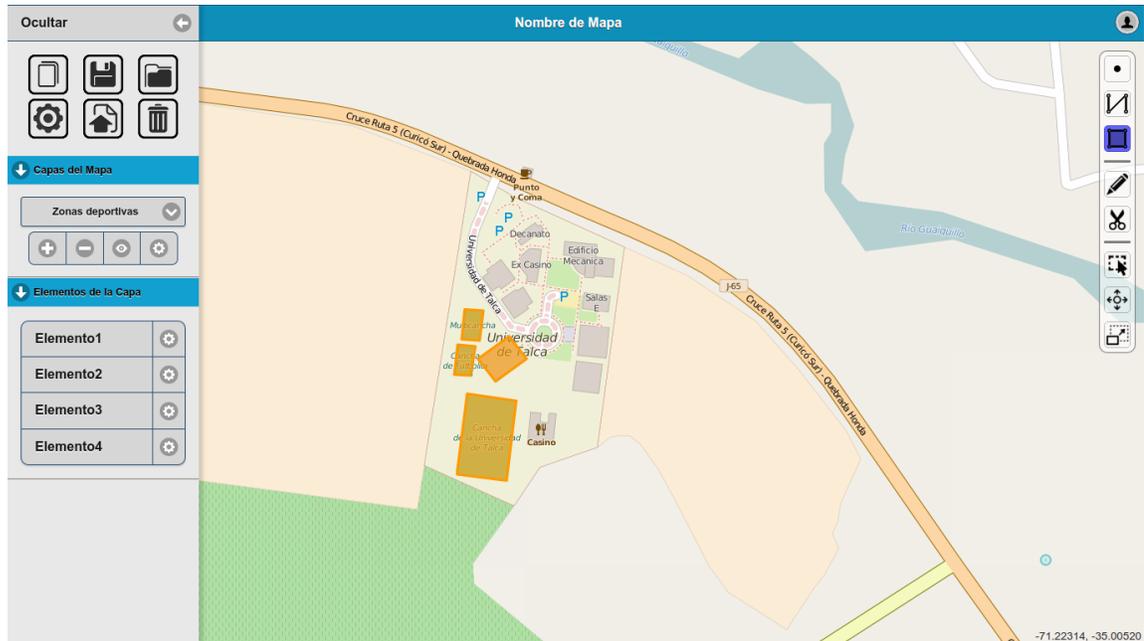


Figura 6.4: Captura del producto de la cuarta iteración: Vista Editor

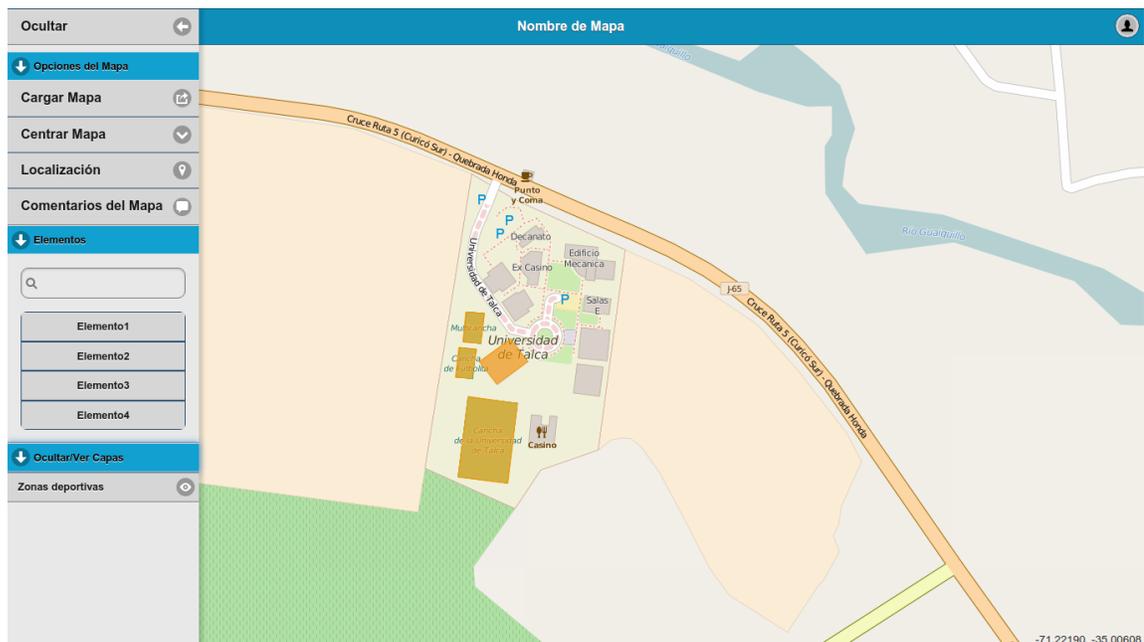


Figura 6.5: Captura del producto de la cuarta iteración: Vista Visor

## 6.3. Manejo de datos

### 6.3.1. Comunicación con el servidor

Para la comunicación entre el cliente y el servidor, se utilizó JSON

### 6.3.2. Conexión con la Base de Datos

Para la conexión entre el servidor y la Base de Datos, se utilizaron servicios creados en PHP.

### 6.3.3. Servicios Web

Como está especificado en el diseño arquitectónico del proyecto. MyGIS contará con servicios, los cuales se encargaran de administrar el traspaso de datos entre la aplicación cliente y la Base de Datos en el servidor.

A continuación se nombran y describen los distintos servicios creados:

**Editar Usuario:** Se le envía un objeto tipo [Usuario]. Reemplaza la versión existente en la Base de Datos, por la nueva versión del usuario. Entrega un booleano confirmando el éxito de la operación.

- Entrada: [Usuario] usuario.
- Salida: [boolean] confirmación.

**Eliminar Mapa:** Se le envía el identificador de un mapa. Elimina de la Base de Datos el mapa que coincida con el id enviado y sus respectivas capas, elementos y comentarios. Entrega un booleano confirmando el éxito de la operación.

- Entrada: [Int] id mapa.
- Salida: [boolean] confirmación.

**Guardar Capa:** Se le envía un objeto tipo [Capa]. Si la capa no existe en la Base de Datos, crea una nueva instancia de capa, para el respectivo mapa. Si la capa ya existe, reemplaza la versión existente por la nueva versión de la capa. Entrega un booleano confirmando el éxito de la operación.

- Entrada: [Capa] capa.
- Salida: [boolean] confirmación.

