

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

**UNIVERSIDAD DEL BÍO-BÍO**  
FACULTAD DE CIENCIAS EMPRESARIALES  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y  
TECNOLOGÍA DE LA INFORMACIÓN  
CAMPUS CHILLÁN



**“Sistema Web de Geolocalización de Direcciones de Estudiantes  
carrera Ingeniería Civil en Informática”**

**JAIME FERNANDO DURÁN GALLEGOS**

**MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN INFORMÁTICA**

**Chillán, Noviembre 2014**

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

**UNIVERSIDAD DEL BÍO-BÍO**  
FACULTAD DE CIENCIAS EMPRESARIALES  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y  
TECNOLOGÍA DE LA INFORMACIÓN  
CAMPUS CHILLÁN



**“Sistema Web de Geolocalización de Direcciones de Estudiantes  
carrera Ingeniería Civil en Informática”**

**JAIME FERNANDO DURÁN GALLEGOS**

**Profesor Guía : Sr. Luis Gajardo Díaz**  
**Profesor Informante : Sr. Miguel Pincheira Caro**

**Nota Final del Proyecto de Título: \_\_\_\_\_**

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

## *Agradecimientos*

*En primer lugar agradezco a mi Dios por todo lo que ha hecho en mi vida, agradezco a mi abuela por su esfuerzo, dedicación, paciencia y por los valores que me entregó, a mi amada esposa por su apoyo constante y preocupación.*

*Además agradezco el apoyo, la confianza y el tiempo del profesor Luis Gajardo, que creyó en mí en esta última etapa, a mi equipo de trabajo en Equifax por sus consejos y a la Universidad del Bío – Bío por darme la oportunidad de vivir una de las mejores experiencias de mi vida.*

## *Dedicatoria*

*Dedico este trabajo a Laura Acuña, mujer esforzada que se sacrificó cada día de sus últimos años para que yo pudiera alcanzar mis metas.*

## Resumen

El presente informe detalla el desarrollo de un sistema orientado a la integración de la información referente a la residencia y domicilio de los estudiantes de la carrera de Ingeniería Civil en Informática de la Universidad del Bío-Bío, con el fin mejorar el acceso a esta en caso de catástrofes.

Actualmente, solo es posible acceder a la información de un estudiante en particular, si bien es posible obtener los registros de su domicilio, no se puede dar una mayor interpretación a este dato. Por lo que en caso de identificar los estudiantes comprometidos en una catástrofe que afecte a una zona en particular, no es factible obtener dicha información de manera rápida.

Dicha problemática sería solucionable creando un sistema que permita la mantención de los datos de los estudiantes referente al domicilio de estos, para que posteriormente el procesamiento de los datos permita a los funcionarios de la Universidad una interpretación efectiva en caso de ser necesario.

Un sistema así puede generar un nuevo proceso que permita un rápido acceso a la información referente a la residencia y contacto de un estudiante en situaciones de catástrofe. Otorgando una mejor interpretación de los datos mediante el uso de mapas con la ubicación de los estudiantes y zonas demarcadas, dando una percepción de la información más cercana a la realidad.

El sistema web se implementó utilizando el framework de Yii, usando el api de Google Maps para los servicios de georeferencia. Todo esto interactuando con una Base de Datos MySQL, de tipo relacional, cuya arquitectura fue diseñada mediante un MER, poniendo énfasis en evitar la redundancia de datos. Se aplicaron conceptos de usabilidad tales como migajas de pan en la navegación, claridad en los nombres de botones y opciones de menús, buscando mejorar la experiencia del usuario. Los servicios se agruparon por tipos de usuarios, los cuales accederán mediante un usuario y contraseña.

Como resultado se obtiene un sistema web que permite la mantención de una gran cantidad de información, que otorga el servicio de un mapa dinámico que contiene suficiente información para identificar rápidamente la zona afectada y los estudiantes en esta, pudiendo priorizar a estos últimos en casos de emergencia. Creando un nueva fuente de datos cuyo beneficio es la posibilidad de usarlos para nuevos procesamientos.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

## Índice General

<b>1</b>	<b>DESCRIPCIÓN DEL PROYECTO</b>	<b>16</b>
<b>1.1</b>	<b>INTRODUCCIÓN GENERAL</b>	<b>16</b>
<b>1.2</b>	<b>DEFINICIÓN DE LA INSTITUCIÓN</b>	<b>18</b>
1.2.1	DESCRIPCIÓN DE LA INSTITUCIÓN	18
1.2.1.1	Visión	19
1.2.1.2	Misión	19
1.2.2	FACULTAD DE CIENCIAS EMPRESARIALES	20
1.2.2.1	Visión	21
1.2.2.2	Misión	21
1.2.3	DESCRIPCIÓN DEL ÁREA DE ESTUDIO	23
1.2.3.1	Carrera de ICI	23
1.2.3.2	Cargos Administrativos	24
1.2.3.3	Visión	25
1.2.3.4	Misión	25
<b>1.3</b>	<b>DESCRIPCIÓN DEL PROBLEMA</b>	<b>25</b>
<b>1.4</b>	<b>SOLUCIÓN PLANTEADA</b>	<b>26</b>
<b>1.5</b>	<b>OBJETIVOS</b>	<b>26</b>
1.5.1	GENERAL	26
1.5.2	ESPECÍFICOS	26
<b>1.6</b>	<b>ALCANCES Y LIMITACIONES</b>	<b>26</b>
<b>1.7</b>	<b>DEFINICIONES, ABREVIACIONES Y SIGLAS</b>	<b>27</b>
<b>2</b>	<b>MARCO TEÓRICO</b>	<b>31</b>
<b>2.1</b>	<b>INTRODUCCIÓN</b>	<b>31</b>
<b>2.2</b>	<b>SISTEMA DE INFORMACIÓN GEOGRÁFICA</b>	<b>32</b>
2.2.1	DEFINICIÓN Y FUNCIONAMIENTO	32
<b>2.3</b>	<b>WEB MAPPING</b>	<b>33</b>
2.3.1	API DE MAPAS ONLINE	34
2.3.1.1	Api de Google Maps	35
2.3.1.2	Utilidades de GoogleMaps	36
2.3.2	METODOLOGÍA DE DESARROLLO	37
2.3.3	MODELADO	38
<b>2.4</b>	<b>ENFOQUE ORIENTADO A OBJETOS</b>	<b>39</b>
2.4.1	CONCEPTOS BÁSICOS	39
2.4.2	FUNDAMENTOS DEL ENFOQUE ORIENTADO A OBJETOS	40
<b>2.5</b>	<b>ARQUITECTURA DEL SOFTWARE</b>	<b>41</b>
<b>2.6</b>	<b>TECNOLOGÍA PARA ALMACENAMIENTO PERSISTENTE</b>	<b>42</b>
<b>2.7</b>	<b>LENGUAJE DE PROGRAMACIÓN</b>	<b>42</b>
<b>2.8</b>	<b>FRAMEWORK DE DESARROLLO</b>	<b>42</b>
<b>2.9</b>	<b>USABILIDAD</b>	<b>44</b>
<b>2.10</b>	<b>GEOCODIFICACIÓN</b>	<b>46</b>
2.10.1	PRINCIPIOS	46

2.10.2	COMPONENTES EN EL PROCESO DE GEOCODIFICACIÓN .....	46
2.10.3	PROCESO DE GEOLOCALIZACIÓN .....	46
2.10.4	NORMALIZACIÓN Y ESTANDARIZACIÓN .....	46
<b>3</b>	<b><u>ANÁLISIS DE REQUERIMIENTOS .....</u></b>	<b>50</b>
<b>3.1</b>	<b>INTRODUCCIÓN .....</b>	<b>50</b>
<b>3.2</b>	<b>REQUISITOS OPERACIONALES .....</b>	<b>51</b>
<b>3.3</b>	<b>INTERFAZ DE USUARIO .....</b>	<b>51</b>
<b>3.4</b>	<b>INTERFAZ DE HARDWARE.....</b>	<b>51</b>
<b>3.5</b>	<b>INTERFAZ DE SOFTWARE.....</b>	<b>52</b>
<b>3.6</b>	<b>INTERFACES DE COMUNICACIÓN.....</b>	<b>52</b>
<b>3.7</b>	<b>REQUERIMIENTOS ESPECÍFICOS .....</b>	<b>52</b>
3.7.1	REQUERIMIENTOS FUNCIONALES.....	52
3.7.1.1	Usuario, roles y tipos.....	52
3.7.1.2	Ingreso y visualización de Estudiantes.....	53
3.7.1.3	Ingreso y visualización de Eventos.....	53
3.7.1.4	Reportes.....	53
3.7.1.5	Sesiones .....	53
3.7.2	INFORMACIÓN DE ENTRADA .....	54
3.7.3	INFORMACIÓN DE REPORTE .....	54
<b>3.8</b>	<b>ATRIBUTOS DEL PRODUCTO .....</b>	<b>54</b>
3.8.1	USABILIDAD - OPERATIVIDAD.....	54
3.8.2	EFICIENCIA- TIEMPO DE EJECUCIÓN/RESPUESTA.....	54
3.8.3	FUNCIONALIDAD-SEGURIDAD .....	55
<b>4</b>	<b><u>FACTIBILIDAD.....</u></b>	<b>58</b>
<b>4.1</b>	<b>INTRODUCCIÓN .....</b>	<b>58</b>
<b>4.2</b>	<b>FACTIBILIDAD TÉCNICA.....</b>	<b>60</b>
<b>4.3</b>	<b>FACTIBILIDAD OPERATIVA.....</b>	<b>61</b>
<b>4.4</b>	<b>FACTIBILIDAD ECONÓMICA .....</b>	<b>61</b>
4.4.1	COSTO DE DESARROLLO.....	63
4.4.2	COSTO DE INSTALACIÓN .....	63
4.4.3	COSTO DE OPERACIÓN .....	63
4.4.4	COSTO MANTENCIÓN .....	63
4.4.5	BENEFICIOS INTANGIBLES .....	64
<b>5</b>	<b><u>DISEÑO.....</u></b>	<b>66</b>
<b>5.1</b>	<b>INTRODUCCIÓN .....</b>	<b>66</b>
<b>5.2</b>	<b>DISEÑO INTERFAZ Y NAVEGACIÓN.....</b>	<b>67</b>
<b>5.3</b>	<b>CASOS DE USO .....</b>	<b>68</b>
5.3.1	DIAGRAMA DE CASOS DE USO .....	68
5.3.2	DEFINICIÓN DE CASOS DE USO .....	70
5.3.2.1	Caso de Uso: Iniciar Sesión .....	70
5.3.2.2	Caso de Uso: Ver Listado de Estudiantes .....	70
5.3.2.3	Caso de Uso: Ver Información Estudiante .....	71
5.3.2.4	Caso de Uso: Actualizar Estudiante .....	71

5.3.2.5	Caso de Uso: Ingresar Estudiante .....	72
5.3.2.6	Caso de Uso: Eliminar Estudiante .....	72
5.3.2.7	Caso de Uso: Ver Listado de Eventos .....	73
5.3.2.8	Caso de Uso: Ver Información Evento.....	73
5.3.2.9	Caso de Uso: Actualizar Evento .....	73
5.3.2.10	Caso de Uso: Ingresar Evento .....	74
5.3.2.11	Caso de Uso: Eliminar Evento.....	74
5.3.2.12	Caso de Uso: Ingresar Estudiante Afectado.....	75
5.3.2.13	Caso de Uso: Reportes .....	75
5.3.2.14	Caso de Uso: Finalizar Sesión.....	76
5.3.2.15	Caso de Uso: Ver Listado de Usuarios.....	76
5.3.2.16	Caso de Uso: Ver Información Usuario .....	76
5.3.2.17	Caso de Uso: Actualizar Usuario.....	77
5.3.2.18	Caso de Uso: Ingresar Usuario.....	77
5.3.2.19	Caso de Uso: Eliminar Usuario.....	78
<b>5.4</b>	<b>DESCRIPCIÓN GLOBAL DEL PRODUCTO .....</b>	<b>78</b>
5.4.1	PÁGINA DE INICIO.....	78
5.4.2	PÁGINA LISTADO DE ESTUDIANTES.....	79
5.4.3	PÁGINA DETALLE DE ESTUDIANTE .....	79
5.4.4	PÁGINA AGREGAR NUEVO ESTUDIANTE.....	80
5.4.5	PÁGINA LISTADO DE EVENTOS.....	81
5.4.6	PÁGINA DETALLE DE UN EVENTO.....	81
5.4.7	PÁGINA CREAR NUEVO EVENTO.....	82
5.4.8	PÁGINA AGREGAR ESTUDIANTE AFECTADO .....	83
5.4.9	PÁGINA DE REPORTES .....	83
<b>5.5</b>	<b>MODELAMIENTO DE DATOS.....</b>	<b>84</b>
<b>5.6</b>	<b>DISEÑO FÍSICO DE LA BASE DE DATOS.....</b>	<b>86</b>
<b>5.7</b>	<b>DISEÑO DE ARQUITECTURA FUNCIONAL.....</b>	<b>87</b>
<b>6</b>	<b>IMPLEMENTACIÓN.....</b>	<b>90</b>
<b>6.1</b>	<b>INTRODUCCIÓN .....</b>	<b>90</b>
<b>6.2</b>	<b>API DE GOOGLE MAPS.....</b>	<b>91</b>
6.2.1	INTRODUCCIÓN.....	91
6.2.2	CARACTERÍSTICAS DE GOOGLE MAPS.....	91
6.2.3	GOOGLE MAPS JAVASCRIPT API .....	92
6.2.4	FUNCIONAMIENTO DE GOOGLE MAPS.....	92
<b>6.3</b>	<b>SISTEMA INTEGRADO DE INFORMACIÓN TERRITORIAL.....</b>	<b>93</b>
<b>6.4</b>	<b>CONVERSIÓN DE SHP A KML.....</b>	<b>93</b>
<b>6.5</b>	<b>USO DE MAPAS.....</b>	<b>97</b>
6.5.1	EGMAP .....	97
6.5.2	GOOGLE MAPS Y JAVASCRIPT .....	98
<b>6.6</b>	<b>USABILIDAD.....</b>	<b>98</b>
6.6.1	MIGAJAS DE PAN .....	98
6.6.2	VISIBILIDAD DEL ESTADO DEL SISTEMA .....	99
6.6.3	CONTROL POR PARTE DEL USUARIO.....	99
6.6.4	SIMILITUD ENTRE EL SISTEMA Y EL MUNDO REAL.....	100

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

6.6.5	CONSISTENCIA Y ESTANDARIZACIÓN.....	100
6.6.6	RECONOCER ANTES DE MEMORIZAR.....	100
6.6.7	ESTÉTICA Y MINIMALISMO .....	100
<b>7</b>	<b><u>SEGURIDAD Y PRUEBAS .....</u></b>	<b><u>103</u></b>
<b>7.1</b>	<b>INTRODUCCIÓN .....</b>	<b>103</b>
<b>7.2</b>	<b>SEGURIDAD .....</b>	<b>104</b>
7.2.1	SEGURIDAD MEDIANTE OSCURIDAD.....	104
7.2.2	SEGURIDAD MEDIANTE CONTRASEÑAS.....	104
7.2.3	ENCRIPCIÓN.....	105
7.2.4	NIVELES DE ACCESO.....	105
7.2.5	TIPOS DE ATAQUE .....	105
<b>7.3</b>	<b>CASOS DE PRUEBA .....</b>	<b>106</b>
7.3.1	ELEMENTOS DE PRUEBA.....	106
7.3.1.1	Módulo Estudiantes .....	106
7.3.1.2	Módulo Eventos .....	106
7.3.1.3	Módulo Reportes.....	106
7.3.2	ESPECIFICACIÓN DE LAS PRUEBAS .....	107
7.3.2.1	Características a Probar .....	107
7.3.2.2	Nivel de Pruebas.....	107
7.3.2.3	Objetivo de la Prueba.....	107
7.3.2.4	Enfoque.....	107
7.3.2.5	Actividades de Prueba.....	107
7.3.3	RESPONSABLE DE LAS PRUEBAS .....	107
7.3.4	DETALLES DE LAS PRUEBAS .....	108
7.3.4.1	Pruebas de Unidad .....	108
7.3.4.2	Pruebas de Seguridad .....	111
7.3.5	CONCLUSIONES DE PRUEBAS.....	112
<b>8</b>	<b><u>CONCLUSIONES.....</u></b>	<b><u>115</u></b>
<b>8.1</b>	<b>CONCLUSIONES FINALES .....</b>	<b>115</b>
<b>8.2</b>	<b>TRABAJO FUTURO.....</b>	<b>116</b>
	<b><u>BIBLIOGRAFÍA .....</u></b>	<b><u>119</u></b>
	ANEXO A - El API de codificación geográfica de Google .....	<b><u>124</u></b>
	ANEXO B - Servicio de asignación de identificadores geográficos.....	<b><u>136</u></b>

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

## Índice Figuras

Figura 1. Estructura Organizacional Universidad del Bío-Bío.....	20
Figura 2 Estructura Organizacional Facultad de Ciencias Empresariales.....	22
Figura 3 Estructura Organizacional Ingeniería Civil en Informática .....	25
Figura 4 SIG .....	32
Figura 5 Vista desde Google Maps.....	35
Figura 6 Modelo Incremental de Larman.....	37
Figura 7 UML, Esquema conceptual de diagramas .....	38
Figura 8 Diagrama Modelo Vista - Controlador .....	41
Figura 9 Típico flujo de trabajo en Yii.....	43
Figura 10 Geocodificación y normalización dirección Campus Fernando May.....	47
Figura 11 Diseño Interfaz Sitio Web .....	67
Figura 12 Diagrama Casos de Uso.....	69
Figura 13 Página de Inicio a)Login b)Menus de acceso para sesión iniciada.....	79
Figura 14 Listado de Estudiantes .....	79
Figura 15 Detalle de un Estudiante.....	80
Figura 16 Ingresar nuevo Estudiante .....	80
Figura 17 Listado de Eventos.....	81
Figura 18 Detalle de un Evento.....	82
Figura 19 Creación nuevo Evento .....	82
Figura 20 Ingreso de un Estudiante Afectado por un Evento.....	83
Figura 21 Página de Reportes.....	83
Figura 22 Modelo Entidad Relación .....	84
Figura 23 Modelo Físico Base de Datos.....	86
Figura 24 Modelo de Arquitectura Funcional.....	87
Figura 25 Segmento de código en formato KML .....	92
Figura 26 Shape Escape Interfaz.....	94
Figura 27 Fusion Table creada por Shape Escape .....	94
Figura 28 Resultado de Conversión de Fusion Table a CSV.....	95
Figura 29 Columnas contenidas en archivo csv.....	95
Figura 30 Conversión de CSV a KML.....	96
Figura 31 Archivo en estándar KML .....	96
Figura 32 Visualización de un polígono en un mapa.....	97
Figura 33 Migajas de Pan .....	98
Figura 34 Visibilidad del estado del sistema .....	99
Figura 35 Visibilidad de las posibles acciones.....	99
Figura 36 Confirmación de Eliminación de un Estudiante.....	100
Figura 37 Diseños conceptuales previos a la asesoría de diseñadora gráfica.....	101

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

## Índice Tablas

Tabla 1. Información de Entrada.....	54
Tabla 2 Información de salida en los Reportes.....	54
Tabla 3 Requerimientos de Hardware y Software .....	61
Tabla 4 Inversión Inicial .....	63
Tabla 5 Caso de Uso Iniciar Sesión.....	70
Tabla 6 Caso de Uso Listar Estudiantes.....	70
Tabla 7 Caso de Uso Ver el detalle de un Estudiante.....	71
Tabla 8 Caso de Uso Modificar datos de un Estudiante .....	71
Tabla 9 Caso de Uso Agregar nuevo Estudiante .....	72
Tabla 10 Caso de Uso Eliminar Estudiante.....	72
Tabla 11 Listar Eventos .....	73
Tabla 12 Caso de Uso Ver Detalle de un Evento.....	73
Tabla 13 Caso de Uso Editar datos de un Evento.....	74
Tabla 14 Caso de Uso Crear nuevo Evento .....	74
Tabla 15 Caso de Uso Eliminar Evento.....	75
Tabla 16 Caso de Uso Ingresar Estudiante Afectado.....	75
Tabla 17 Caso de Uso Reportes.....	75
Tabla 18 Caso de Uso Cerrar Sesión.....	76
Tabla 19 Listar Usuarios.....	76
Tabla 20 Caso de Uso Ver Detalle de un Usuario .....	77
Tabla 21 Caso de Uso Editar datos de un Usuario .....	77
Tabla 22 Caso de Uso Crear nuevo Usuario .....	78
Tabla 23 Caso de Uso Eliminar Usuario .....	78
Tabla 24 Caso de Prueba Inicio de sesión .....	108
Tabla 25 Caso de Prueba Ingresar Nuevo Estudiante.....	109
Tabla 26 Caso de Prueba Ingresar Nuevo Evento.....	109
Tabla 27 Caso de Prueba Agregar estudiante a un evento.....	109
Tabla 28 Caso de Prueba Modificar Estudiante .....	110
Tabla 29 Caso de Prueba Modificar Evento.....	110
Tabla 30 Obtener un Reporte .....	110
Tabla 31 Resultado Prueba de Seguridad .....	111

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

## Descripción del Proyecto

---

# 1 DESCRIPCIÓN DEL PROYECTO

---

## 1.1 Introducción General

Hoy en día es fácil comprender el rápido proceso de expansión espacial por el cual están pasando muchas de las grandes ciudades de Chile, esto conlleva que creen nuevas zonas residenciales a un ritmo acelerado. Ciudades como Chillán y Los Ángeles, conocidas como ciudades intermedias, en estos últimos 30 años han presentado una gran mejora en su calidad de vida y con esto un importante crecimiento a nivel poblacional, debido en su mayoría a las migraciones desde el sector rural.

Este crecimiento se ha visto acompañado por la necesidad de aumentar el nivel educacional de las personas, es por ello que muchos estudiantes se han encontrado en la necesidad de dejar sus lugares de origen para cumplir con sus ambiciones profesionales, lo que ha dotado a las universidades de un número de estudiantes de diferentes lugares.

Este aumento en el número de matrículas obliga a las universidades a mantener registros con la diferente información personal de sus estudiantes, incluyendo claramente el o los domicilios de estos. Esta información, al ser entregada por primera fuente, se puede concluir que es fidedigna y fácil de conseguir. Estos datos obtenidos pueden almacenarse fácilmente en grandes volúmenes, lo cual no supone un problema para las universidades.

Pese a la facilidad mencionada anteriormente, muchas veces el propio formato de las direcciones no permite una interpretación correcta y rápida, ya sea de manera individual o cuando se encuentran agrupados, entregando en algunas ocasiones una visión errónea o ambigua de la información contenida en estos datos.

Quizás para algunos o en circunstancias normales dicho problema de interpretación no resulte algo grave, en no muchas ocasiones es necesario identificar dónde se ubica geográficamente una dirección, ni si dos direcciones son cercanas entre sí. Pero el terremoto ocurrido en el país en Febrero de 2010 afectó a gran parte de la zona central, dejando aislados a muchos sectores, incluyendo a un gran número de estudiantes de las diferentes casas de estudio.

Esta incomunicación produjo una incertidumbre dentro de las autoridades, sobre todo de los centros de educación, debido a que si bien eran conocidas las zonas afectadas, no había una manera rápida de diagnosticar cuáles y cuántos eran los estudiantes afectados o que pertenecían a los lugares siniestrados.

Esta situación también sucede cuando se trata de utilizar los datos disponibles para generar nueva información útil, ejemplo de esto es el caso de querer conocer el número de estudiantes pertenecientes a un determinado sector o localidad, lo cual si bien pareciera algo sencillo de interpretar, no lo es cuando se trata de datos tan diferentes como puede ser las direcciones, donde no existe un patrón que pudiese indicar que dos direcciones con distinto valor puedan ser próximas o no.

Es esta dificultad en la interpretación lo que motiva el desarrollo de una solución que facilite a personas con mínimos conocimientos cartográficos obtener información valiosa sobre las ubicaciones de los estudiantes, en este caso estudiantes de la Universidad del Bío-Bío, utilizando solo las direcciones de estos. Siendo esta solución una aplicación cuya interfaz permita la visualización de las ubicaciones referenciales de los estudiantes, pudiendo obtener diferentes reportes e incluso graficar en un mapa, por ejemplo, todos los estudiantes que pertenezcan a un sector en particular.

Por lo anteriormente expuesto es que se desea desarrollar una sistema basado en plataforma web que permita a algunos funcionarios de la Universidad del Bío-Bío obtener información referente a la ubicación de los estudiantes de la carrera de Ingeniería Civil en Informática de las diferentes sedes de una manera visualmente comprensible.

Para el desarrollo de este proyecto, es necesario definir una estructura que servirá de columna vertebral dentro del informe, la cual contemplará los siguientes tópicos:

Capítulo 1. Descripción del proyecto: Se analizará el entorno que rodea al proyecto, desde el sector institucional donde estará inserto, la problemática que resuelve, los objetivos que persigue y las limitaciones de este.

Capítulo 2. Marco teórico: Se analizará la teoría relacionada con aquellos aspectos esenciales utilizados en el desarrollo de este proyecto.

Capítulo 3. Especificación de requerimientos: Se definirán los detalles del producto a desarrollar, desde visión global hasta las propias especificaciones de este.

Capítulo 4. Factibilidad: En este capítulo se analiza qué tan viable puede llegar a ser el desarrollo del proyecto.

Capítulo 5. Diseño: Se realizan una serie de esquemas y diagramas para concretar la etapa previa a la construcción del proyecto.

Capítulo 6. Implementación: Se analizaran los aspectos más importantes de la codificación de los módulos del proyecto incluyendo el desarrollo de prototipos.

Capítulo 7. Seguridad y pruebas de integración: Conjunto de parámetros que se construyeron para realización de pruebas del sistema, con el fin de observar su estado de consistencia y robustez.

Capítulo 8. Conclusiones: Se hace un análisis del trabajo realizado, abordando los temas más relevantes y proponiendo ideas para trabajos futuros.

## 1.2 Definición de la Institución

### 1.2.1 Descripción de la Institución

La Universidad del Bío-Bío es heredera de la más antigua tradición de la educación superior estatal y pública en la Región del Bío-Bío. Sus orígenes se remontan a la creación de la Universidad Técnica del Estado, UTE, el 9 de abril de 1947, bajo la presidencia de Gabriel González Videla.

Dependiente del Ministerio de Educación Pública, la nueva institución fusionó en su interior la Escuela de Ingenieros Industriales y los grados técnicos de la Escuela de Artes y Oficios de Santiago; Escuela de Minas de Antofagasta, Copiapó y La Serena y las Escuelas Industriales de Concepción, Temuco y Valdivia.

La UTE abrió oficialmente sus puertas en 1952, luego que el Senado aprobó su Estatuto Orgánico, contrariando la férrea oposición de la Universidad de Chile, cuyas autoridades consideraban que la nueva casa de estudios superiores debía funcionar bajo su tuición. En Concepción, el plantel jugaría un importante papel para responder a las necesidades y desafíos que plantea la Región como uno de los polos del desarrollo industrial del país, no sólo a través de la docencia de pregrado sino que también mediante la investigación científica y tecnológica.

A las carreras técnicas de Electricidad, Mecánica y Textil que se impartían en 1959 se sumaron, en 1969, las de Ingeniería de Ejecución en Electricidad, Mecánica y Madera. Ese mismo año se creó la carrera de Arquitectura, hecho que constituye un hito significativo: De acuerdo con la nueva Ley de Universidades dictada en 1980, la Sede Concepción de la UTE pasó a ser una universidad autónoma –la Universidad de Bío-Bío- por impartir Arquitectura, definida entonces como una de las 12 carreras universitarias.

No corrió igual suerte la Universidad de Chile, Sede Ñuble, que venía funcionando en Chillán desde 1966. La nueva legislación, que obligó a convertir en universidades a las sedes regionales, de acuerdo con el concepto de regionalización y desconcentración, dio lugar al surgimiento del Instituto Profesional de Chillán, IPROCH, ya que entre sus carreras no figuraba ninguna de las consideradas universitarias.

La creación de la Sede Ñuble de la Universidad de Chile, primero como Colegio Regional, había sido el fruto de un amplio movimiento ciudadano destinado a evitar que los jóvenes egresados de la Enseñanza Media tuvieran que emigrar a Santiago u otras ciudades para continuar la enseñanza superior. En sus inicios, ocupó las dependencias cedidas por la Sociedad Musical Santa Cecilia, además de un edificio en avenida Libertad, donde funcionó la Escuela de Idiomas. Posteriormente, en 1973, recibió la donación de 33 hectáreas del fundo El Mono, propiedad de Fernando May Didier, comenzando la construcción del actual Campus Fernando May. El año 1981, en tanto, el patrimonio del naciente IPROCH se incrementó con la incorporación de las antiguas instalaciones de la Escuela Normal de Chillán donde ahora se encuentra el Campus La Castilla. Más tarde, en 1988, la fusión de la Universidad de Bío-Bío y el Instituto Profesional de Chillán dio origen a la que es hoy la Universidad del Bío-Bío, uniendo a dos instituciones que asumieron el desafío de caminar juntas y construir una historia en común.

### **1.2.1.1 Visión**

Ser reconocida a nivel nacional como una Universidad estatal, pública, regional, autónoma, compleja e innovadora con énfasis en la formación de capital humano, vinculada al desarrollo sustentable de la Región del Biobío y que aporta a la sociedad del conocimiento y al desarrollo armónico del país.

### **1.2.1.2 Misión**

La Universidad del Bío-Bío es una institución de educación superior, pública, estatal y autónoma, de carácter regional, que se ha propuesto por misión:

- Formar profesionales de excelencia capaces de dar respuesta a los desafíos de futuro, con un modelo educativo cuyo propósito es la formación integral del estudiante a partir de su realidad y sus potencialidades, promoviendo la movilidad social y la realización personal.
- Fomentar la generación de conocimiento avanzado mediante la realización y la integración de actividades de formación de postgrado e investigación fundamental, aplicada y de desarrollo, vinculadas con el sector productivo, orientadas a áreas estratégicas regionales y nacionales.
- Contribuir al desarrollo armónico y sustentable de la Región del Biobío, a través de la aplicación del conocimiento, formación continua y extensión, contribuyendo a la innovación, productividad y competitividad de organizaciones, ampliando el capital cultural de las personas, actuando de manera interactiva con el entorno y procurando la igualdad de oportunidades.
- Desarrollar una gestión académica y administrativa moderna, eficiente, eficaz y oportuna, centrada en el estudiante, con estándares de calidad certificada que le permiten destacarse a nivel nacional y avanzar en la internacionalización.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera Ingeniería Civil en Informática

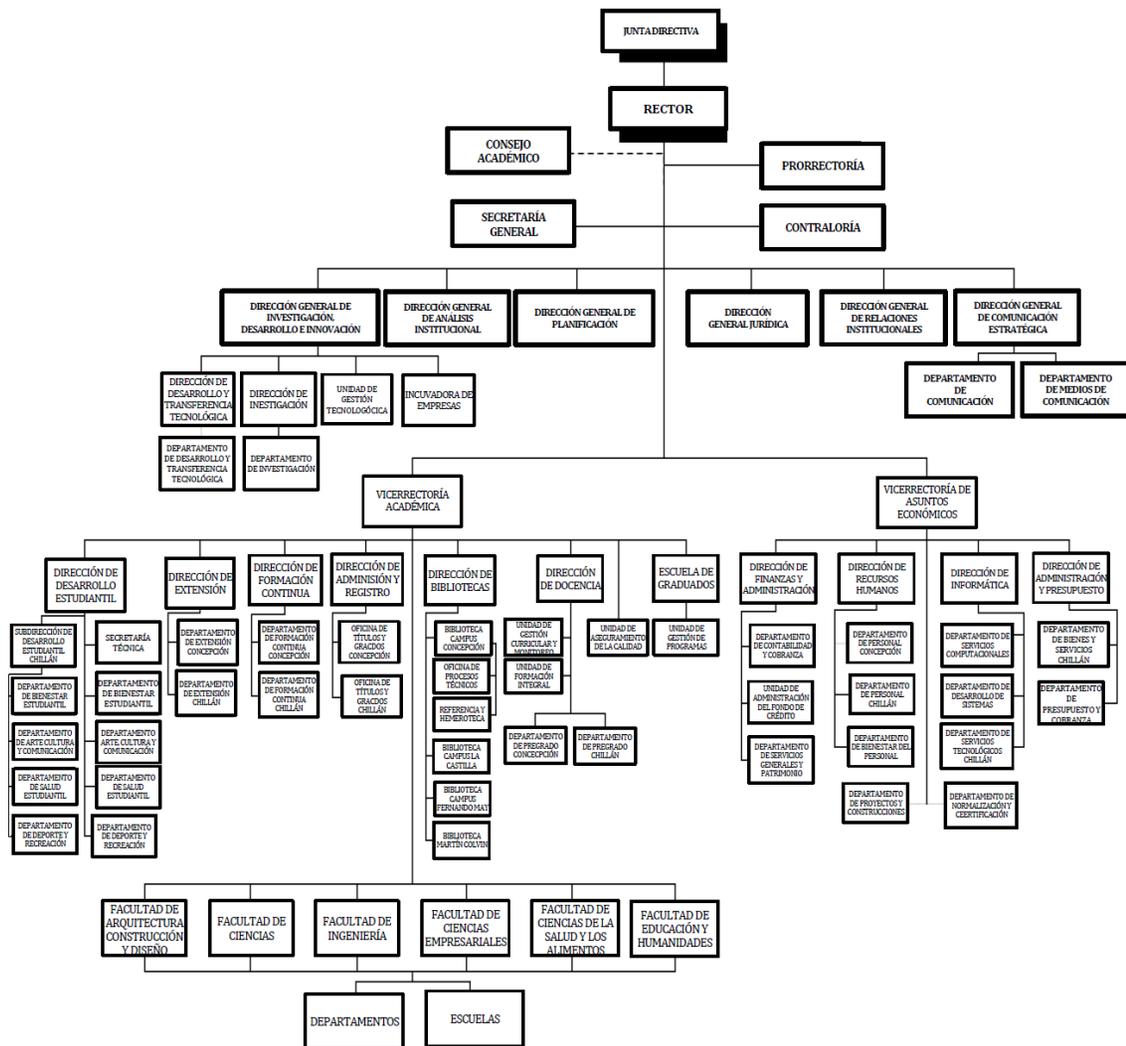


Figura 1. Estructura Organizacional Universidad del Bío-Bío

### 1.2.2 Facultad de Ciencias Empresariales

Creada en 1989, la Facultad de Ciencias Empresariales tiene como objetivo prioritario el cultivo de las disciplinas de Administración, Auditoría, Finanzas, Computación e Informática, enfatizando como áreas de estudio el Desarrollo Regional, la Pequeña y Mediana Empresa, la Planificación y el Control de Gestión Estratégico, Política de Negocios y las Tecnologías de Información y Gestión Informática.

Su permanente búsqueda de este objetivo se manifiesta no sólo en la docencia regular que imparte, sino también en la creación de programas de postítulo, el desarrollo de proyectos de investigación y en la realización de actividades de extensión, asistencia técnica y capacitación en las áreas recién mencionadas.

La Facultad se preocupa también de desarrollar la capacidad emprendedora de sus estudiantes y de enriquecer su proceso de formación con una visión humanista e integradora de su futuro quehacer profesional. La Facultad de Ciencias Empresariales cuenta con tres Departamentos en la Sede Concepción: Administración y Auditoría, Economía y Finanzas, Sistemas de Información; y dos en Chillán: Departamento de Gestión Empresarial y Departamento de Ciencias de la Computación y Tecnología de Información.

#### **1.2.2.1 Visión**

Ser una facultad destacada por su excelencia ACADÉMICA en el ámbito de las ciencias empresariales e informática, altamente comprometida con los requerimientos de la región.

#### **1.2.2.2 Misión**

Somos una facultad comprometida en la formación de profesionales de excelencia y en la creación y transmisión de conocimientos que sean un aporte de calidad para el desarrollo regional, basándonos en el cultivo de las áreas disciplinarias de gestión e informática.