**Guardar Comentario:** Se le envía un objeto tipo [Comentario]. Crea una nueva instancia en la Base de Datos del comentario, para el respectivo mapa. Entrega un booleano confirmando el éxito de la operación.

- Entrada: [Comentario] comentario.
- Salida: [boolean] confirmación.

**Guardar Elemento:** Se le envía un objeto tipo [Elemento]. Si el elemento no existe en la Base de Datos, crea una nueva instancia de elemento, para el respectivo mapa. Si el elemento ya existe, reemplaza la versión existente por la nueva versión del elemento. Entrega un booleano confirmando el éxito de la operación.

- Entrada: [Elemento] elemento.
- Salida: [boolean] confirmación.

**Guardar Mapa:** Se le envía un objeto tipo [Mapa]. Si el mapa no existe en la Base de Datos, crea una nueva instancia de mapa. Si el mapa ya existe, reemplaza la versión existente por la nueva versión del mapa. Entrega un booleano confirmando el éxito de la operación.

- Entrada: [Mapa] mapa.
- Salida: [boolean] confirmación.

**Guardar Usuario:** Se le envía un objeto tipo [Usuario]. Crea una nueva instancia en la Base de Datos del usuario. Entrega un booleano confirmando el éxito de la operación.

- Entrada: [Usuario] usuario.
- Salida: [boolean] confirmación.

**Obtener Capa:** Se le envía el índice de una capa (id) y el identificador del mapa al que pertenece la capa. Entrega desde la Base de Datos, un objeto tipo [Capa] que coincida con los identificadores del mapa y la capa enviados.

- Entrada: [Int] id mapa, [Int] id capa
- Salida: [Capa] capa.

**Obtener Capas Del Mapa:** Se le envía el identificador de un mapa. Entrega desde la Base de Datos, una lista de objetos [Capas], que coincida con el identificador del mapa enviado.

- Entrada: [Int] id mapa.
- Salida: [Array Capa] capas.