Nuestra responsabilidad es ser un referente válido en la renovación permanente del conocimiento, por medio de la excelencia académica, respondiendo apropiadamente a las inquietudes del medio empresarial y de la sociedad en general, en las disciplinas que nos son propias.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

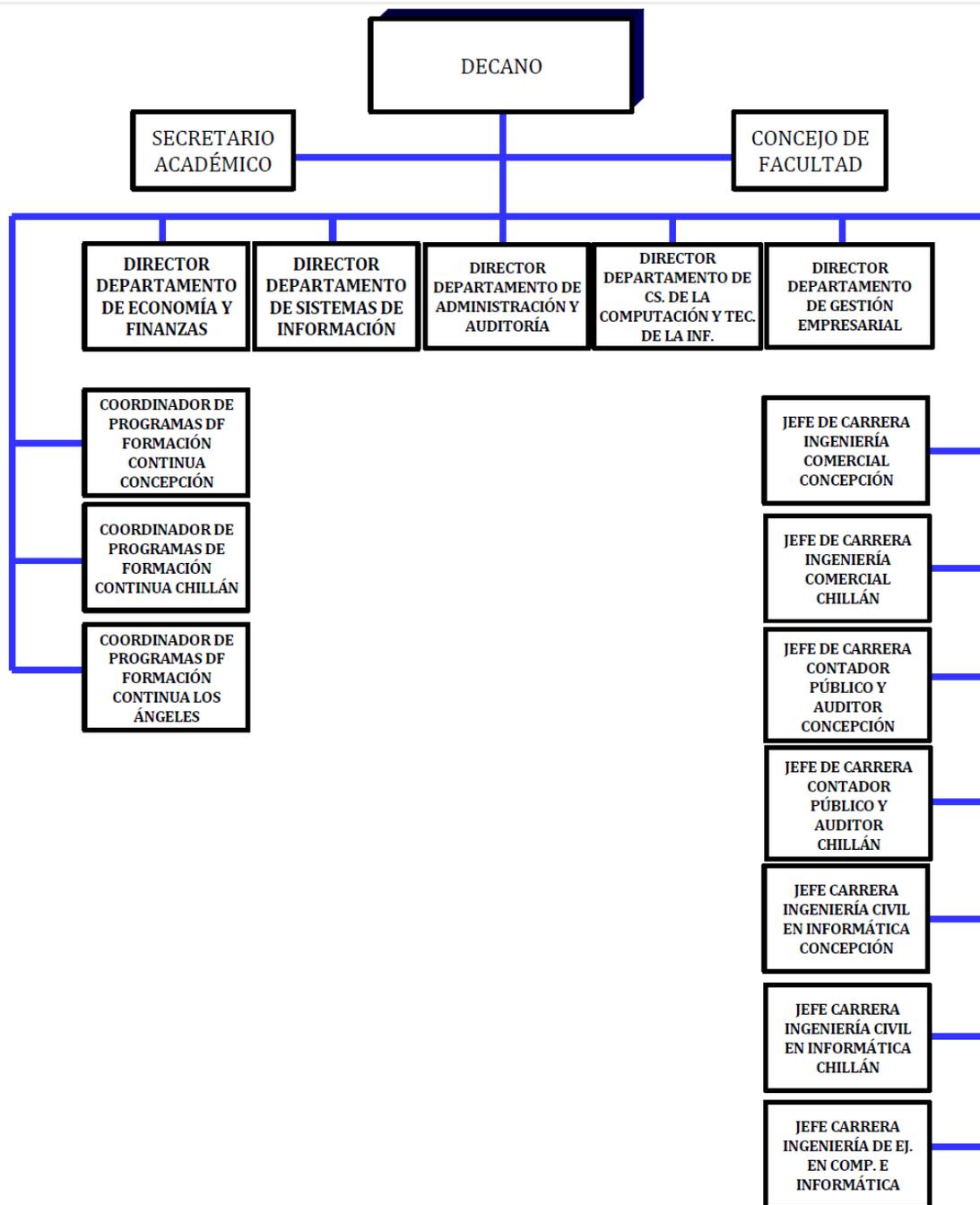


Figura 2 Estructura Organizacional Facultad de Ciencias Empresariales

## 1.2.3 Descripción del área de estudio

### 1.2.3.1 Carrera de ICI

En el año 1989, el Departamento de Sistemas de Información, perteneciente a la Facultad de Ciencias Empresariales, crea la carrera de ICI a través del decreto N°442-1 con fecha del 18 de Octubre de 1989. Con el decreto N° 456, del mismo año, se otorga el grado de Licenciado en Ciencias de la Informática para todos los estudiantes que cumplieran los requisitos establecidos en el decreto N°080-3 de 1989, esto último, corresponde a la aprobación de las asignaturas de los ocho primeros semestres del plan de estudio de la carrera.

Al momento de la creación de la carrera se define como perfil de egreso: un profesional que amalgama, junto a los conocimientos propios de las ciencias básicas y de la Ingeniería, aquellos provenientes de tres grandes áreas: Ciencias de la Computación, Sistemas de Información y Sistemas de Computación.

En el decreto N° 932 del 19 de julio de 1993, el plan de estudios es modificado con el fin de adecuar planes a los nuevos requerimientos de los profesionales en informática, se realizan cambios en algunas asignaturas de los dos últimos años de la carrera, destacando la incorporación de asignaturas electivas y el proyecto de título, que antecede a la actividad final de titulación, la Habilitación Profesional.

En el año 2003, el Departamento de Sistemas de Información se adjudica el proyecto Mecesup UBB0305 “Educación Centrada en el Estudiante: Innovación en el Proceso de Enseñanza/Aprendizaje de las Carreras Informáticas en la Universidad del Bío-Bío”. Este proyecto constituyó la base para el programa de mejoramiento de la carrera de ICI.

Por otra parte, debido al crecimiento de la planta académica de Chillán y a la creciente competencia en la provincia de Ñuble, el 13 de julio de 2005 el Consejo Académico aprueba apertura de cupos en la sede Chillán para el año 2006, lo que consta en certificado C/A N° 17/2005.

La actual malla se establece en Noviembre del año 2005, la que fue aprobada por el Consejo Académico de la Universidad según consta en el certificado C/A N°43/2005 y también por la Honorable Junta Directiva según da cuenta el certificado J/D N°25/2005.

El Decreto N° 243 del 3 de marzo de 2009 indica lo siguiente:

1. Se modifica el Plan de Estudios que fue establecido en el Decreto N° 932, del 19 de Julio de 1993, fijándose la malla nueva.
2. Se modifica el grado Académico de Licenciado en Ciencias de la Informática, por el actual grado de Licenciado en Ciencias de la Ingeniería.
3. Se mantiene el nombre del Título Profesional del antiguo Plan, que corresponde a “Ingeniero Civil en Informática”.

Los motivos que justificaron este cambio, los cuales se presentan en el Proyecto Modificación de la Carrera de ICI, se indican a continuación:

- Incorporar un enfoque por competencias.
- Cambiar desde un proceso de enseñanza centrado en el profesor hacia un proceso de aprendizaje centrado en el estudiante.
- Otorgar el grado de Licenciado en Ciencias de la Ingeniería, lo cual implicaba potenciar las asignaturas de las áreas de ciencias básicas y de ciencias de la ingeniería.
- Actualizar los contenidos (conocimientos, habilidades y actitudes) del plan curricular en el área profesional. Cabe recordar que el plan vigente a esa fecha fue definido en 1989.
- Potenciar el área de formación general aumentando el número de créditos.
- Incorporar el idioma Inglés a fin de desarrollar en los estudiantes las 4 habilidades (escuchar, hablar, escribir y leer).
- Fortalecer las habilidades comunicativas en la lengua materna.
- Incorporar asignaturas integradoras de contenidos principalmente prácticas (talleres).
- Potenciar la vinculación con el medio, esto mediante prácticas profesionales y actividades prácticas en las asignaturas.
- Disminuir la duración de la carrera de 6 a 5 años.

En el año 2006 ingresan los primeros estudiantes al actual Plan de Estudios y se espera que en el año 2010 egresen los primeros profesionales. No obstante, ya se cuenta con algunos egresados de este plan curricular a partir de 2008, dado que en el año 2006 se trasladaron estudiantes desde el antiguo plan de estudios al actual.

En la sede de Concepción se trasladaron 99 estudiantes al actual plan en Marzo de 2006.

En Chillán, por su parte, 6 estudiantes de la carrera de Ing. de Ejecución en Computación e Informática se trasladaron a la carrera de ICI, de tal suerte que en 2010 un estudiante se encuentra realizando únicamente su actividad de titulación, y los restantes se encuentran cursando el quinto año de la carrera.

Actualmente la carrera de ICI sede Concepción cuenta con 324 estudiantes y la sede Chillán cuenta con 255 estudiantes, dando un total de 604 estudiantes a la fecha de la realización de este proyecto.

### **1.2.3.2 Cargos Administrativos**

- Jefe de Carrera Concepción: Karina Rojas Contreras
- Jefe de Carrera Chillán: Luis Gajardo Díaz
- Secretaria Concepción: Andrea Vidal Riveros

- Secretaria Chillán: Yaquelin Badillo Fuentes

### 1.2.3.3 Visión

Ser reconocida por su excelencia, por implementar un proceso de aprendizaje centrado en el estudiante que forma profesionales competentes técnica, humana y socialmente, con una gestión curricular efectiva y eficiente que fomenta el emprendimiento, la creatividad y la innovación en los estudiantes.

### 1.2.3.4 Misión

Formar profesionales competentes e integrales capaces de gestionar proyectos en el ámbito de la informática a fin de resolver problemas del medio empresarial y de la sociedad a través del uso de tecnologías de información, contribuyendo al desarrollo del país y de la región. Asimismo, ofrece un espacio de desarrollo a los jóvenes posibilitando la movilidad social.

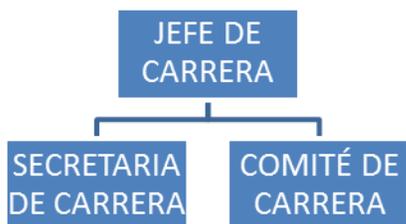


Figura 3 Estructura Organizacional Ingeniería Civil en Informática

## 1.3 Descripción del problema

El creciente número de estudiantes provenientes de diversos lugares del país produce que exista gran cantidad de datos referentes a los domicilios de estos, ya sea dirección de residencia durante periodo académico o dirección de procedencia.

Esto produce actualmente la posibilidad de una interpretación errónea o demora en el análisis de la información, llevando en algunas ocasiones al desconocimiento total sobre la información contenida en un set de datos. Ya que al tratar de realizar interconexiones sobre los domicilios almacenados no hay garantías de que quien analice la información pueda obtener una interpretación correcta.

Por interpretación correcta entiéndase a quien revise un grupo de direcciones, no tiene como saber si estas son próximas entre sí, podrían pertenecer a un sector común; sin necesidad de contar con conocimientos geográficos.

Esta problemática en el análisis de los datos puede producir ante catástrofes incertidumbre sobre la existencia de estudiantes que pudieran ser afectados, debido a que estén ubicados en las zonas afectadas.

Además, llegado su momento, puede existir la necesidad de conocer la identidad de aquellos estudiantes que pertenezcan a una determinada ciudad o sector y confirmar con ello si existe una tendencia desde alguna zona por elegir alguna sede.

## 1.4 Solución planteada

Un sistema de gestión de direcciones permitiría a los funcionarios de la Universidad del Bío-Bío mantener información sobre los domicilios de sus estudiantes. La integración con un sistema de mapa, permitiría una rápida visualización e interpretación de estos datos, otorgando la capacidad de analizar de manera simple.

Dando también la posibilidad de simplificar el análisis en caso de observar un grupo de direcciones, obteniendo de manera rápida si estas se encuentran cercas entre sí. Junto con la posibilidad de obtener reportes que permitan reconocer sobre un mapa los estudiantes que pertenezcan a un determinado sector.

Además, en caso de algún evento catastrófico, el sistema posibilitaría el registro del suceso y de los alumnos afectados por este.

## 1.5 Objetivos

### 1.5.1 General

Desarrollar bajo tecnología web, una plataforma que permita la visualización de las distintas direcciones de los estudiantes de la carrera de Ingeniería Civil en Informática.

Facilitar la visualización e interpretación en un mapa, el lugar de residencia de los estudiantes de la carrera.

Registrar diferentes eventos catastróficos y los alumnos pudieran verse afectados.

### 1.5.2 Específicos

1. Desarrollar el marco teórico sobre sistemas de información de geolocalización y manejo de mapas.
2. Aplicar conceptos de usabilidad en el diseño de interfaces con mapas.
3. Selecciona una plataforma para el manejo de información geolocalizada.
4. Desarrollar un módulo para visualizar lugares de residencia de los estudiantes sobre un mapa.
5. Desarrollar un módulo que permita la mantención de los datos de los estudiantes.
6. Desarrollar un módulo para la obtención de diferentes reportes relacionados con la ubicación de los estudiantes

## 1.6 Alcances y limitaciones

La aplicación a desarrollar permitirá mostrar, sobre un mapa, las ubicaciones disponibles tanto para un estudiante específico como para un grupo de ellos y utilizando esta información se podrá entregar al usuario reportes filtrando los estudiantes por carrera y sede, además obtener el detalle de aquellos estudiantes que pertenezcan a un determinado sector, en este caso, una comuna o una región en particular.

Debido a la naturaleza privada de los datos, esta solo podrá ser usada por algunos funcionarios de la Universidad del Bío-Bío, específicamente de la carrera de Ingeniería Civil en Informática.

Este proyecto contempla el uso de plataforma web y acceso controlado mediante credenciales de usuario.

## 1.7 Definiciones, Abreviaciones y Siglas

**API:** Del inglés, Application Programming Interface. Es el conjunto de funciones y métodos que ofrece cierta biblioteca para ser usado por otro software.

**SIG:** Sistema de Información Geográfica. Es un conjunto de herramientas que permiten la manipulación de datos espaciales.

**OGC:** Del inglés, Open Geospatial Consortium. Su fin es definir estándares abiertos e interoperables dentro de los SIGs.

**GPS:** Es un sistema que está constituido por 24 satélites y utiliza la triangulación para determinar en todo el globo la posición de un objeto, persona o vehículo.

**UML:** Del inglés Unified Modeling Language. Es el lenguaje de modelado de sistemas software más conocido y utilizado en la actualidad. Es un "lenguaje de modelado" para especificar o para describir métodos o procesos.

**Google Maps:** Es un servidor de aplicaciones de mapas en la Web.

**Yii:** Del inglés "Yes it is". Es un framework orientado a objetos, software libre, de alto rendimiento basado en componentes, PHP y framework de aplicaciones web.

**JavaScript:** Es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

**MVC:** Modelo Vista-Controlador. Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones

**GUI:** Del inglés Graphic User Interface. Es el conjunto de formas y métodos que posibilitan la interacción de un sistema con los usuarios utilizando formas gráficas e imágenes

**Open Source:** "Código Abierto", es la expresión con la que se conoce al software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

**Front Controller:** Patrón de diseño para el manejo de peticiones de la capa de presentación, en la que esta debe controlar y coordinar el procesamiento de todos los usuarios a través de varias peticiones.

**GML:** Del inglés Geography Markup Language (Lenguaje de Marcado Geográfico). Es un sublenguaje de XML descrito como una gramática en XML Schema para el modelaje, transporte y almacenamiento de información geográfica.

**Framework:** es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje

interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**XMLHttpRequest:** (XHR), también referida como XMLHTTP (Extensible Markup Language / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web.

**Ajax:** Es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

**PNG:** es un formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps, desarrollado para almacenar imágenes con una mayor profundidad de contraste y otros importantes datos.

**jQuery:** Es una bibliotecas que ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

**Geocoder:** Es cualquier aplicación o servicio utilizado en el proceso de geocodificación.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

# Marco Teórico

## 2 MARCO TEÓRICO

---

### 2.1 Introducción

El uso de información geográfica siempre ha estado presente en nuestra historia, desde los primeros hombres para identificar rutas de migración para sus cacerías [26] hasta el caso del doctor John Snow, el cual utilizó mapas para identificar zonas de brotes de cólera [27]. En este último caso no solo representaba la realidad sino que por primera vez analizaba conjuntos de fenómenos geográficos dependientes.

En el desarrollo de este proyecto no solo se pretende mostrar una realidad sino que además enriquecer esta con información. Para ello es fundamental entender cómo funcionan los sistemas de información geográfica y cómo implementar estos en una aplicación web.

A continuación, se darán a conocer algunos conceptos importantes relacionados tanto con los sistemas de georreferencia como con las herramientas empleadas para el llevar a cabo el proyecto en sí, incluyendo además, la metodología de desarrollo empleada y el entorno en que será instalado.

## 2.2 Sistema de Información Geográfica

### 2.2.1 Definición y Funcionamiento

Un Sistema de Información Geográfica (SIG) es una herramienta de análisis de información, una integración organizada de hardware, software, procedimientos y datos geográficos, diseñado para capturar, administrar, analizar, modelar y graficar datos y objetos espacialmente referenciados [1].

El SIG funciona como una base de datos con información geográfica, asociada por un código común a los objetos gráficos de un mapa. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente. (Véase Figura 4).

Un SIG permite la gestión de información espacial, 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 podría obtener de otra forma [2].

Las principales necesidades que puede resolver un Sistema de Información Geográfica, ordenadas de menor a mayor complejidad [3], son:

- Localización: preguntar por las características de un lugar concreto.
- Condición: el cumplimiento o no de unas condiciones impuestas al sistema.
- Tendencia: comparación entre situaciones temporales o espaciales distintas de alguna característica.
- Rutas: cálculo de rutas óptimas entre dos o más puntos.
- Pautas: detección de pautas espaciales.
- Modelos: generación de modelos a partir de fenómenos o actuaciones simuladas.

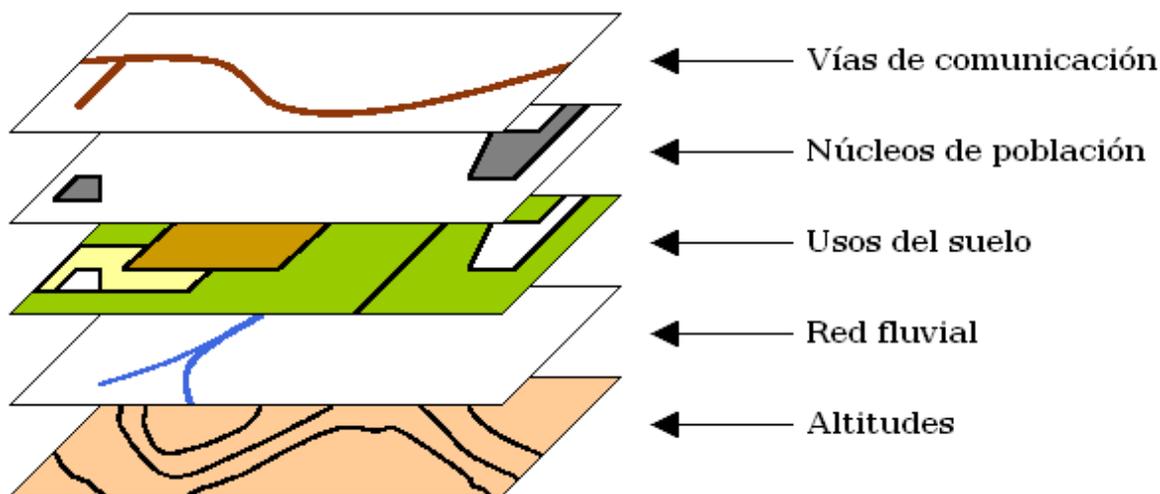


Figura 4 SIG

La información de un SIG puede mostrarse en dos formatos: *celular* o *raster* y vectorial.

El formato *raster* es la digitalización de los datos por medio de scanner, imágenes de satélite, fotografías, videos, etc.; el formato vectorial se representa por medio de segmentos orientados de rectas o vectores (pares ordenados de coordenadas).

Entre los análisis más importantes que un SIG debe realizar:

- Contigüidad: localizar áreas en una región determinada.
- Coincidencia: análisis de superposición de puntos.
- Enrutamiento: movimiento de un elemento a lo largo de una red.
- Radio de acción: alcance del elemento.
- Apareamiento: acople de información de direcciones con entidades gráficas.
- Análisis digital del terreno: análisis de la información de superficie para modelar fenómenos geográficos.
- Operación sobre mapas: uso de expresiones lógicas y matemáticas para el análisis y modelamiento de atributos geográficos.
- Geometría de coordenadas: operaciones para el manejo de coordenadas terrestres.

El abaratamiento y uso masivo de la tecnología GPS (Global Position System) integrada con equipos móviles (celulares, computadores, etc.), el desarrollo del internet y las redes de comunicación y los estándares OGC, han impulsado la tecnología web mapping y el surgimiento de numerosas aplicaciones que permiten la publicación de información geográfica a través de la web.

El *mapeo web* (web mapping) es el proceso de diseñar, aplicar, generar y visualizar datos geoespaciales en la world wide web. Dicha tecnología procura entregar los datos en formato GML (geographic markup lenguaje, sublenguaje para el modelamiento, transporte y almacenamiento de información geográfica), de acuerdo con las especificaciones del Open Geospatial Consortium (OGC, grupo de organizaciones públicas y privadas creado en 1994), con el fin de conseguir una interoperación de los datos espaciales. Aunque el GML se presenta como el futuro lenguaje web para la visualización de mapas, en la actualidad todavía no es utilizado por la mayoría de los generadores de mapas.

## 2.3 Web Mapping

En forma general, Web Mapping es el término utilizado para referirse a la visualización de datos espaciales a través de internet. Neumann [4] describe el Web Mapping como “el proceso de diseñar, implementar, generar y entregar mapas en el *World Wide Web*” y lo diferencia tanto de los SIGs como del termino Web GIS o Web SIG aunque muchas veces se utilizan como sinónimos.

Las aplicaciones de Web Mapping permiten mostrar en internet no sólo de forma estática, como podría ser una carta cartográfica escaneada, sino que también hacerlos dinámicos, permitiendo así la interacción del usuario. Se destacan entre sus funciones:

- Generación de mapas con todos los elementos requeridos incluyendo herramientas para la navegación (acercar, alejar, etc.).
- Superposición de capas de información.
- Despliegue de información descriptiva de los elementos del mapa.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

- Ejecución de consultas de tipo espacial.
- Interacción con bases de datos.

El Web Mapping brinda algunas ventajas como la posibilidad de compartir e intercambiar información con un amplio público, acceso a herramientas para análisis y toma de decisiones, actualización continua de información, facilidad para actualizar las aplicaciones, entre otras.

### **2.3.1 Api de Mapas online**

Pese a la gran utilidad que presentan los sistemas de información geográfica, por lo general estos tienen licencia comercial, por lo que una conveniente alternativa dentro de los Web Mappings son los servicios de mapas online, también conocidos como Web Mapping Services, tal como Google Maps [5]. Si bien no es posible considerar los apis de mapas como un SIG, es válido para construir un sistema de geo localización.

Se puede expresar que una API (Application Programming Interface) es una interfaz de programación de aplicaciones y está compuesta por funciones y procedimientos (o métodos, en la programación orientada a objetos) incluidos en cierta biblioteca y que pueden ser utilizados por otro software en un capa de abstracción.

### 2.3.1.1 Api de Google Maps

Google Maps, durante una época llamado Google Local, es un servidor de aplicaciones de mapas en web desarrollado con la tecnología de Google.

En sus inicios, ofrecía mapas de ciudades de diversos países (EEUU, Canadá, Reino Unido, Japón), en principio como complemento y ayuda al usuario que realiza búsquedas en Google Local.

Google Maps fue anunciado por primera vez en el Google Blog el 8 de Febrero de 2005. Originariamente era sólo soportado por los navegadores Internet Explorer y Mozilla Firefox. El soporte para Opera y Safari fue agregado el 25 de Febrero de ese mismo año. El software estuvo en su fase beta durante 6 meses antes de convertirse en parte de Google Local, el 6 de Octubre de 2005

En Abril de 2005 comenzó a ofrecer además, imágenes vía satélite provenientes de la firma Keyhole, adquirida por Google. De esta manera, podemos ver fotografías aéreas de todo el planeta de mayor o menor resolución dependiendo si se trata o no de importantes núcleos urbanos.

Todas las imágenes de satélite de GoogleMaps (Véase Figura 5) son las mismas que las que podemos encontrar en otra famosa herramienta de Google llamada GoogleEarth.

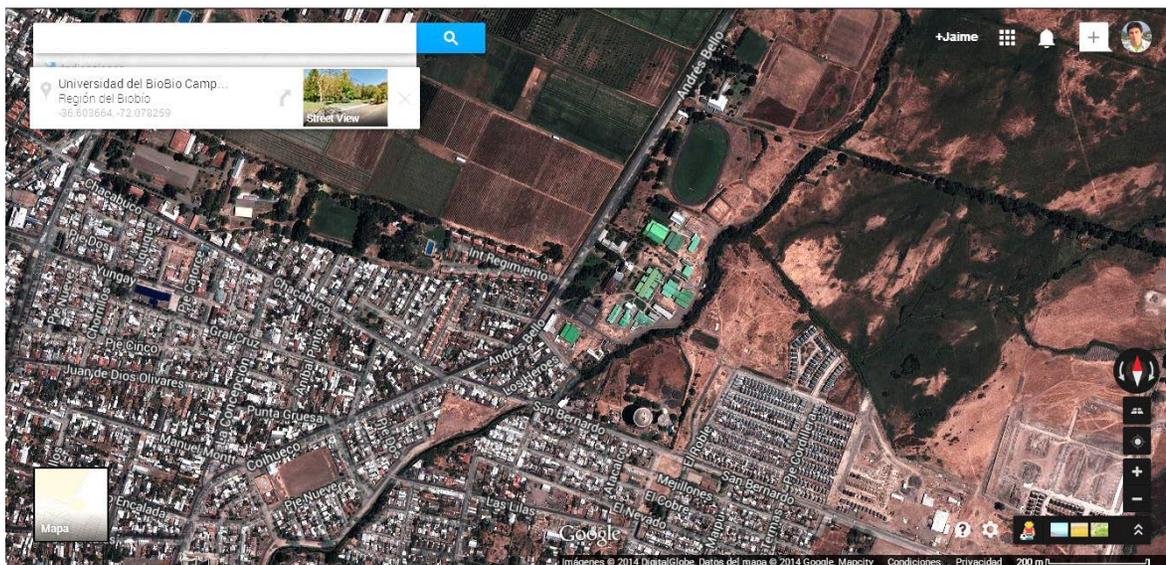


Figura 5 Vista desde Google Maps

Del mismo modo que otras aplicaciones web desarrolladas por Google, para implementar Google Maps, se usan un gran número de archivos JavaScript. Como el usuario puede mover el mapa, la visualización del mismo se descarga desde el servidor.

Cuando este busca un punto determinado, la ubicación está marcada por un indicador en forma de pin, el cual es una imagen PNG transparente sobre el mapa.

Para conseguir la conectividad sin sincronía con el servidor, y así proporcionar al usuario mayor interactividad con el mapa, mediante la realización de peticiones asíncronas a la red con JavaScript y *XMLHttpRequest*, se usa una técnica conocida como *AJAX*.

Además, en Junio de 2005 se presentó la API de Google Maps que permite a los programadores crear sus propias aplicaciones web utilizando las imágenes y mapas de este servicio.

Para implantar esta tecnología en una página web, Google proporciona una clave “single Maps API key”, que es válida para un único directorio o dominio. Para obtener esta clave se debe tener una cuenta de Google, y la clave que se proporciona estará conectada a dicha cuenta.

Google ofrece dos tipos de documentación: Una de ellas, está diseñada para permitir al nuevo usuario, empezar rápidamente a experimentar y desarrollar sencillas aplicaciones con Google Maps API. El otro tipo de documentación que se ofrece, se trata de una guía de referencia completa y exhaustiva: Google Maps API Reference. Recientemente esta documentación se ha reorganizado para ofrecer una información más conceptual y así focalizar las posibles discusiones en las distintas áreas dominantes.

### 2.3.1.2 Utilidades de GoogleMaps

GoogleMaps permite a los desarrolladores incorporar elementos dentro de sus mapas, con el fin de darle una mayor diversificación al uso de estos.

Entre la gran lista de utilidades que se puede encontrar en la documentación de referencia de GoogleMaps<sup>1</sup>, las más importantes desde la perspectiva del desarrollo del presente proyecto, está la posibilidad de incorporar información dentro del mapa, la manipulación por parte del usuario y la visualización de elementos dentro de este. A continuación se listaran aquellos que serán necesarios para nuestro sistema:

- Map: Crea un mapa dentro de la página.
- Marker: Crea un marcador con las opciones especificadas. Si se especifica un mapa, se añadidira el marcador en el mapa.
- Marker Cluster: Dependiendo del nivel de acercamiento y el número de marcadores, es posible agruparlos, visualizando una marca circular indicando el número de marcadores en esa zona del mapa.
- Polygon: Dada una lista de coordenadas, se dibuja un polígono en el mapa conformado por la unión de cada punto y en el orden de la lista.
- Geocoder: Servicio que permite convertir, ya sea una coordenada en una dirección estandarizada o viceversa.
- InfoWindow: Permite asociar a un marcador una ventana con información en su interior.
- ZoomControl: Integra una control de zoom en el mapa, el cual posibilita, tanto acercar como alejar una zona en el mapa.
- Controls: Control que permite al usuario navegar por el mapa.

---

<sup>1</sup> <https://developers.google.com/maps/documentation/javascript/reference>

### 2.3.2 Metodología de Desarrollo

En el presente proyecto, dada su naturaleza evolutiva a través del tiempo, se utilizará el modelo iterativo incremental. Modelo de proceso que permite desarrollar versiones cada vez más completas mediante la construcción de prototipos.

Esta metodología permite en cada incremento identificar a grandes rasgos cuáles serán los servicios más y menos importantes que proporcionará el sistema. Por lo que cada incremento proporciona un subconjunto de funcionalidades que tendrá el sistema una vez finalizado. Ver Figura 6

Es posible identificar varias ventajas en el modelo iterativo incremental [19], según Sommerville.

1. Los clientes no tienen que esperar hasta que el sistema completo se entregue para sacar provecho de él. El primer incremento satisface los requerimientos más críticos de tal forma que pueden utilizar el software inmediatamente.
2. Los clientes pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
3. Existe un bajo riesgo de un fallo total del proyecto. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue de forma satisfactoria al cliente.
4. Puesto que los servicios de más alta prioridad se entregan primero, y los incrementos posteriores se integran en ellos, es inevitable que los servicios más importantes del sistema sean a los que se les hagan más pruebas. Esto significa que es menos probable que los clientes encuentren fallos de funcionamiento del software en las partes más importantes del sistema

Se justifica el uso del modelo iterativo incremental en este proyecto, porque así se puede ir revisando con cada incremento fallas anteriores que tenga este proyecto, y arreglarlas a tiempo antes de la fecha de entrega del proyecto.

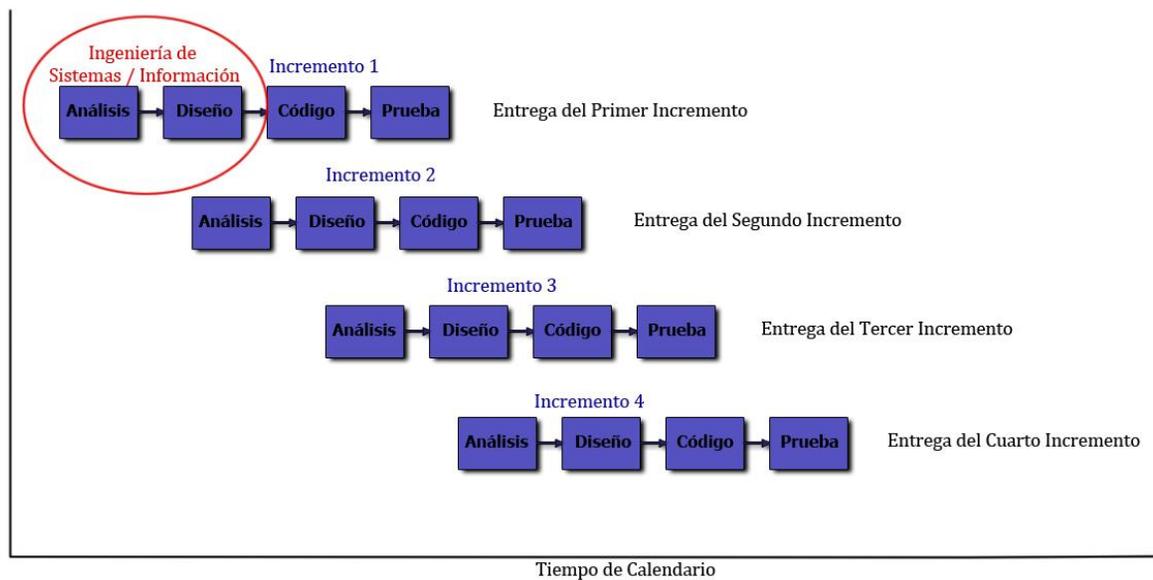


Figura 6 Modelo Incremental de Larman

### 2.3.3 Modelado

“Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir el modelo del sistema, incluyendo aspectos conceptuales tales como procesos y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.” [20]

UML es un sistema de notaciones destinado a los sistemas de modelado que utilizan conceptos orientados a objeto. Cabe mencionar, que es un conjunto de herramientas para construir y leer modelos, pero no orienta a cómo crearlos, lo que lo hace independiente a la metodología de desarrollo adoptada.

UML consta de tres categorías de diagramas (Ver Figura 7), los cuales son:

- Diagramas de Estructura, enfatiza en los elementos que deben existir en el sistema modelado.
- Diagramas de Comportamiento, enfatiza en lo que debe suceder en el sistema modelado.
- Diagramas de Interacción, Corresponde a un subtipo de diagramas de comportamiento, que se enfatiza sobre el flujo de control y datos entre los elementos del sistema modelado.

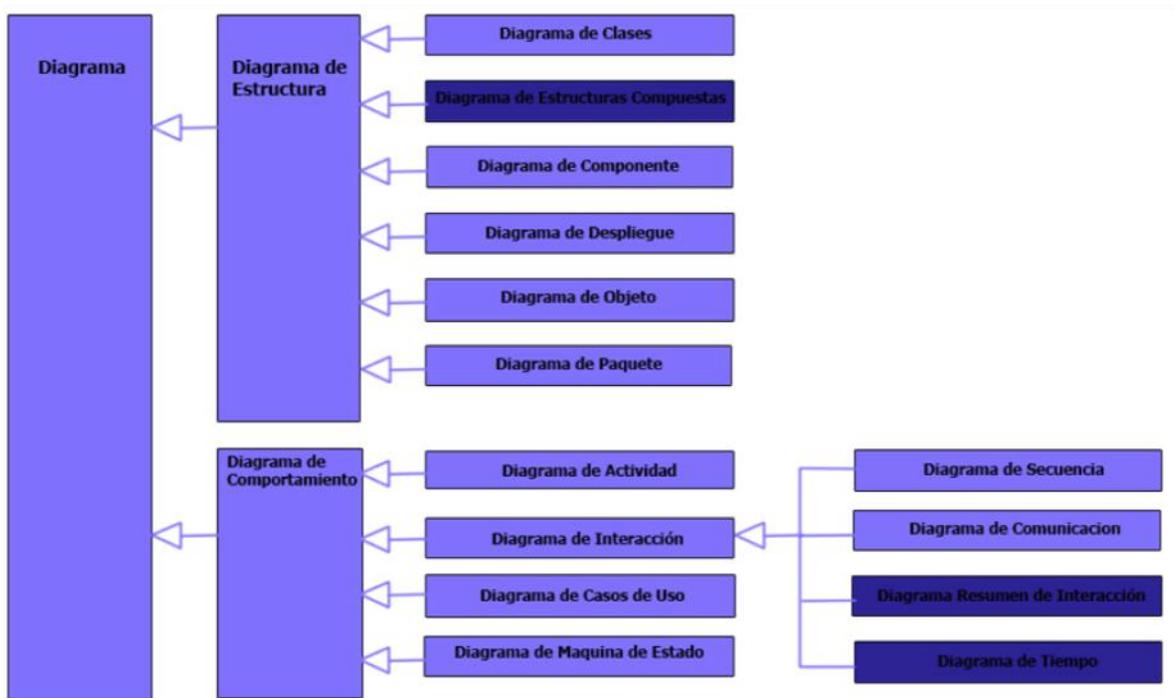


Figura 7 UML, Esquema conceptual de diagramas

## 2.4 Enfoque Orientado a Objetos

La orientación a objetos es un paradigma más de programación en el que un sistema se expresa como un conjunto de objetos que interactúan entre sí. Más que describir cada objeto individualmente se describe un conjunto de objetos que poseen las mismas propiedades.

En la Programación Orientada a Objetos los programas son representados por un conjunto de objetos que interactúan. Un objeto engloba datos y operaciones sobre estos datos.

La Programación Orientada a Objetos constituye una buena opción a la hora de resolver un problema, sobre todo cuando éste es muy extenso. En este enfoque de programación, se facilita evitar la repetición de código, no sólo a través de la creación de clases que hereden propiedades y métodos de otras, sino además que el código es reutilizable por sistemas posteriores que tengan alguna similitud con los ya creados.

### 2.4.1 Conceptos básicos

- **Objeto:** Un objeto es una entidad caracterizada por sus atributos propios y cuyo comportamiento está determinado por las acciones o funciones que pueden modificarlo, así como también las acciones que requiere de otros objetos. Un objeto tiene identidad e inteligencia y constituye una unidad que oculta tanto datos como la descripción de su manipulación. Puede ser definido como una encapsulación y una abstracción: una encapsulación de atributos y servicios, y una abstracción del mundo real.
- **Clase:** La clase es la unidad de modularidad en el EOO. La tendencia natural del individuo es la de clasificar los objetos según sus características comunes (clase). Por ejemplo, las personas que asisten a la universidad se pueden clasificar (haciendo abstracción) en estudiante, docente, empleado e investigador.

La clase puede definirse como la agrupación o colección de objetos que comparten una estructura común y un comportamiento común.

- **Relación entre Clase y Objeto:** Las clases son básicamente plantillas que describen objetos. Su rol es definir nuevos tipos conformados por atributos y operaciones. A diferencia de las clases, los objetos son instancias particulares de estas. Por ello, las clases son una especie de molde de fábrica, que sirven de base para la construcción de los objetos.
- **Atributo:** Son los datos o variables que caracterizan al objeto y cuyos valores en un momento dado indican su estado.

Un atributo es una característica de un objeto. Mediante los atributos se define información oculta dentro de un objeto, la cual es manipulada solamente por los métodos definidos sobre dicho objeto.

- **Método:** Son las operaciones (acciones o funciones) que se aplican sobre los objetos y que permiten crearlos, cambiar su estado o consultar el valor de sus atributos.

- **Mensaje:** Es la petición de un objeto a otro para solicitar la ejecución de alguno de sus métodos o para obtener el valor de un atributo público.

## 2.4.2 Fundamentos del Enfoque Orientado a Objetos

El Enfoque Orientado a Objeto se basa en cinco principios que constituyen la base de todo desarrollo orientado a objetos. Estos principios son: la Abstracción, el Encapsulamiento, la Modularidad, la Herencia y el Polimorfismo.

- **Abstracción:** Es el principio de ignorar aquellos aspectos de un fenómeno observado que no son relevantes, con el objetivo de concentrarse en aquellos que si lo son. Una abstracción denota las características esenciales de un objeto (datos y operaciones), que lo distingue de otras clases de objetos. Decidir el conjunto correcto de abstracciones de un determinado dominio, es el problema central del diseño orientado a objetos.
- **Encapsulamiento** (Ocultamiento de información): Es la propiedad del EOO que permite ocultar al mundo exterior la representación interna del objeto. Esto quiere decir que el objeto puede ser utilizado, pero los datos esenciales del mismo no son conocidos fuera de él. La idea central del encapsulamiento es esconder los detalles y mostrar lo relevante.
- **Modularidad:** Es la propiedad que permite tener independencia entre las diferentes partes de un sistema. La modularidad consiste en dividir un programa en módulos o partes, que pueden ser compilados separadamente, pero que tienen conexiones con otros módulos.
- **Herencia:** Es el proceso mediante el cual un objeto de una clase adquiere propiedades definidas en otra clase que lo preceda en una jerarquía de clasificaciones. Permite la definición de un nuevo objeto a partir de otros, agregando las diferencias entre ellos (Programación Diferencial), evitando repetición de código y permitiendo la reusabilidad.