**Obtener Comentarios del Mapa:** Se le envía el identificador de un mapa. Entrega desde la Base de Datos, una lista de objetos [Comentario], que coincida con el identificador del mapa enviado.

- Entrada: [Int] id mapa.
- Salida: [Array Comentario] comentarios.

**Obtener Elementos de la capa:** Se le envía el índice de una capa (id) y el identificador del mapa al que pertenece la capa. Entrega desde la Base de Datos, una lista de objetos [Elemento] que coincida con los ids del mapa y la capa enviados.

- Entrada: [Int] id mapa, [Int] id capa
- Salida: [Array Elemento] elementos.

**Obtener Mapa:** Se le envía el identificador de un mapa. Entrega un objeto tipo [Mapa] de la Base de Datos que coincida con el id del mapa.

- Entrada: mapa id.
- Salida: [Mapa] mapa.

**Obtener Nombres de capas:** Entrega una lista con los nombres de las capas registradas en la Base de Datos.

- Entrada: *sin entrada*.
- Salida: [Array String] capas

**Obtener Nombres de mapas:** Entrega una lista con los nombres de los mapas registrados en la Base de Datos.

- Entrada: *sin entrada*.
- Salida: [Array String] mapas

**Obtener Usuario:** Compara un código de un mapa, para ver si hay coincidencia con la registrada en la Base de Datos. Entrega un booleano dependiendo si se encontró coincidencia o no.

- Entrada: [String] correo, [String] contraseña.
- Salida: [boolean] validación.

**Validar Mapa:** Compara un correo del usuario (id) y una contraseña, para ver si hay coincidencia con las registradas en la Base de Datos. Entrega un booleano dependiendo si se encontró coincidencia o no.

- Entrada: [String] id, [String] código.
- Salida: [boolean] validación.

## 6.4. Manual de usuario

Si bien se emplearon grandes esfuerzos en que la aplicación sea lo más sencilla de utilizar e intuitiva. Es inevitable la necesidad del dominio de ciertos conceptos básicos al momento de crear un mapa. Estos conceptos aunque intuitivos para ciertos usuarios, no son tan evidentes para todos, en especial para los usuarios más novatos.

Por los motivos mencionados, es que se ha implementado un manual de usuario a modo de guiar al usuario al momento de crear un mapa con el editor de mapas. Este manual se encuentra accesible desde la misma aplicación cuando accede a la vista del editor de mapas.

#### 6.4.1. Descripción del manual

El manual es una interactiva y sencilla guía que cuenta con 7 pasos. Cada uno de estos pasos están constituidos por:

- Número y título del paso actual.
- Texto central explicativo del paso actual.
- Imágenes y texto laterales, los cuales señalan los distintos botones y funcionalidades que posee el sistema.

Estos pasos muestran al usuario las acciones necesarias a seguir, para la creación y publicación de un mapa. Los pasos y el texto central que presenta el manual son los siguientes:

**1. Iniciar Sesión:** Antes de poder acceder a un mapa, debes haber ingresado como usuario. Esto lo puedes hacer pulsando el botón “Iniciar Sesión” de la página de bienvenida o el botón de la barra, en la esquina superior derecha.

**2. Panel de Mapa:** Una vez hayas ingresado como usuario, puedes acceder a las herramientas de edición de mapa. Esto lo puedes hacer pulsando el botón de la barra, en la esquina superior izquierda.<sup>4</sup>

**3. Gestionar Mapa:** En el “panel de mapa” tendrás las herramientas para poder acceder, editar y publicar un mapa. Para acceder a un mapa, debes crear un nuevo mapa o cargar cualquier mapa público o uno privado mediante el código de edición. Esto lo puedes hacer pulsando los botones respectivos en la página de bienvenida o en el panel de mapa.

**4. Gestionar Capas:** Una vez hayas accedido a un mapa, debes crear una capa para poder crear elementos. Esto lo puedes hacer en el sector de gestión de capa en el panel de mapa con el botón “+”. Se pueden crear múltiples capas y seleccionar la capa en la que se desea trabajar.