La herencia puede ser simple (cada clase tiene sólo una superclase) o múltiple (cada clase puede tener asociada varias superclases). La clase Docente y la clase Estudiante heredan las propiedades de la clase Persona (superclase, herencia simple). La clase Preparador (subclase) hereda propiedades de la clase Docente y de la clase Estudiante (herencia múltiple).

- **Polimorfismo:** Es una propiedad del EOO que permite que un método tenga múltiples implementaciones, que se seleccionan en base al tipo objeto indicado al solicitar la ejecución del método.

El polimorfismo operacional o Sobrecarga operacional permite aplicar operaciones con igual nombre a diferentes clases o están relacionados en términos de inclusión. En este tipo de polimorfismo, los métodos son interpretados en el contexto del objeto particular, ya que los métodos con nombres comunes son implementados de diferente manera dependiendo de cada clase

## 2.5 Arquitectura del Software

El Modelo Vista-Controlador [6], es una arquitectura que separa la lógica de negocio de la interfaz de usuario, facilitando la evolución por separado de ambos aspectos e incrementando la reutilización y flexibilidad. Divide una aplicación en tres módulos estos módulos son claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador. Véase Figura 88.

En esta arquitectura es posible encontrar 3 “Responsabilidades”[6] principales:

- **Controlador**, encargado de la lógica. Define que se ha de hacer y cuando.
  - Cómo se ha de responder ante un evento de la interfaz gráfica (si es que la hay).
  - Cómo se ha de responder ante un mensaje por la red.
  - Cómo se han de producir los resultados.
  - Qué vistas se han de usar.

Este conoce tanto el modelo de datos como la vista que se está usando, con esto puede orquestarlos. Siendo la vista la que notifica al controlador de algún evento, ya sea actualizando el modelo o la vista.

- **Modelo**, es la parte de la aplicación que representa los datos sobre los cuales se operará. Su complejidad dependerá de las necesidades de representación de datos. Encapsula y entrega valor agregado a los datos.

Este modelo puede ser compartido por varias aplicaciones, por lo que es independiente del controlador y de la vista.

- **Vista**, se encarga de la representación del modelo, comunicando a esta con el mundo exterior. Con esto muchas veces interactúa con el usuario, lo que hace que se confunda con la interfaz gráfica de usuario (GUI por sus siglas en inglés). Además, es la parte de la aplicación que recibe las acciones o eventos.

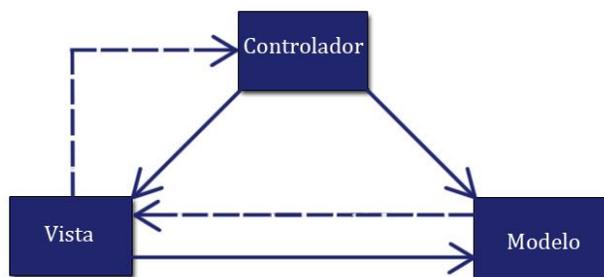


Figura 8 Diagrama Modelo Vista - Controlador

## 2.6 Tecnología para Almacenamiento Persistente

**MySQL** es un sistema de administración de base de datos relacional [7]. Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos.

Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos. MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de relacionales. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos.

## 2.7 Lenguaje de Programación

**PHP** (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje "*open source*" interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor [8].

La mayor ventaja de usar PHP es que es bastante simple para usarla en un principio y debido a que su línea de aprendizaje es rápida, es posible mejorar su implementación con características avanzadas. Aunque el desarrollo de PHP está concentrado en la programación de scripts en la parte del servidor, se puede utilizar para muchas otras cosas.

## 2.8 Framework de desarrollo

Yii [9] es un framework que comenzó su desarrollo en Enero de 2008, está basado en PHP y posee un extraordinario rendimiento comparado con otros framework para PHP. Yii incorpora el diseño modular de Joomla!, integrando **jQuery** como principal framework para JavaScript.

Yii implementa el Modelo Vista-Controlador (MV) como patrón de diseño, que es ampliamente adoptado en la programación web.

Además, de la implementación MVC, Yii también introduce un **front-controller**, llamado **application**, que encapsula el contexto de ejecución para el procesamiento de una solicitud. El application recopila cierta información acerca de una solicitud de un usuario y luego lo envía a un controlador adecuado para su posterior manipulación. Véase Figura 9.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

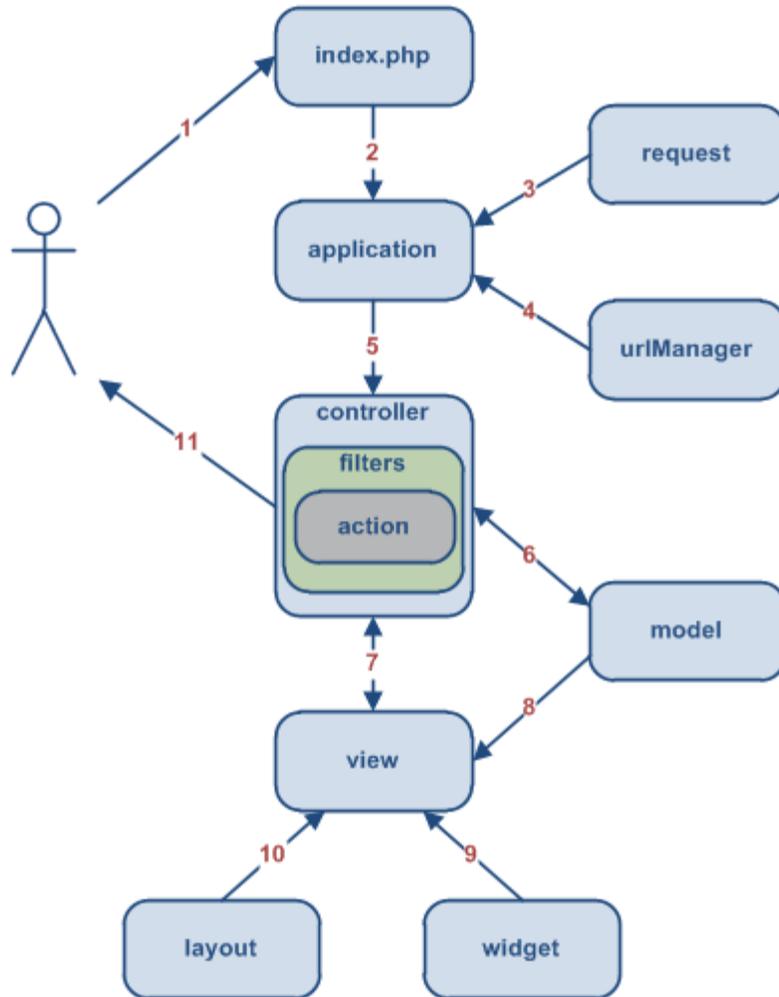


Figura 9 Típico flujo de trabajo en Yii

## 2.9 Usabilidad

La calidad de un sistema de información puede ser medida usando el concepto de usabilidad. Considerando el equilibrio entre navegabilidad, accesibilidad, productividad y optimización según sea el objetivo del negocio. Se toma en cuenta un diseño en el cual el usuario tenga una experiencia óptima y una interacción positiva.

Todo desarrollo web debe ser realizado buscando intensificar y optimizar los intercambios de valores con sus usuarios.

Jorge Pérez Ruiz, nos muestra los siguientes ítems de usabilidad [23]

- **Tiempo mínimo de descarga:**  
Consiste en la optimización de los elementos de una página y en la utilización de manera inteligente de los recursos gráficos.
- **Facilidad de Uso**  
**Accesibilidad:** Ante la universalidad de internet, el diseño debe ser flexible y accesible para todas las configuraciones de los clientes.
- **Navegabilidad:** Para el usuario debe ser fácil razonar donde se encuentra la información que busca. Se puede utilizar buscadores o navegación sencilla e intuitiva. Conseguir además una consistencia de todos los elementos y facilitar múltiples maneras y vías de acceder a la información.
- **Productividad:** Sin importar el objetivo que persiga un sitio, se debe intentar minimizar el trabajo del usuario. La arquitectura y el diseño deben ser pensados para que el usuario realice el menor número de tareas repetitivas posibles.
- **Contenido de Calidad:**  
El mayor activo de un sitio web, es el valor que aportan sus contenidos. Por lo que el principal objetivo debe ser proporcionar contenidos de calidad.
- **Actualización permanente:**  
Dado que ningún sitio es estático ni deja de evolucionar con el tiempo. De igual modo los contenidos de un sitio web necesitan de una constante adecuación a los cambios de su entorno

Además, existe un listado de principios y heurísticas de usabilidad proporcionados por Jakob Nielsen [24], el cual se listara a continuación.

1. **Visibilidad del estado del sistema:** el sistema siempre debería mantener informados a los usuarios de lo que está ocurriendo, a través de retroalimentación apropiada dentro de un tiempo razonable.

2. Relación entre el sistema y el mundo real: el sistema debería hablar el lenguaje de los usuarios mediante palabras, frases y conceptos que sean familiares al usuario, más que con términos relacionados con el sistema. Seguir las convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
3. Control y libertad del usuario: hay ocasiones en que los usuarios elegirán las funciones del sistema por error y necesitarán una “salida de emergencia” claramente marcada para dejar el estado no deseado al que accedieron, sin tener que pasar por una serie de pasos. Se deben apoyar las funciones de deshacer y rehacer.
4. Consistencia y estándares: los usuarios no deberían cuestionarse si acciones, situaciones o palabras diferentes significan en realidad la misma cosa; siga las convenciones establecidas.
5. Prevención de errores: mucho mejor que un buen diseño de mensajes de error es realizar un diseño cuidadoso que prevenga la ocurrencia de problemas.
6. Reconocimiento antes que recuerdo: se deben hacer visibles los objetos, acciones y opciones, El usuario no tendría que recordar la información que se le da en una parte del proceso, para seguir adelante. Las instrucciones para el uso del sistema deben estar a la vista o ser fácilmente recuperables cuando sea necesario.
7. Flexibilidad y eficiencia de uso: la presencia de aceleradores, que no son vistos por los usuarios novatos, puede ofrecer una interacción más rápida a los usuarios expertos que la que el sistema puede proveer a los usuarios de todo tipo. Se debe permitir que los usuarios adapte el sistema para usos frecuentes.
8. Estética y diseño minimalista: los diálogos no deben contener información que es irrelevante o poco usada. Cada unidad extra de información en un diálogo, compite con las unidades de información relevante y disminuye su visibilidad relativa.
9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores: los mensajes de error se deben entregar en un lenguaje claro y simple, indicando en forma precisa el problema y sugerir una solución constructiva al problema.
10. Ayuda y documentación: incluso en los casos en que el sistema pueda ser usado sin documentación, podría ser necesario ofrecer ayuda y documentación. Dicha información debería ser fácil de buscar, estar enfocada en las tareas del usuario, con una lista concreta de pasos a desarrollar y no ser demasiado extensa.

## 2.10 Geocodificación

### 2.10.1 Principios

Geocodificación, conocida también como dirección coincidente, es el proceso de asignar coordenadas  $x, y$  a la descripción de un lugar, en este caso la dirección. Donde las coordenadas  $x, y$  son puntos que pueden ser mostrados en un mapa [10].

Este proceso involucra varias disciplinas como teoría de estructuras de información, teoría de decisión, teoría de la probabilidad y la semántica de las palabras.

### 2.10.2 Componentes en el proceso de geocodificación

El proceso de geocodificación requiere de tres componentes básicos: **datos de referencia**, **datos descriptivos** y el **software** que realiza el proceso.

Los datos de referencias están compuestos por las diferentes fuentes de información geográfica basada en archivos, en esta categoría de información se encuentra tanto archivos codificados con coordenadas, como elementos más complejos, como son las estructuras de información espacial y la capa de líneas que representan las vías de la ciudad.

Los datos descriptivos son la información candidata al proceso de geocodificación, generalmente es la entrada del proceso.

El software es la herramienta sistémica que permite procesar la entrada y dar un resultado de acuerdo a las reglas definidas en el proceso.

### 2.10.3 Proceso de geolocalización

La geocodificación es el proceso de asignar un par de coordenadas  $X, Y$ . En general la geocodificación es el proceso que inicia con la entrada de una descripción (ej. Dirección), se identifica la semántica de la descripción para descomponerla por componentes definidos en el algoritmo de normalización, se estandariza estos componentes para unificar y homologar las estructura de los componentes y posteriormente se realiza la búsqueda en una fuente de datos para realizar la comparación por componentes y entrega el mejor candidato de la búsqueda para mostrarlo sobre un mapa.

El proceso de geocodificación se basa en el concepto de relacionamiento o vinculación de registros.

### 2.10.4 Normalización y estandarización

La normalización hace parte de los métodos de vinculación de registros. Es un componente tan especializado en los SIG, que permite normalizar nombres y direcciones especificando manualmente las reglas de transformación. Una cadena de texto es separada dentro de variables individuales, de esta manera cada variable es reconocida de acuerdo a las reglas de transformación, las cuales dependen de la semántica de las palabras a normalizar.

El proceso descrito anteriormente, obedece a un enfoque determinista, esto quiere decir que cada uno de los componentes de una dirección son conocidos y normalizados, por ejemplo, tipo de vía principal, número de la vía, número de la casa, entre otros.

La estandarización es el proceso posterior a la normalización, el cual determina la estructura fonética de las palabras y las estandariza en textos simples según las equivalencias determinadas para asignación de variables, por ejemplo, la palabra “calle” se estandariza en “CL”, la palabra “avenida” se estandariza en “AV” para posterior asignación a las variables definidas [11].

En el presente proyecto, se utilizará el API de Google Maps para el proceso de geocodificación, ya que el servicio codificación geográfica de Google proporciona una forma directa de acceder a un **geocoder** mediante solicitudes HTTP, como se puede ver en la Figura 10, es posible visualizar un punto en el mapa mediante una dirección.

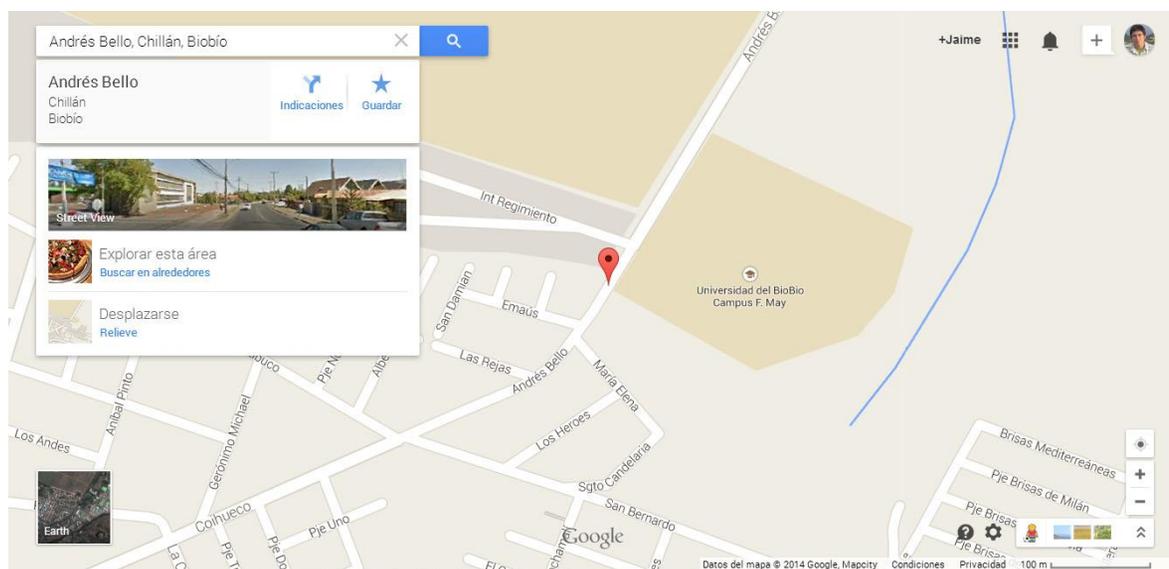


Figura 10 Geocodificación y normalización dirección Campus Fernando May

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

# ANÁLISIS DE REQUERIMIENTOS

---

## 3 ANÁLISIS DE REQUERIMIENTOS

---

### 3.1 Introducción

*“Los requerimientos son una descripción de las necesidades y deseos de un producto. La meta primaria de la fase de requerimiento es identificar y documentar lo que en realidad se necesita, en una forma que claramente se comunique al cliente y los miembros del equipo de desarrollo. El reto consiste en definirlos de manera inequívoca, de modo de detectar los riesgos y no se presenten sorpresas al momento de entregar el producto.”[20]*

Los requerimientos son el punto de acuerdo entre el cliente y el proyecto de desarrollo de un sistema, los cuales son necesarios para poder construir un software que satisfaga todas sus necesidades.

Los requerimientos sirven para describir las necesidades del cliente y describen lo que el sistema debe hacer. Por lo que en este capítulo se realizará un análisis de requerimientos de la aplicación a desarrollar, obteniendo que información y servicios se ha de incluir en este.

### 3.2 Requisitos Operacionales

Los usuarios del sistema son aquellos profesionales pertenecientes al área administrativa de la carrera de Ingeniería Civil en Informática de la Universidad del Bío-Bío.

Han sido establecidos 3 tipos de usuario, los cuales podrán interactuar con el sistema de distintas formas, de acuerdo al nivel de privilegios que cada tipo de usuario posea. Las funciones que podrán desempeñar los usuarios indicados anteriormente se detallan a continuación:

- **Administrativo:** este usuario hace referencia a las secretarías de jefatura de la carrera, teniendo como privilegios la visualización e ingreso de información referente a los estudiantes.
- **Jefatura:** este usuario estará definido por el Jefe de Carrera/Director de Escuela, teniendo este los permisos del rol administrativo además de poder editar la información de los estudiantes de la carrera.
- **Administrador:** Este usuario será quien administre los roles y usuarios del sistema.

### 3.3 Interfaz de Usuario

Debido a que este proyecto de la Universidad del Bío-Bío, se privilegiará el uso de colores y logos institucionales de la propia universidad.

Además, contará con un mapa interactivo para mejorar la experiencia del usuario al momento de ingresar la información.

### 3.4 Interfaz de Hardware

Las características mínimas que debe tener el servidor que va a alojar al sitio y a su respectiva base de datos son:

- Disco Duro: 20GB
- Procesador: Pentium 4 (R) de 1.90 GHz.
- RAM: 1 GB

### 3.5 Interfaz de Software

Los recursos de software necesarios que se utilizarán para el desarrollo de este sitio Web son:

- Servidor de aplicaciones:
  - **Apache** v2.4.3
- Base de Datos:
  - **MySQL** v5.5.27
- Interprete de Traductor:
  - **PHP** v5.4.7
- Administrador de Base de Datos:
  - **phpMyAdmin** v3.5.2.2
- Servidor FTP:
  - **FileZilla FTP Server** v0.9.37
- Cliente FTP:
  - **FileZilla Client** v3.5.1

### 3.6 Interfaces de Comunicación

Durante el periodo de desarrollo del sitio Web, se utilizará el protocolo de transferencia de datos seguro scp/ftp y sftp, para alimentar con archivos a la base de datos del servidor web. Paralelamente, se utilizará el protocolo de transferencia de hipertexto (HTTP) para la navegación y pruebas del sitio.

Una vez que el sitio esté operativo y puesto en marcha las interfaces de conexión con la base de datos y la interfaz de comunicación con el browser serán las primordiales, es decir, las conexiones realizadas por el protocolo HTTP.

### 3.7 Requerimientos Específicos

#### 3.7.1 Requerimientos Funcionales

##### 3.7.1.1 Usuario, roles y tipos

3.7.1.1.1 La administración de los usuarios y sus respectivos roles será mediante un super usuario.

3.7.1.1.2 Los tipos de roles presentes para los diferentes tipos de usuarios serán, ordenados de menor a mayor jerarquía y son los siguientes:

- **Administrativo:** Rol autorizado a visualizar e ingresar nuevos estudiantes con sus respectivas direcciones según la sede a la cual pertenezca.
- **Jefatura:** Autorizado a visualizar y obtener reportes de referente a los estudiantes de la sede de la cual está a cargo.
- **Super Usuario:** Posee control del sitio, pudiendo realizar operación de actualización, tanto de la información como de los usuarios.

3.7.1.1.3 Los tipos de usuarios que interactuarán con el sistema serán los siguientes:

- **Secretaria:** Perteneciente al área administrativa de la carrera.
- **Jefe de Carrera:** Perteneciente a la Jefatura de la carrera, tiene acceso a información de los estudiantes además de toma de decisiones.
- **Administrador:** Usuario responsable del sistema.

### 3.7.1.2 Ingreso y visualización de Estudiantes.

Como mantenedor web de información de los estudiantes de la carrera, el sistema debe permitir la creación de estudiantes, a fin de poseer un registro de los estudiantes que conforman la carrera.

Debido a la naturaleza del sistema, este registro de estudiantes solo debe contener información suficiente para identificarlos individualmente, excluyendo a ésta datos sobre rendimiento académico o de carácter personal.

A fin de mantener registro sobre el domicilio de los estudiantes de la carrera, el sistema debe permitir el ingreso de esta información.

Se debe permitir la identificación del domicilio o residencia de estudiantes, ya sea ingresando la dirección de esta o la posibilidad de ubicar está en un mapa.

### 3.7.1.3 Ingreso y visualización de Eventos.

El sistema debe permitir el registro de eventos catastróficos, posibilitando la identificación y estado de los estudiantes que se vean afectados por estos.

### 3.7.1.4 Reportes

El sistema debe permitir la generación de reportes respecto a los domicilios registrados de los estudiantes, pudiendo este reporte responder a las siguientes solicitudes:

- Informe de estudiantes de una respectiva región.
  - Indicará todos los estudiantes cuya dirección esté registrada en una determinada región del país.
- Informe de los estudiantes residentes en una determinada comuna.
  - Corresponde a los estudiantes cuya dirección pertenece a una determinada comuna.
- Informe de las direcciones de estudiantes de la una carrera.
  - Todos aquellos estudiantes que pertenecen a una determinada carrera.
- Informe de los estudiantes de una determinada sede.
  - Estudiantes que estudian en una determinada sede de la Universidad del Bío Bío

### 3.7.1.5 Sesiones

El sistema deberá permitir tanto el inicio como el término de una sesión, a fin de dar seguridad a los datos recolectados, esta sesión deberá iniciarse mediante una contraseña, todo esto dentro de un ambiente seguro, utilizando al menos un método de encriptación de acuerdo al entorno web.

### 3.7.2 Información de Entrada

En la Tabla 1 se define la interfaz para el ingreso de datos para el sistema.

Identificador	Item	Detalle
IE_01	Datos de Usuario	UserID, password, rol
IE_02	Datos del Estudiante	Rut, Nombre, Apellidos, Dirección, Carrera
IE_03	Datos de Dirección	Calle, Numero, Departamento, Ciudad, Comuna, Región

Tabla 1. Información de Entrada

### 3.7.3 Información de Reporte

Se define la interfaz con la información de salida para los diferentes reportes (Véase Tabla 2)

Identificador	Reporte	Detalle
IR_01	Lista Estudiantes por Ciudad	Ciudad, Rut, Nombre, Apellidos, Carrera
IR_02	Lista Estudiantes por Carrera	Carrera, Rut, Nombre, Apellidos, Direcciones
IR_03	Lista Estudiantes por Sector	Rut, Nombre, Apellidos
IR_04	Lista Estudiantes residentes fuera de domicilio durante periodo estudiantil	Rut, Nombre, Apellidos, Dirección

Tabla 2 Información de salida en los Reportes

## 3.8 Atributos del Producto

### 3.8.1 Usabilidad - Operatividad

Para la emisión de mensajes al usuario se crea un formato estándar, de esa forma se cumple que:

- Los mensajes de error: Cada mensaje de error que sea enviado al usuario mediante el sitio será destacado de color rojo (y degradados de éste) y advertirá con un texto explicativo de la situación que ha ocurrido.
- Los mensajes de Aviso: los mensajes de respuestas exitosas serán enviados mediante el sitio destacados con color azul (y degradados de éste), con un texto explicativo relacionado con la acción.

Todos los mensajes serán desplegados entre el borde inferior del menú y la parte superior de la zona de contenidos.

### 3.8.2 Eficiencia- Tiempo de Ejecución/Respuesta

La construcción del sitio, el uso de recursos de hardware adecuados y la disponibilidad de banda ancha en las instalaciones de servidores de la Universidad garantizan que el sitio tenga tiempos de respuesta inferiores a los 5 segundos contemplando que las condiciones del usuario son las adecuadas (que no existan problemas de conexión y no esté utilizando programas que se apoderen de todo su ancho de banda), todo esto contemplando una cantidad no superior a 50 usuarios conectados simultáneamente. Si la cantidad de usuarios conectados simultáneamente es superior a 50 (realizando tareas que impliquen agregar información a la base de datos), existe una alta

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

probabilidad que el tiempo de respuesta del sitio aumente considerablemente (de 10 o más segundos por acción).

### **3.8.3 Funcionalidad-Seguridad**

Respecto de la seguridad del sitio propiamente tal, se mantiene una estricta jerarquía de permisos de acuerdo al rol que cumple ese usuario en la Universidad. Por otro lado las passwords utilizadas son la pertenecientes al sistema institucional, por lo que solo se realiza una comparación entre el password que el usuario ingresó y la que se encuentra almacenada, ambas codificadas.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

# FACTIBILIDAD

## 4 FACTIBILIDAD

---

### 4.1 Introducción

El estudio de factibilidad permite analizar qué tan viable puede llegar a ser el desarrollo del sistema en cuestión, permitiendo tomar la decisión de si es conveniente llevarlo a cabo.

Es muy probable que la realización de un proyecto de software esté plagado de escasez de recursos y de fechas de entregas no realistas, por lo mismo es necesario evaluar la viabilidad de un proyecto, ya que gracias a éste es posible evitar meses de esfuerzos sin resultados, la pérdida de dinero y el bochorno profesional si se reconoce que el sistema es un fracaso, además ayuda a disminuir los riesgos y asegura el valor del trabajo.

En relación con la factibilidad, ésta se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos del proyecto, la factibilidad se apoya en 3 aspectos básicos que serán desarrollados en el capítulo, estos son:

- Factibilidad técnica.
- Factibilidad operativa.
- Factibilidad económica.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

## 4.2 Factibilidad Técnica

Para que el proyecto de la construcción del sitio Web se lleve a cabo se debe contar con los siguientes elementos de hardware:

- Un servidor
- Conexión a la red de internet
- Hardware de red para que el servidor tenga acceso a ésta.

Recursos de Software Necesarios

- Motor de base de datos MySQL v5.5.27
- Servidor HTTP Apache v2.4.3
- Interprete traductor PHP v5.4.7
- Software de Administración de Base de datos phpMyAdmin v3.5.2.2
- Servidor FTP

Todos los recursos de software cuentan con licencias gratuitas y se encuentran disponibles en el servidor facilitado por la Universidad.

Por parte de la Universidad se cuenta con el personal capacitado para administrar de manera correcta y eficiente los servidores.

Respecto a los conocimientos del desarrollador, estos abarcan áreas de modelado de datos, metodologías de desarrollo de software, dominio del lenguaje SQL y dominio en programación

Si bien el sistema consiste en un sitio web, no existe una experiencia en el área de diseño visual, por lo que para dar factibilidad en esta área al proyecto se contará con un diseñador gráfico que apoye en los temas visuales y relacionados con la experiencia de usuario. La Universidad cuenta con el profesional calificado para esta tarea, por lo que es posible afirmar que esta falencia será suplida.

Respecto al punto anterior, también se contará con una diseñadora gráfica con amplia experiencia en el área de “experiencia de usuario” como consultora.

En la Tabla 3 se especifican los requerimientos de hardware y software para este proyecto.

Hardware		Software	
<b>Procesador</b>	Pentium (R) Dual Core T4400 2.2 GHz	<b>Sistema Operativo</b>	Debian Server 6.0.4
<b>Memoria RAM</b>	2 GB	<b>Base de Datos</b>	MySQL v5.5.27
<b>Disco Duro</b>	10GB Disponible	<b>Servidor Web</b>	HTTP Apache v2.4.3
<b>Periféricos</b>	Mouse y teclado	<b>Entorno desarrollo</b>	Interprete traductor PHP v5.4.7
		<b>Herramienta administración de base de datos</b>	phpMyAdmin v3.5.2.2
		<b>Navegador web</b>	Mozilla 18 ó superior Chrome 18 ó superior IE8 ó superior

**Tabla 3 Requerimientos de Hardware y Software**

### 4.3 Factibilidad Operativa

El presente proyecto busca implementar una mejora tecnológica al actual sistema de manejo de información referente al domicilio de los estudiantes de la carrera.

Referente a los impactos positivos, se puede mencionar que mejorará la interpretación de la información ya disponible en cuanto a los estudiantes tanto de manera individual como en agrupaciones de esto, acercando más a una visión del mundo real.

Esta solución no solo es aplicable a los docentes y administrativos de la carrera, también es posible que esta pueda ser utilizada por el área de Bienestar Estudiantil, debido a que sus asistentes sociales deben realizar visitas al domicilio de los estudiantes de la universidad, esta herramienta podría ayudar a la planificación de estas.

### 4.4 Factibilidad Económica

En este estudio se determinan los recursos necesarios para desarrollar el proyecto y los costos en los que se debe incurrir para su fabricación, se realiza una comparación entre los costos en hardware, software y mano de obra con los beneficios que se obtendrán cuando el sistema esté en su fase de explotación.

A continuación se detallan los costos considerados en el desarrollo del proyecto, los cuales corresponden a valores de mercado, obtenidos a través de diversas consultas a personas que trabajan en el área del desarrollo de software y además consultando la página Web del Ministerio de Educación de Chile, [www.mifuturo.cl](http://www.mifuturo.cl)

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

#### 4.4.1 Costo de desarrollo

**Hardware y Software de desarrollo:** detallado en la Tabla 3, tiene un costo total de \$0, ya que el hardware se encuentra instalado y operativo hace algún tiempo en la Universidad. Respecto del software son herramientas de libre acceso sin costo. Así su costo final total es de \$0.

**Encargado del desarrollo:** para llevar a cabo el proyecto se requiere de un ingeniero civil informático.

- Costo de mercado de un analista desarrollador hora/hombre es de \$4.444 aprox.
- El trabajo se estima en un período de 4 meses y se trabajará 45 horas semanales, lo que se traduce en un total de 720 horas de elaboración del proyecto.
- El costo total del informático es de \$3.199.680

El costo total calculado anteriormente, no es considerado, debido a que el desarrollador, es un estudiante que se encuentra realizando su proyecto de título.

#### 4.4.2 Costo de Instalación

**Hardware y software del servidor:** este costo no es considerado, puesto que el servidor se encuentra en funcionamiento en la institución y cuenta con herramientas de software gratuitas.

**Redes:** Al igual que el costo anterior este no es considerado, debido a que la infraestructura de redes de datos se encuentra implementada en la institución.

#### 4.4.3 Costo de Operación

**Hardware y software:** para la operación del sistema se requerirá del Hardware y Software detallado en la tabla N°3. Como se mencionó anteriormente éste hardware y software ya se encuentra funcionando por lo que se considera costo de operación \$0.

#### 4.4.4 Costo Mantención

Este costo no será considerado, ya que una vez instalado el sistema, será responsabilidad del encargado del área de informática la mantención.

Del análisis de los costos se puede obtener el detalle de la inversión inicial (Véase Tabla 4) en que se debe incurrir para la puesta en marcha.

Inversión Inicial:

Desarrollo	\$0
Instalación	\$0
Operación	\$0
<b>Total</b>	<b>\$0</b>

**Tabla 4 Inversión Inicial**

#### 4.4.5 Beneficios Intangibles

Debido a que este proyecto no pretende mejorar ni reemplazar algún procedimiento ya existente dentro de la universidad, solamente es posible mencionar los beneficios que este generara a la institución, lo cuales se detallaran a continuación.

- Rápido acceso a la información: debido al propósito de la aplicación, toda la información referente a la residencia y contacto de un estudiante será rápidamente obtenida.
- Mejor interpretación de la información: el uso de mapas en donde se ubicaran los datos, además de la posibilidad de demarcar una zona, dan la posibilidad de que la interpretación de esta sea cercana a la realidad.
- Registro de catástrofe: ante una eventual catástrofe, se podrá registrar las zonas afectadas, además de los estudiantes afectados, pudiendo priorizar a estos últimos en caso de emergencias.
- Nueva fuente de información: la implementación generará una nueva fuente de datos, dando como beneficio la posibilidad de extraer datos para nuevos procesamiento.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

# DISEÑO

## **5 DISEÑO**

---

### **5.1 Introducción**

Después de la finalización de los documentos del análisis, es posible pasar al a fase de diseño. Durante este paso se logra una solución lógica que se funda en el paradigma orientado a objetos. Su esencia es la elaboración de diagramas de interacción, que muestran gráficamente cómo los objetos se comunicarán entre ellos a fin de cumplir con los requerimientos.

En este capítulo se muestra el diseño aplicando una serie de pasos basados en el desarrollo de software para la construcción de un sistema web, permitiendo realizar una especificación detallada de los principales aspectos que deben ser considerados en la implementación de este.

## 5.2 Diseño interfaz y navegación

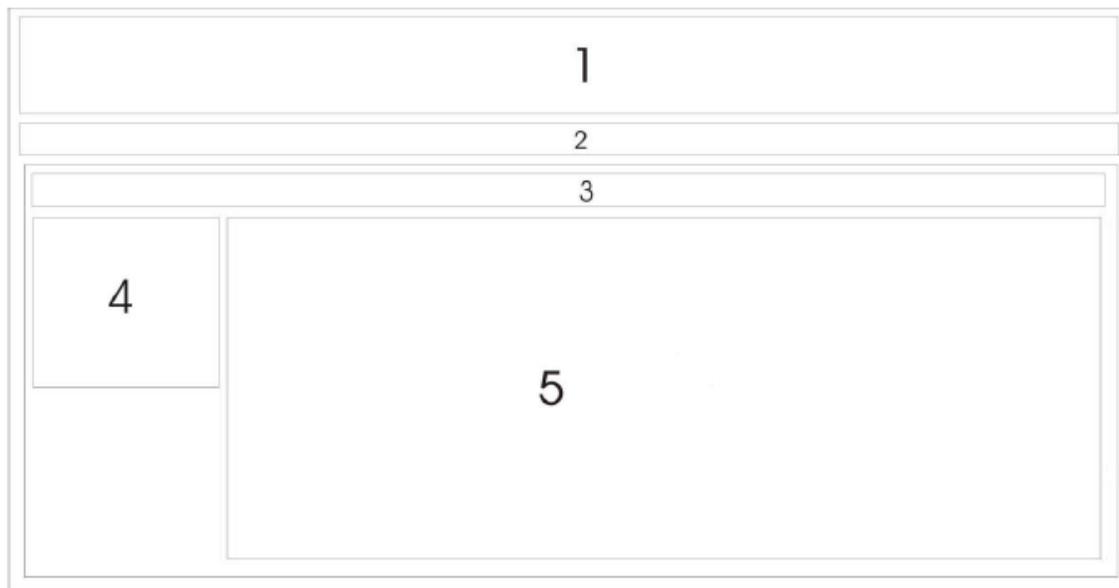


Figura 11 Diseño Interfaz Sitio Web

Como se puede visualizar en la Figura **11**, la interfaz de la aplicación se distribuirá de la siguiente manera:

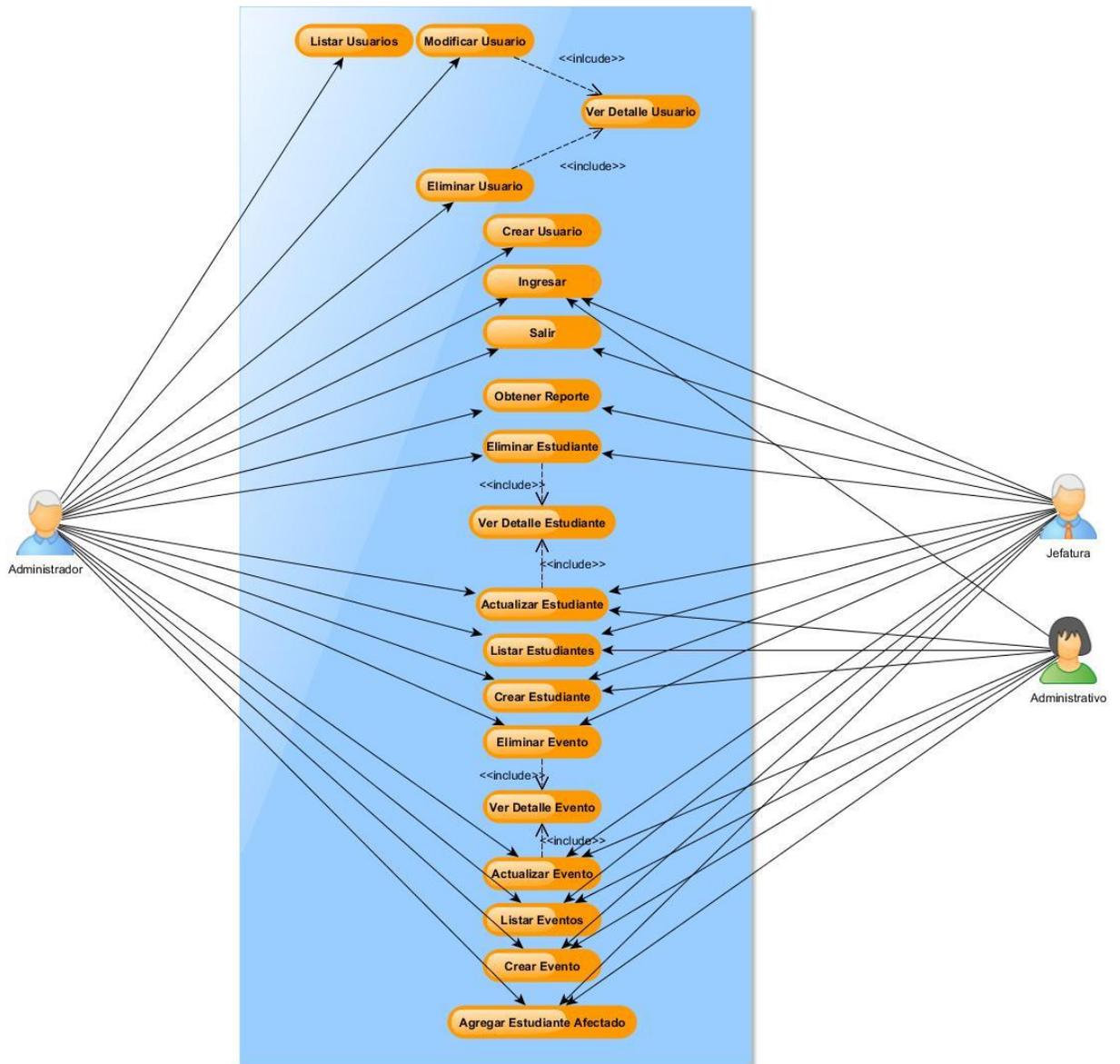
- Área 1 Banner: incluye una imagen con el nombre de la carrera, el logo de la Universidad siguiendo los colores institucionales.
- Área 2 Menú: el menú principal del sitio.
- Área 3 Navegación: sección que muestra la ruta de navegación dentro del sitio que tiene el usuario.
- Área 4 Sección de Login: cuando un usuario no ha iniciado sesión, esta área muestra un ítem para ingresar el Rut y otro para la contraseña y un botón para iniciar sesión, una vez que se ha iniciado sesión muestra un saludo con el nombre del usuario y un botón para finalizar la sesión.
- Área 5 Sección de contenido: esta sección muestra el contenido de la sección del sitio seleccionada, su interior varía de acuerdo al módulo que se esté mostrando.