**5. Gestionar Elementos:** Una vez hayas seleccionado una capa, puedes dibujar elementos sobre ella. Esto lo puedes hacer en la barra de herramientas de elementos, que aparecen en la esquina derecha de la pantalla, al seleccionar una capa. Podrás crear 3 tipos de elementos distintos: Puntos, Líneas y Polígonos.

**6. Ingresar Información a Elementos:** Una vez hayas dibujado elementos, aparecerán en la lista de elementos, donde podrás gestionar su nombre e información. Para gestionar un elemento, pulsa el botón de edición al costado del nombre del elemento en la lista. El tipo de información que puedes agregar, es muy amplia.

**7. Publicar Mapa:** Una vez hayas terminado de construir tu mapa con la información que desees comunicar, asegurate de guardar los cambios del mapa. Una vez guardado el mapa, puedes compartir tu mapa en las redes sociales. Esto lo puedes hacer en el sector de gestión de mapa.

## 6.5. Capturas del producto final

## 7. Evaluación

---

El éxito de este proyecto va más allá de obtener un software que cumpla los requisitos propuestos para su desarrollo, si no que en la utilidad que se le pueda dar a estos requisitos por los usuarios finales. Es por esto que además de las pruebas unitarias y de integración llevadas a cabo en la etapa de implementación, se recurrirá a otros métodos para la evaluación final del proyecto.

Con la finalidad de evaluar la efectividad del software, se hace necesario recurrir a la apreciación de los distintos usuarios, los cuales podrían hallar una herramienta útil para sus necesidades.

### 7.1. Pruebas de usuarios

Este proyecto de titulación, tiene como objetivo la generación de un prototipo de una red social con funcionalidades georeferenciales. Esto tiene el fin de generar una herramienta novedosa que satisfaga la necesidad de comunicar una información de forma efectiva. Para generar los requisitos y funcionalidades del sistema, no se contó con la participación de cliente específico que los facilitara, por lo tanto se recurrió a investigación en distintas índoles para poder determinar cuáles podrían ser los requisitos que mejor se adecuen a los objetivos propuestos para el proyecto.

Para culminar esta fase del proyecto (lo que comprende el módulo de proyecto de título), es necesario la evaluación de elementos difíciles de cuantificar y por lo tanto, difíciles de medir para realizar una evaluación satisfactoria y retroalimentativa. Es por esto que con el objetivo de realizar una mejor evaluación de esta fase, se

ideó la forma de que los usuarios finales participen de manera activa, aportando información decisiva para la evaluación del sistema. Esto se pretende lograr mediante la elaboración y aplicación de un cuestionario a diferentes usuarios, los cuales hayan utilizado el sistema con anterioridad.

### 7.1.1. Cuestionario

El cuestionario que se aplicará a los usuarios que vayan a contribuir en el proceso de evaluación, pretende determinar diferentes aspectos, los cuales son fundamentales para el sistema. Los aspectos que se pretenden medir con las pruebas de usuario son:

- Usabilidad del sistema.
- Eficiencia de la interfaz gráfica.
- Novedad que entrega el sistema.
- Utilidad que entrega el sistema.

### Survio

Con el objetivo de agilizar el proceso de evaluación de las pruebas de usuario, se recurre a la utilización de una aplicación web. La aplicación elegida es survio, la cual es un software para la elaboración y aplicación de encuestas online, con distintos propósitos. Se decidió utilizar esta herramienta, ya que facilita en gran medida las tareas de creación, aplicación y procesamiento de los resultados de la encuesta.

Aunque esta aplicación tiene una opción de pago para su uso, también cuenta con la opción de uso gratuito, bajo algunas restricciones respecto a la opción de pago. Sin embargo, la opción de uso gratuito cumple con los requerimientos necesarios para efectuar la encuesta del proceso de evaluación del proyecto.

### Elaboración de cuestionario

La aplicación Survio, cuenta con plantillas prefabricadas para distintos propósitos. Entre las múltiples plantillas que contiene este software, existen tres que satisfacen los objetivos de la evaluación de este proyecto. Los nombres de estas plantillas son:

- Evaluación de software
- Valoración del sitio web
- Evaluación de página web

Para la elaboración del cuestionario, se tomaron en cuenta algunas de las preguntas incluidas en estas plantillas mencionadas. Además se agregó algunas preguntas extra que se consideró conveniente añadir, con la finalidad de obtener una mejor evaluación de los aspectos deseados.