### 5.3 Casos de Uso

#### 5.3.1 Diagrama de Casos de Uso

Un diagrama de casos de uso explica gráficamente un conjunto de caso de usos de un sistema, los actores y las relaciones entre éstos y los casos de uso. El diagrama tiene por objeto ofrecer una clase de diagrama contextual que nos permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan [20].

En la Figura 12 se define el diagrama de Casos de Uso para la aplicación.



Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

Figura 12 Diagrama Casos de Uso

### 5.3.2 Definición de Casos de Uso

En cada una de las especificaciones de casos de uso que a continuación se detallan se ven reflejados atributos tales como, el nombre del caso de uso, la Id, la descripción, actor del sistema, actor secundario, flujo principal, flujo alternativo, precondiciones y post-condiciones.

#### 5.3.2.1 Caso de Uso: Iniciar Sesión

En la Tabla 5 se especifica el caso de uso “Ingresar al Sistema”, donde el usuario iniciará sesión en el sistema, esto corresponde al requerimiento de uso de sesiones con el fin de dar seguridad a los datos recolectados..

Caso de Uso	Ingresar al Sistema
<b>ID</b>	<b>1</b>
<b>Descripción</b>	<b>El usuario ingresa al sistema</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>No haber iniciado sesión</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página de inicio los campos requeridos para iniciar sesión. 2.- El usuario llenará estos campos.</b>
<b>Post-condiciones</b>	<b>Se actualizará la página de inicio con los accesos directos a algunas de las principales funcionalidades del sistema y la barra de menú mostrará las opciones según el usuario ingresado.</b>

Tabla 5 Caso de Uso Iniciar Sesión

#### 5.3.2.2 Caso de Uso: Ver Listado de Estudiantes

En la Tabla 6 se especifica el caso de uso “Ver Listado de Estudiantes”, donde el usuario solicitará al sistema el listado de los estudiantes con el fin de obtener todos los estudiantes ingresados y sus direcciones, según el requerimiento relacionado con el visualización de estudiantes. .

Caso de Uso	Ver Listado de Estudiantes
<b>ID</b>	<b>2</b>
<b>Descripción</b>	<b>El usuario visualiza un listado de estudiantes y sus marcadores en un mapa</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber iniciado sesión</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la información de distintos estudiantes.</b>
<b>Post-condiciones</b>	<b>En la página estará disponible un menú de acciones y además el usuario podría ver el detalle de cada estudiante haciendo click en el rut de estos</b>

Tabla 6 Caso de Uso Listar Estudiantes

### 5.3.2.3 Caso de Uso: Ver Información Estudiante

En la Tabla 7 se especifica el caso de uso “Ver Información Estudiante”, donde el usuario visualizará el detalle de un estudiante en particular correspondiente al requerimiento de visualización de estudiantes.

Caso de Uso	Ver Información Estudiante
<b>ID</b>	<b>3</b>
<b>Descripción</b>	<b>El usuario visualizará el detalle respecto a la información de un determinado estudiante</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un usuario de la lista</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos de un determinado estudiante</b>
<b>Post-condiciones</b>	<b>En la página estará disponible un menú de acciones.</b>

Tabla 7 Caso de Uso Ver el detalle de un Estudiante

### 5.3.2.4 Caso de Uso: Actualizar Estudiante

En la Tabla 8 se especifica el caso de uso “Actualizar Estudiante”, donde el usuario modificará los datos de un estudiante, corresponde al requerimiento de ingreso de estudiantes.

Caso de Uso	Actualizar Estudiante
<b>ID</b>	<b>4</b>
<b>Descripción</b>	<b>El usuario editará los datos de un determinado estudiante</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un usuario de la lista</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos de un determinado estudiante. 2.- El usuario modificará aquellos datos que desee 3.- El usuario guardará los cambios realizados</b>
<b>Post-condiciones</b>	<b>Los datos del estudiante serán actualizados y serán vistos en la página.</b>

Tabla 8 Caso de Uso Modificar datos de un Estudiante

### 5.3.2.5 Caso de Uso: Ingresar Estudiante

En la Tabla 9 se especifica el caso de uso “Ingresar Estudiante”, donde el usuario ingresará al sistema un nuevo estudiante, esto da cumplimiento requerimiento de ingreso de estudiantes.

Caso de Uso	Ingresar Estudiante
<b>ID</b>	<b>5</b>
<b>Descripción</b>	<b>El usuario ingresará los datos de un nuevo estudiante</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Ninguna</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos para un estudiante.</b> <b>2.- El usuario llenará los campos con los datos del estudiante.</b> <b>2b.- El usuario podrá hacer uso del mapa para ubicar la dirección del estudiante con mayor precisión.</b> <b>3.- El usuario guardara datos.</b>
<b>Post-condiciones</b>	<b>El estudiante será creado y sus datos serán vistos en la página.</b>

Tabla 9 Caso de Uso Agregar nuevo Estudiante

### 5.3.2.6 Caso de Uso: Eliminar Estudiante

En la Tabla 10 se especifica el caso de uso “Eliminar Estudiante”, donde el usuario removerá del sistema un estudiante, esto es con el fin de mantener información consistente de los estudiantes, respecto al requerimiento de ingreso y visualización de estudiantes.

Caso de Uso	Eliminar Estudiante
<b>ID</b>	<b>6</b>
<b>Descripción</b>	<b>El usuario eliminará un estudiante y la información relacionada con este.</b>
<b>Actores del sistema</b>	<b>Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un usuario de la lista.</b>
<b>Flujos principales</b>	<b>1.- El sistema solicitará una confirmación para la eliminación.</b> <b>2.- El usuario deberá confirmar la eliminación del estudiante del sistema.</b>
<b>Post-condiciones</b>	<b>El sistema cargará la página con la lista de estudiantes y ya no será visible la información del estudiante eliminado.</b>

Tabla 10 Caso de Uso Eliminar Estudiante

### 5.3.2.7 Caso de Uso: Ver Listado de Eventos

En la Tabla 11 se especifica el caso de uso “Ver Listado de Eventos”, donde el usuario visualizará los eventos registrados en el sistema, con el fin acceder a los eventos en el sistema, respecto al requerimiento de visualización de eventos catastróficos.

Caso de Uso	Ver Listado de Eventos
<b>ID</b>	<b>7</b>
<b>Descripción</b>	<b>El usuario visualiza un listado de eventos</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber iniciado sesión</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará la información de los distintos eventos.</b>
<b>Post-condiciones</b>	<b>En la página estará disponible un menú de acciones y además el usuario podría ver el detalle de cada evento haciendo click en el id de estos</b>

Tabla 11 Listar Eventos

### 5.3.2.8 Caso de Uso: Ver Información Evento

En la Tabla 12 se especifica el caso de uso “Ver Información Evento”, donde el usuario visualizará el detalle de un evento, respecto al requerimiento de visualización de eventos catastróficos.

Caso de Uso	Ver Información Evento
<b>ID</b>	<b>8</b>
<b>Descripción</b>	<b>El usuario visualizará el detalle respecto a la información de un determinado evento</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un evento de la lista</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos de un determinado evento</b>
<b>Post-condiciones</b>	<b>En la página estará disponible un menú de acciones.</b>

Tabla 12 Caso de Uso Ver Detalle de un Evento

### 5.3.2.9 Caso de Uso: Actualizar Evento

En la Tabla 13 se detalla el caso de uso “Actualizar Evento”, donde el usuario modificará un evento en particular, cumpliendo con el requerimiento de visualización de eventos catastróficos.

Caso de Uso	Actualizar Evento
<b>ID</b>	<b>9</b>

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

<b>Descripción</b>	<b>El usuario editará los datos de un determinado evento</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un evento de la lista</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos de un determinado evento.</b> <b>2.- El usuario modificará aquellos datos que desee</b> <b>3.- El usuario guardará los cambios realizados</b>
<b>Post-condiciones</b>	<b>Los datos del evento serán actualizados y serán vistos en la página.</b>

Tabla 13 Caso de Uso Editar datos de un Evento

### 5.3.2.10 Caso de Uso: Ingresar Evento

En la Tabla 14 se especifica el caso de uso “Ingresar Evento” donde el usuario ingresará un nuevo evento al sistema, conforme al requerimiento de ingreso de eventos catastróficos.

<b>Caso de Uso</b>	<b>Ingresar Evento</b>
<b>ID</b>	<b>10</b>
<b>Descripción</b>	<b>El usuario ingresará los datos de un nuevo evento</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Ninguna</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos para un evento.</b> <b>2.- El usuario llenará los campos con los datos del evento.</b> <b>2b.- El usuario vinculara el evento con los estudiantes afectados</b> <b>3.- El usuario guardará datos.</b>
<b>Post-condiciones</b>	<b>El estudiante será creado y sus datos serán vistos en la página.</b>

Tabla 14 Caso de Uso Crear nuevo Evento

### 5.3.2.11 Caso de Uso: Eliminar Evento

En la Tabla 15 se detalla el caso de uso “Eliminar Evento”, en el cual el usuario removerá un evento del sistema, para generar una mejor calidad de datos y respondiendo al requerimiento de ingreso y visualización de eventos catastróficos.

<b>Caso de Uso</b>	<b>Eliminar Evento</b>
<b>ID</b>	<b>11</b>
<b>Descripción</b>	<b>El usuario eliminará un evento y la información relacionada con este.</b>
<b>Actores del sistema</b>	<b>Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un evento de la lista.</b>

<b>Flujos principales</b>	<b>1.- El sistema solicitará una confirmación para la eliminación. 2.- El usuario deberá confirmar la eliminación del evento del sistema.</b>
<b>Post-condiciones</b>	<b>El sistema cargará la página con la lista de eventos y ya no será visible la información del evento eliminado.</b>

Tabla 15 Caso de Uso Eliminar Evento

### 5.3.2.12 Caso de Uso: Ingresar Estudiante Afectado

En la Tabla 16 se especifica el caso de uso “Ingresar Estudiante Afectado”, donde el usuario relacionará un estudiante con un evento, conforme al requerimiento de ingreso de eventos catastróficos.

<b>Caso de Uso</b>	<b>Ingresar Estudiante Afectado</b>
<b>ID</b>	<b>12</b>
<b>Descripción</b>	<b>El ingresará el detalle de un estudiante afectado por algún evento.</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un estudiante y un evento de la lista.</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará los campos relacionada con los detalles 2.- El usuario deberá llenar los campos. 3.- El usuario guardará la información.</b>
<b>Post-condiciones</b>	<b>El sistema cargará la página con la lista de eventos.</b>

Tabla 16 Caso de Uso Ingresar Estudiante Afectado

### 5.3.2.13 Caso de Uso: Reportes

En la Tabla 17 se detalla el caso de uso “Reportes”, donde el usuario solicitará al sistema los diferentes reportes especificados según los requerimientos de reportes.

<b>Caso de Uso</b>	<b>Reportes</b>
<b>ID</b>	<b>13</b>
<b>Descripción</b>	<b>El usuario generará un reporte.</b>
<b>Actores del sistema</b>	<b>Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Ninguno</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará una lista con los reportes disponibles. 2.- El usuario seleccionará el reporte que desee. 3.- El sistema desplegará los campos con las opciones para el filtrado de la información. 4.- El usuario ingresará o seleccionara los datos para la generación del reporte.</b>
<b>Post-condiciones</b>	<b>El sistema generará el reporte que podrá ser guardado en PDF.</b>

Tabla 17 Caso de Uso Reportes

### 5.3.2.14 Caso de Uso: Finalizar Sesión

En la Tabla 18 se detalla el caso de uso “Cerrar Sesión”, en el cual el usuario cerrará su sesión, cumpliendo el requerimiento referente al uso de sesiones.

Caso de Uso	Cerrar Sesión
<b>ID</b>	<b>14</b>
<b>Descripción</b>	<b>El usuario cerrará la sesión.</b>
<b>Actores del sistema</b>	<b>Administrativo, Jefatura, Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber iniciado sesión.</b>
<b>Flujos principales</b>	<b>1.- El usuario seleccionará la opción “Salir” en la barra de menú. 2.- El sistema solicitará confirmación de cierre de sesión. 3.- El usuario confirmará el cierre de sesión.</b>
<b>Post-condiciones</b>	<b>El sistema cargará la página inicio con los campos para iniciar una sesión.</b>

Tabla 18 Caso de Uso Cerrar Sesión

### 5.3.2.15 Caso de Uso: Ver Listado de Usuarios

En la Tabla 19 se especifica el caso de uso “Ver Listado de Usuarios”, donde el Administrador solicitará al sistema el listado de los usuarios.

Caso de Uso	Ver Listado de Usuarios
<b>ID</b>	<b>15</b>
<b>Descripción</b>	<b>El administrador visualiza un listado de usuarios del sistema</b>
<b>Actores del sistema</b>	<b>Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber iniciado sesión</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará la información de los distintos usuarios.</b>
<b>Post-condiciones</b>	<b>En la página estará disponible un menú de acciones y además el usuario podría ver el detalle de cada usuario haciendo click en el id de estos</b>

Tabla 19 Listar Usuarios

### 5.3.2.16 Caso de Uso: Ver Información Usuario

En la Tabla 20 se detalla el caso de uso “Ver Información Usuario”, donde el Administrador visualizará el detalle un determinado usuario, según el requerimiento de usuarios y roles.

Caso de Uso	Ver Información Usuario
<b>ID</b>	<b>16</b>

Sistema Web de Geocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

<b>Descripción</b>	<b>El administrador visualizará el detalle respecto a la información de un determinado usuario</b>
<b>Actores del sistema</b>	<b>Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un usuario de la lista</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos de un determinado usuario</b>
<b>Post-condiciones</b>	<b>En la página estará disponible un menú de acciones.</b>

Tabla 20 Caso de Uso Ver Detalle de un Usuario

### 5.3.2.17 Caso de Uso: Actualizar Usuario

En la Tabla 21 se detalla el caso de uso “Actualizar Usuario”, donde el Administrador editará la información de un usuario, según el requerimiento de usuarios y roles.

<b>Caso de Uso</b>	<b>Actualizar Usuario</b>
<b>ID</b>	<b>17</b>
<b>Descripción</b>	<b>El administrador editará los datos de un determinado usuario</b>
<b>Actores del sistema</b>	<b>Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un usuario de la lista</b>
<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos de un determinado usuario. 2.- El usuario modificará aquellos datos que desee 3.- El usuario guardará los cambios realizados</b>
<b>Post-condiciones</b>	<b>Los datos del usuario serán actualizados y serán vistos en la página.</b>

Tabla 21 Caso de Uso Editar datos de un Usuario

### 5.3.2.18 Caso de Uso: Ingresar Usuario

En la Tabla 22 se especifica el caso de uso “Ingresar Usuario”, donde el Administrador ingresará un nuevo usuario al sistema, según el requerimiento de usuarios y roles.

<b>Caso de Uso</b>	<b>Ingresar Usuario</b>
<b>ID</b>	<b>18</b>
<b>Descripción</b>	<b>El usuario ingresará los datos de un nuevo evento</b>
<b>Actores del sistema</b>	<b>Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Ninguna</b>

<b>Flujos principales</b>	<b>1.- El sistema desplegará en la página los datos para un usuario.</b> <b>2.- El administrador llenará los campos con los datos del usuario.</b> <b>3.- El administrador guardará datos.</b>
<b>Post-condiciones</b>	<b>El usuario será creado y sus datos serán vistos en la página.</b>

Tabla 22 Caso de Uso Crear nuevo Usuario

### 5.3.2.19 Caso de Uso: Eliminar Usuario

En la Tabla 23 se especifica el caso de uso “Eliminar Usuario”, en el cual el Administrador eliminará del sistema un determinado usuario, según el requerimiento de usuarios y roles.

<b>Caso de Uso</b>	<b>Eliminar Usuario</b>
<b>ID</b>	<b>19</b>
<b>Descripción</b>	<b>El administrador eliminará un usuario y la información relacionada con este.</b>
<b>Actores del sistema</b>	<b>Administrador</b>
<b>Actores secundarios</b>	<b>No tiene.</b>
<b>Precondiciones</b>	<b>Haber seleccionado previamente un evento de la lista.</b>
<b>Flujos principales</b>	<b>1.- El sistema solicitará una confirmación para la eliminación.</b> <b>2.- El usuario deberá confirmar la eliminación del usuario del sistema.</b>
<b>Post-condiciones</b>	<b>El sistema cargará la página con la lista de usuarios y ya no será visible la información del usuario eliminado.</b>

Tabla 23 Caso de Uso Eliminar Usuario

## 5.4 Descripción global del producto

El producto a desarrollar es un sistema web para el mantenimiento de información de estudiantes. Este está compuesto por diferentes módulos, cada cual otorga al usuario una funcionalidad diferente. A continuación se detallan los principales módulos del sistema.

### 5.4.1 Página de Inicio

La Figura 13 representa la página de inicio de la aplicación, esta variará dependiendo si el usuario ha ingresado o no al sistema.