### **Preguntas**

El cuestionario cuenta con 12 preguntas o ítems, las cuales se pueden dividir en tres secciones diferentes:

1. Las 2 primeras preguntas van dirigidas a identificar la experiencia del usuario y en qué dispositivo se ejecuto la aplicación. Estas preguntas tienen alternativas para su respuesta.
2. Luego viene una sección de 9 preguntas, las cuales van dirigidas a obtener la apreciación del usuario respecto al sistema. Estas preguntas cuentan con una breve explicación y su respuesta consiste en una calificación numérica.
3. Finalmente existe una pregunta de carácter opcional, que entrega la opción de escribir de forma libre algún comentario respecto a la aplicación evaluada o a alguna de las preguntas del cuestionario.

Las preguntas de la segunda sección (de la 3 a la 11), consisten en la evaluación mediante una calificación en una escala de 1 a 7 de los diversos aspectos que se desea evaluar con la encuesta. Esto con el objetivo de que el cuestionario sea fácil de contestar y también para facilitar luego la interpretación de los resultados. Además cada una de estas preguntas, cuenta con pequeño texto explicativo respecto a que se refiere la pregunta, para evitar mal entendidos y hacer más objetiva la respuesta.

A continuación se presentan las preguntas que conforman el cuestionario:

**1. Indique en qué clase de dispositivo accedió a la aplicación.**

- Computadora personal
- Dispositivo móvil

**2. Indique la experiencia como usuario de aplicaciones computacionales y dispositivos móviles que tenga.**

- Usuario con poca experiencia
- Usuario con experiencia media
- Usuario experimentado

**3. Califique la complejidad del acceso a MyGIS.**

*Esto se refiere a que tan complicado fue entrar a la aplicación MyGIS (tanto al Editor como al Visor de mapas).*

**4. Califique el diseño de la interfaz gráfica de MyGIS.**

*Esto se refiere a que tan agradable es la estética de la interfaz gráfica de la aplicación en general (diseño de botones, colores, paneles, transiciones, etc.).*

**5. Califique la usabilidad de la interfaz de MyGIS-Editor.**

*Esto se refiere a que tan fácil es entender los controles, para acceder a las distintas funcionalidades del editor de mapas.*

**6. Califique la usabilidad de la interfaz de MyGIS-Visor.**

*Esto se refiere a que tan fácil es entender los controles, para acceder a las distintas funcionalidades del visor de mapas.*

**7. Califique la navegación por la interfaz de MyGIS-Editor.**

*Esto se refiere a que tan rápido es el acceso a las distintas funcionalidades del editor de mapas (una vez se conocen los controles de la aplicación).*

**8. Califique la navegación por la interfaz de MyGIS-Visor.**

*Esto se refiere a que tan rápido es el acceso a las distintas funcionalidades del visor de mapas (una vez se conocen los controles de la aplicación).*

**9. Califique la ayuda que entrega MyGIS-Editor para su uso.**

*Esto se refiere a que tan útil fue el tutorial o guía que tiene incorporado el editor de mapas.*

**10. Califique la novedad que entrega MyGIS como herramienta de comunicación.**

*Esto se refiere a si conoce una o más aplicaciones que equipare las funciones que entrega la aplicación. Especialmente la funcionalidad de agregar información georeferenciada de forma libre.*

**11. Califique el rendimiento de MyGIS.**

*Esto se refiere a que tan rápido y estable encuentra que funciona la aplicación MyGIS.*

**12. Si lo desea, puede dejar alguna observación sobre la aplicación MyGIS o sobre alguna pregunta del cuestionario.****Aplicación de cuestionario**

Para la aplicación del cuestionario, lo primero es que el usuario pruebe de forma libre la aplicación desarrollada (MyGIS). Con este fin, se subió la aplicación para que sea accesible de forma pública, a través de internet. Mediante el acceso online, se agiliza el proceso de evaluación, ya que el usuario podrá acceder al sistema de forma remota.

La dirección para acceder a la aplicación MyGIS es:

**Editor:** <http://tourism.ventures/mygis/editor>

**Visualizador:** <http://tourism.ventures/mygis/visor>

Una vez el usuario a utilizado la aplicación, puede contestar las preguntas propuestas en el cuestionario que se encuentra en la siguiente dirección:

**Cuestionario:** <http://www.surveio.com/survey/d/C9J6A9X8B4M9X5U8B>

Los usuarios se separados en 3 grupos, dependiendo de la experiencia que tengan en el uso de aplicaciones computacionales. A su vez, cada uno de esto 3 grupos se separó en 2 subgrupos, dependiendo de con qué dispositivo ejecutaron la prueba del sistema (computadora personal o dispositivo móvil).

Se eligió a 22 usuarios a los cuales aplicar el cuestionario, de los cuales:

- 6 cuentan con poca experiencia como usuarios de aplicaciones.
- 6 son experimentados como usuarios de aplicaciones.
- 10 cuentan con una experiencia media como usuarios de aplicaciones.

A su vez, a la mitad de los usuarios de cada uno de estos grupos se les pidió que probaran la aplicación en computadores personales y a la otra mitad se les pidió la probaran en teléfonos móviles.

### 7.1.2. Resultado de las pruebas de usuario

Para la representación de los resultados de la encuesta, se generaron tablas separadas de acuerdo a los grupos mencionados anteriormente. Las columnas de las tablas representan las preguntas, abreviadas “P [Número de pregunta]” y las filas representan a cada uno de los usuarios que respondió la encuesta. Cada celda indica la calificación que dio un usuario para una determinada pregunta.

A continuación se muestran las tablas con los resultados del cuestionario:

Cuadro 7.1: Usuarios con poca experiencia y teléfono móvil.

	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11
Usuario 1	7	6	3	6	4	6	4	6	5
Usuario 2	6	6	2	5	5	5	2	5	7
Usuario 3	6	7	4	7	4	7	4	7	6

Cuadro 7.2: Usuarios con poca experiencia y computadora personal.

	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11
Usuario 1	7	7	5	7	5	7	6	5	7
Usuario 2	7	7	4	6	5	6	5	7	7
Usuario 3	7	6	3	6	4	6	4	7	6

Cuadro 7.3: Usuarios con experiencia media y teléfono móvil.

	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11
Usuario 1	6	7	5	6	6	7	6	6	6
Usuario 2	7	5	3	4	6	7	3	5	5
Usuario 3	7	7	6	7	7	7	4	6	6
Usuario 4	7	5	4	7	6	6	4	7	6
Usuario 5	7	6	3	5	5	6	4	6	7

Cuadro 7.4: Usuarios con experiencia media y computadora personal.

	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11
Usuario 1	7	6	4	7	5	7	5	6	7
Usuario 2	7	7	4	7	6	7	6	7	7
Usuario 3	7	6	6	7	7	7	6	6	7
Usuario 4	7	6	5	6	6	6	4	5	6
Usuario 5	7	7	6	7	7	7	7	7	7

Cuadro 7.5: Usuarios experimentados y teléfono móvil.

	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11
Usuario 1	7	5	5	7	5	6	2	5	6
Usuario 2	7	6	4	5	6	7	4	6	5
Usuario 3	7	6	4	7	4	6	5	7	6

Cuadro 7.6: Usuarios experimentados y computadora personal.

	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11
Usuario 1	7	7	5	7	6	6	6	7	6
Usuario 2	7	6	6	7	6	7	6	7	5
Usuario 3	7	6	4	7	6	7	5	5	6

## 7.2. Resultados final de la evaluación

A continuación se presenta la tabla con el promedio para cada grupo y finalmente la media de todos los grupos para cada pregunta. Los grupos son:

1. Usuarios con poca experiencia y teléfono móvil.
2. Usuarios con poca experiencia y computadora personal.
3. Usuarios con experiencia media y teléfono móvil.
4. Usuarios con experiencia media y computadora personal.
5. Usuarios experimentados y teléfono móvil.
6. Usuarios experimentados y computadora personal.

Cuadro 7.7: Promedios de la evaluación de usuario.

	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11
<b>Grupo 1</b>	6,3	6,3	3	6	4,3	6	3,3	6	6
<b>Grupo 2</b>	7	6,3	4	6,3	3,6	6,3	5	6,3	6,6
<b>Grupo 3</b>	6,8	6	4,2	5,8	6	6,6	4,2	6	6,2
<b>Grupo 4</b>	7	6,4	5	6,8	6,2	6,8	5,6	6,2	6,8
<b>Grupo 5</b>	7	5,6	4,3	6,3	5	6,3	3,6	6	5,6
<b>Grupo 6</b>	7	6,3	5	7	6	6,3	5,6	6,3	6,3
	6,9	6,2	4,3	6,4	5,2	6,4	4,6	6,1	6,3

Si observamos la tabla (Cuadro 7.7) y se saca la media de todas las calificaciones, da como resultado un “5,8”. Esta nota se puede considerar bastante aceptable, por lo que se puede decir que se tuvo éxito después de pasar por la evaluación de usuarios. Sin embargo, si vemos en detalle cada una de las preguntas, la puntuación más baja la obtuvo la pregunta 5 con un “4,3”, la cual tiene que ver con la usabilidad del editor de mapas. Seguido por un “4,6” que tuvo la ayuda ofrecida por el editor. Esto hace evidente que aunque en general el prototipo es bien aceptado por los usuarios, tiene falencias en las que hay que trabajar a futuro. Sin embargo, las calificaciones obtenidas en las preguntas referentes al visualizador de mapas fueron muy satisfactorias.

Por otro lado, se puede apreciar que en general se obtuvieron calificaciones menores de los usuarios que utilizaron teléfonos móviles para probar el sistema. Esto se hace mas evidente en las preguntas referentes al editor de mapas. Por lo tanto, para ofrecer una mejor experiencia al usuario, es necesario trabajar también en la interfaz móvil para que la experiencia en un dispositivo móvil con pantalla reducida, sea igual que para un usuario de PC o dispositivos móviles con pantallas más grandes.

## 8. Conclusiones

---

Para concluir este proyecto y luego de haber pasado por cada una de las etapas necesarias para generar un sistema computacional, el cual consiste en el prototipo de una red social con funcionalidades georeferenciales para compartir información personalizada. El cual lleva por nombre MyGIS.

Se puede decir que se tuvo éxito en cumplir los objetivos propuestos. MyGIS satisface la totalidad de los requisitos propuestos y tuvo una evaluación aceptable por los usuarios finales. Sin embargo, al tener un carácter de prototipo, el sistema necesita de más desarrollo para obtener un sistema final. Sin embargo esta instancia ha servido de gran ayuda para evidenciar las fortalezas y falencias que presenta el sistema, lo que posibilita acortar el camino a seguir para seguir con el desarrollo de la aplicación.

Luego de haber pasado por un proceso muy importante en el que se indago y se obtuvo importante conocimiento sobre las tecnologías más usadas hoy en día, y mirando críticamente el trabajo realizado, cabe mencionar la dificultad a la hora de implementar una interfaz compatible con diversos dispositivos con características distintas y que a la vez, sea legible y comprensible para una amplia variedad de usuarios. A pesar de estas dificultades, se realizó un trabajo el cual dio resultados satisfactorios, así como la experiencia obtenida al desarrollar el sistema.

## 8.1. Trabajo futuro

El trabajo actual culminó con la generación de un prototipo de una red social. Al ser un prototipo, implica que no es un sistema completo. El sistema aún necesita de trabajo, para finalmente obtener un producto con la calidad necesaria para poder ponerlo online y que pueda ingresar todo usuario que lo desee.

Ya habiendo concluido el prototipo que se planteó como objetivo, y una vez hecho su evaluación y analizado los resultados. Se puede visualizar las falencias que presenta el prototipo actual para poder estar completa. Lo que se concluyó que es necesario mejorar para obtener una versión final del software son:

**Mejorar la ayuda:** La ayuda aun que entrega una guía general para utilizar el “Editor de mapas” poder construir un mapa. Esta es muy general y no explica mucha de las funcionalidades más específicas que tiene implementado el sistema. Es por esto que es necesario seguir desarrollando la ayuda y hacerla más accesible. Además, aunque el “Visualizador de mapas” tiene funcionalidades más intuitivas y fáciles de usar, este no cuenta con ayuda como el “Editor de mapas”, es por esto que se debería desarrollar una ayuda para esta vista igualmente.

**Mejorar interfaz móvil:** Aunque en general la interfaz gráfica es perfectamente visible y funcional en dispositivos móviles, hay algunas instancias en la que genera conflicto al utilizar una pantalla muy pequeña, sobre todo en el “Editor de mapas” (por ejemplo en la ayuda). Es por esto que se hace necesario rediseñar algunos aspectos de la interfaz, para que sea compatible cien por ciento con dispositivos móviles de cualquier tipo, tanto la vista del visor como la del editor de mapas.

**Mejorar seguridad:** Actualmente la aplicación maneja un sistema muy sencillo de usuarios y la restricción de acceso a los mapas. Por esto es necesario rediseñar y generar un mejor sistema de privilegios y acceso a los mapas.

**Mejorar usabilidad:** Existen controles de la aplicación que no son muy claros con qué funcionalidad están relacionados. Es por esto que hay que mejorar los controles, para que sean más intuitivos para los usuarios nuevos en la aplicación.

**Mejorar representación de elementos:** Muchos de los sistemas de información geográficos que existen en la actualidad, aparte de funcionar con la representación de elementos de forma vectorial (puntos, líneas y polígonos), tienen la opción de representar elementos con iconos e imágenes personalizadas. La posibilidad de subir y utilizar iconos e imágenes personalizados, aumentaría mucho la capacidad para comunicar que posee sistema.

**Incorporar interfaz para creación de mapas automáticos:** Actualmente la aplicación cuenta con una vista para crear mapas. Pero estos mapas solo los puede crear un usuario de forma manual. Agregaría mucho valor al sistema la posibilidad de automatizar la creación de mapas, entregando una interfaz que se comunique con sensores u otros aparatos con la capacidad de transmitir información mediante internet.

# Glosario

**TIC:** Las siglas 'TIC', hacen referencia a Tecnologías de la información y la Comunicación.

**SI:** Las siglas 'SI', hacen referencia a Sistema de Información.

**SIG/GIS:** Las siglas 'SIG' o 'GIS' (en ingles), hacen referencia a Sistema de Información Geográfico.

**BD:** Las siglas 'BD', hacen referencia a Base de Datos.

**GPS:** Las siglas 'GPS', hacen referencia en ingles a Global Positioning System, que en español es Sistema de Posicionamiento Global.

**SOA:** Las siglas 'SOA', hacen referencia en ingles a Arquitectura orientada a servicios, que en español es Arquitectura Orientada a Servicios.

**MVC:** Las siglas 'MVC', hacen referencia al patrón de arquitectura de software Modelo Vista Controlador.

# Bibliografía

- [1] Yesenia Cetina. La computadora y la comunicación. *academiaonline.com.mx*, page 1, 2010.
- [2] Laboratorio Unidad Pacífico Sur CIESAS. Sistemas de información geográfica. <https://langleruben.wordpress.com>, page 1, 2010.
- [3] Confederación de Empresarios de Andalucía. Sistemas de información geográfica, tipos y aplicaciones empresariales. <http://sig.cea.es>, pages 4–6, 2010.
- [4] Beatriz de la Fuente. Comunicación mediante la computación. <http://uva.anahuac.mx>, page 1, 2003.
- [5] Definicion.de. Definición de red social. *definicion.de*, page 1, 2015.
- [6] EcuRed. Arquitectura cliente servidor. [www.guiadigital.gob.cl/](http://www.guiadigital.gob.cl/), page 1, 2011.
- [7] Daniel Gonzáles. Tecnologías de la información y la comunicación (tic's). *Mono-grafías*, pages 1–2, 2014.
- [8] Nicolás Montés. Ranking de sistemas operativos más usados para 2015. [blog.uchceu.es](http://blog.uchceu.es), page 5, 2015.
- [9] José Miguel Santibáñez Allendes. Sistemas de información. <http://jms.caos.cl>, pages 1–2, 2014.
- [10] Julián David Morillo Pozo. Introducción a los dispositivos móviles. [www.exabyteinformatica.com](http://www.exabyteinformatica.com), page 34, 2010.
- [11] Naciones Unidas. Como hacer comprensibles los datos. 2(20-30), 2009.
- [12] Wikipedia. Cliente servidor. *es.wikipedia.org*, page 1, 2015.