Si el usuario aun no ingresa sus credenciales, solo le será posible ver el login del sistema. En el caso que ya haya ingresado, le serán visibles algunos accesos directos en la página de inicio, además de una barra de menú con las opciones disponibles para dicho usuario.

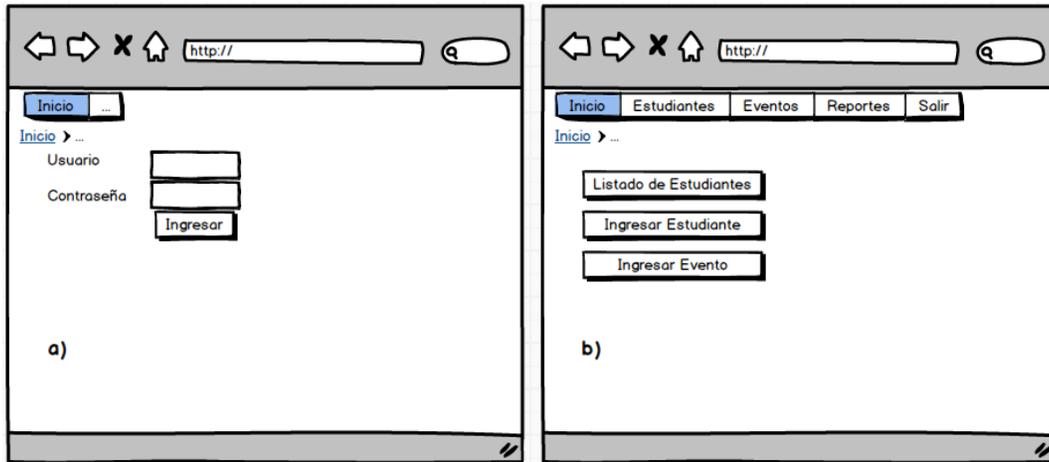


Figura 13 Página de Inicio a)Login b)Menús de acceso para sesión iniciada

### 5.4.2 Página Listado de Estudiantes

La Figura 14 presenta una lista con los estudiantes, de acuerdo a algún filtro para la visualización de estos, además de un mapa con pines que representan a cada uno de los estudiantes, identificados por su nombre.

Al hacer clic sobre el pin se desplegará un globo con información del estudiante correspondiente, además de una fotografía de este.

Es posible acceder a los detalles de un estudiante en el listado haciendo clic sobre el rut de este o en su fotografía en caso se esté observando el globo de información en el mapa.

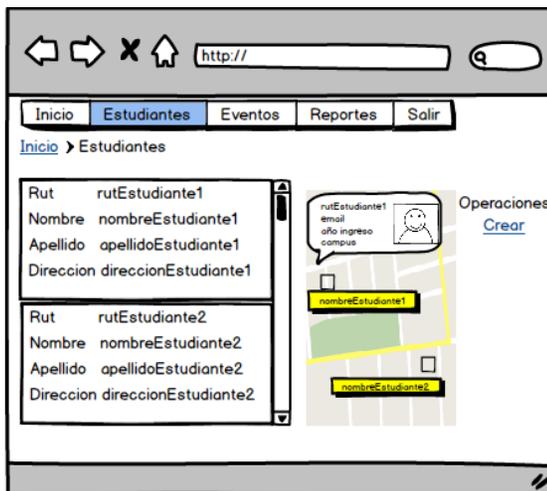


Figura 14 Listado de Estudiantes

### 5.4.3 Página Detalle de Estudiante

La Figura 15 presenta la visualización de un estudiante previamente seleccionado junto con su visualización en el mapa. Esta página incluye un menú lateral con opciones en este, tales como

editar, eliminar o crear un nuevo estudiante. Además de la opción de volver al listado de estudiantes.



Figura 15 Detalle de un Estudiante

#### 5.4.4 Página Agregar nuevo Estudiante

Para la creación de un nuevo estudiante se presentará una página como puede se visualiza en la Figura 16. Esta página incluye los campos correspondientes a los datos del estudiante, además de un mapa en el cual podrá ser desplazado un pin a fin de capturar la ubicación geográfica del estudiante y de este modo registrar con mayor exactitud las coordenadas de su dirección. Esta función es de utilidad en aquellos casos en los cuales la dirección del estudiante no está incluida en la actualización del Google Maps, esto podría ser que la calle haya sido recientemente construida. También en el caso en el cual la dirección no pueda ser geocodificada por el servicio de mapas, como lo son las zonas rurales.

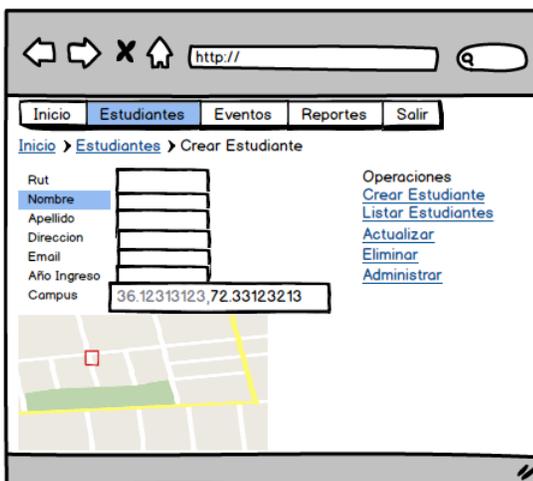


Figura 16 Ingresar nuevo Estudiante

### 5.4.5 Página Listado de Eventos

La Figura 17 presenta una lista con los eventos, de acuerdo a algún filtro para la visualización de estos. Es posible acceder a los detalles de un evento en el listado haciendo clic sobre el identificador de este.

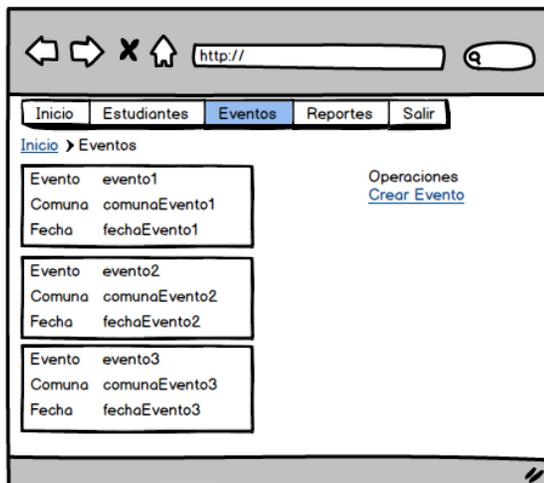


Figura 17 Listado de Eventos

### 5.4.6 Página Detalle de un Evento

Al ser seleccionado un evento en particular, se visualizará una página como la presentada en la Figura 18. Los detalles de esta corresponden a la ubicación del evento, una descripción de este y un listado con los estudiantes que se fueron afectados.

Habrá además un mapa en el cual la zona afectada será demarcada y mediante pines se visualizaran el o los estudiantes afectados.

Un menú lateral dará acceso a algunos servicios de los eventos, tales como la edición o eliminación del presente evento.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera Ingeniería Civil en Informática

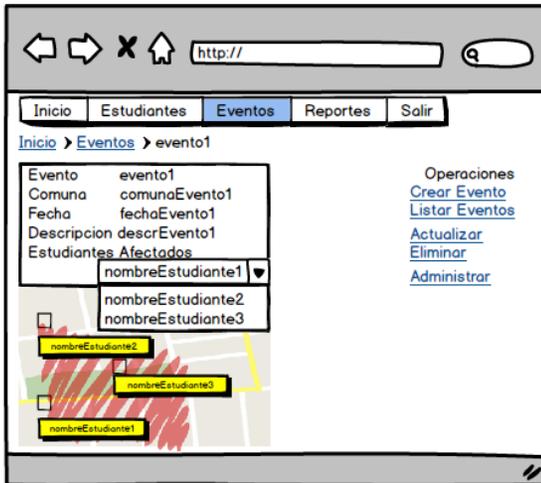


Figura 18 Detalle de un Evento

### 5.4.7 Página Crear Nuevo Evento

En la Figura 19 se visualiza la página correspondiente a la creación de un nuevo evento, en el cual el usuario ingresará los datos correspondiente a un evento sucedido, además seleccionará los estudiantes que fueron afectados por este.

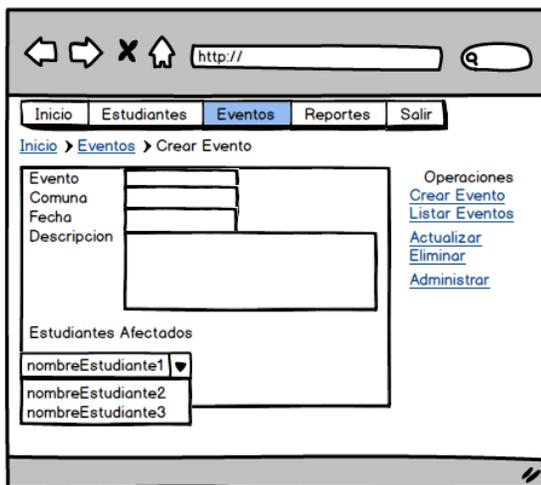


Figura 19 Creación nuevo Evento

### 5.4.8 Página Agregar Estudiante Afectado

En el caso de que un evento haya afectado a uno o más estudiantes, se visualizará una página como la representada en la Figura 20. Esta página contiene los campos en los cuales se ingresaran las observaciones hacia el estudiante referente al evento sucedido y además se le categorizará con una prioridad.

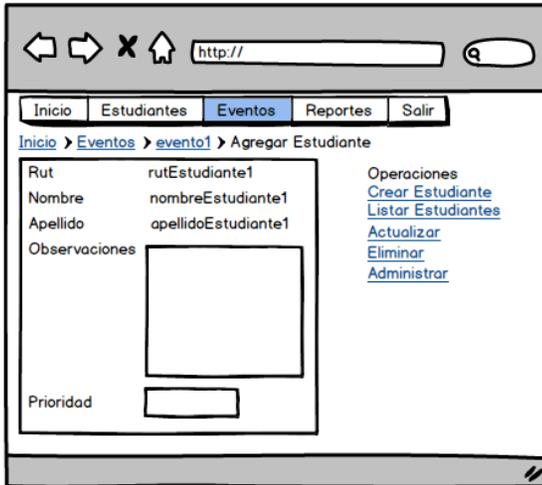


Figura 20 Ingreso de un Estudiante Afectado por un Evento

### 5.4.9 Página de Reportes

Como puede verse en la Figura 21, se contará con una página para la selección de los diferentes reportes que generará el sistema.

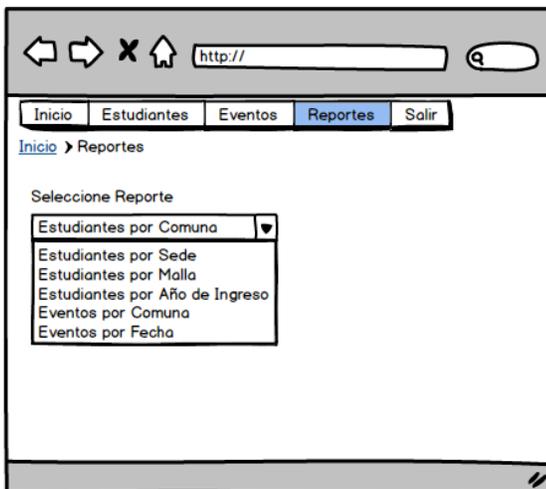


Figura 21 Página de Reportes.

## 5.5 Modelamiento de Datos

El Modelo Entidad Relación, fue introducido por Peter Chen en 1976 y se define como una herramienta de modelamiento de datos que describe las asociaciones que existen entre las diferentes categorías de datos dentro de un sistema de empresa o de información [25].

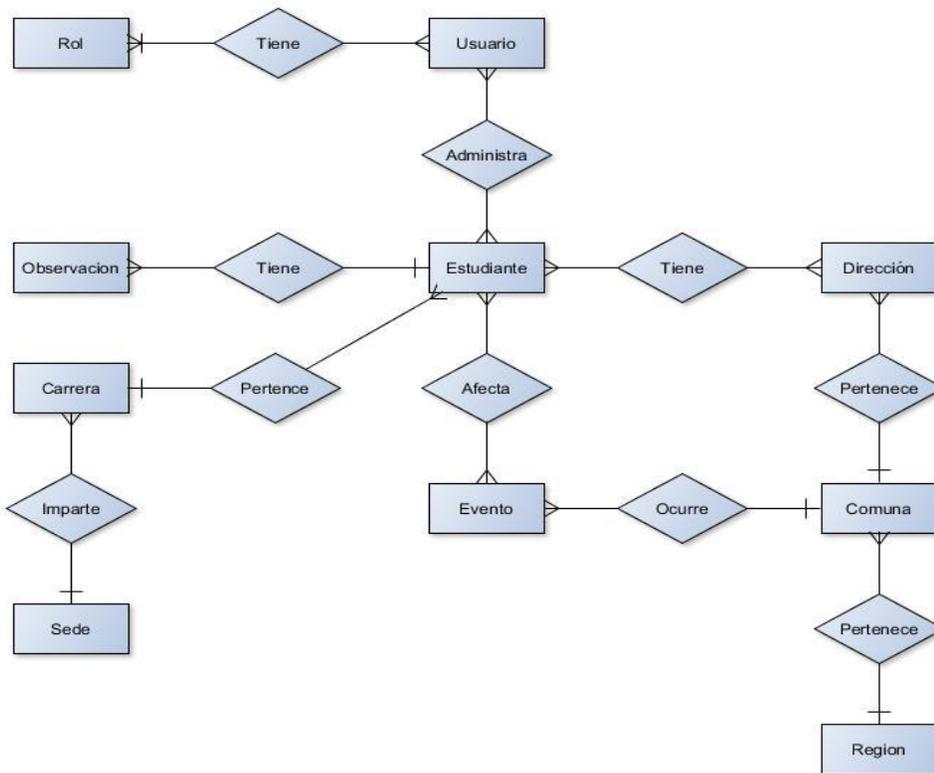


Figura 22 Modelo Entidad Relación

En la Figura 22 se presenta el modelo que representa la información vital para la vida del sitio expresado en las relaciones:

- Entidades:
  - **Estudiante:** entidad que representa a los estudiantes de la universidad del Bío-Bío.
  - **Dirección:** representa al domicilio que puede tener un estudiante.
  - **Comuna:** entidad que representa a la comuna a la cual pertenece el domicilio de un Estudiante.
  - **Región:** representa a la unidad regional en la cual se encuentra la comuna
  - **Carrera:** es la carrera que el estudiante se encuentra cursando
  - **Sede:** entidad que representa la sede a la donde es impartida una carrea
  - **Usuario:** entidad que representa al usuario del sistema.

- **Rol:** esta entidad tiene por objetivo diferenciar los usuarios otorgándoles diferentes permisos y privilegios.
- **Evento:** esta entidad representa a cualquier catástrofe que pudiera afectar a los domicilios de los estudiantes.
- **Observación:** entidad cuyo fin es identificar el daño u similar que haya afectado a un estudiante frente a alguna evento catastrófico.
- Relaciones:
  - Estudiante – tiene – Dirección: un estudiante tiene una o varias direcciones. A la vez una dirección puede pertenecer a uno o varios estudiantes
  - Evento – afecta – Estudiante: un evento puede afectar a varios estudiantes, del mismo modo un estudiante puede ser afectado por varios eventos.
  - Estudiante – tiene – Observación: un estudiante puede tener una o varias observaciones producto de los eventos que le hayan afectado.
  - Usuario – tiene – Rol: Cada usuario solo puede tener un rol asignado dentro del sistema.
  - Usuario – administra – Estudiante: La mantención de información referente a los estudiantes será realizada por un usuario.

### 5.6 Diseño Físico de la Base de Datos

En la Figura 23 se encuentra representado el modelo físico de la base de datos. Si bien presenta diferencias con el modelo entidad relación presentado en la Figura 22, estas se deben al proceso de normalización del modelo de datos.

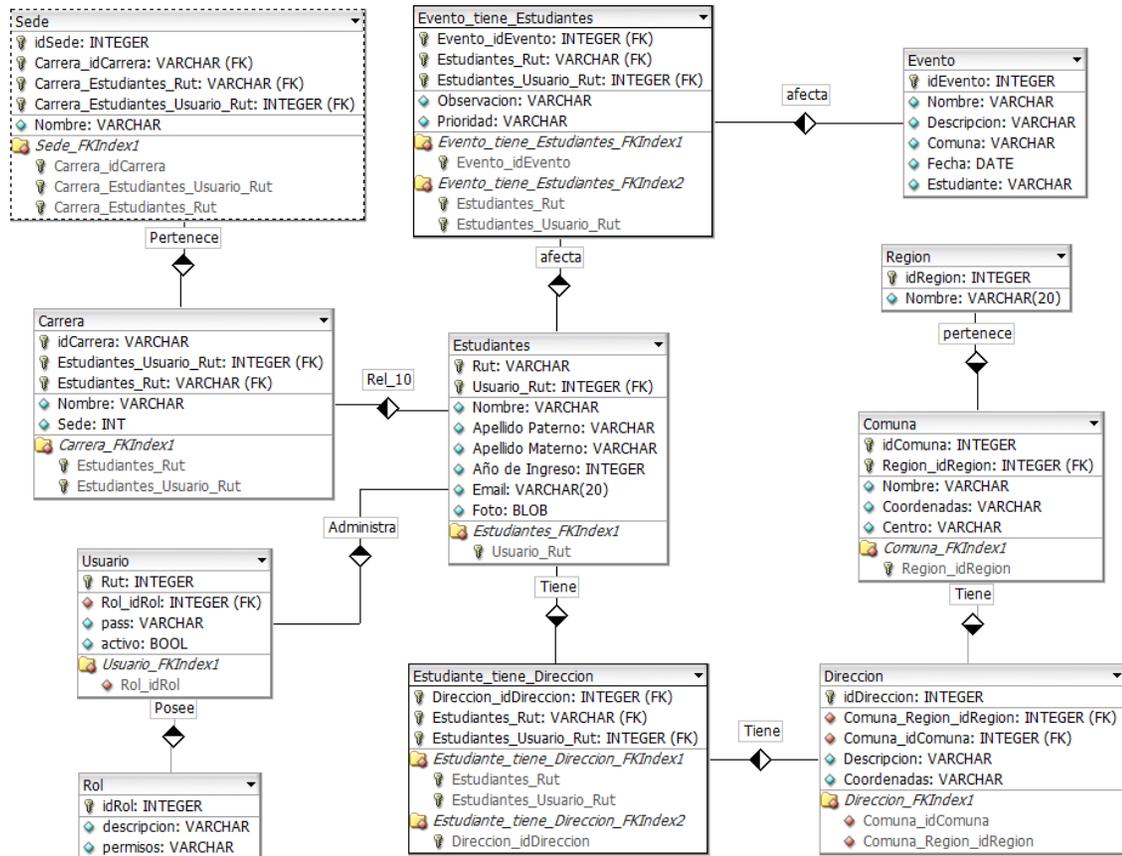


Figura 23 Modelo Físico Base de Datos

## 5.7 Diseño de Arquitectura Funcional

En la Figura 24 puede apreciarse el diagrama funcional del sitio web, el cual nos ilustra la manera en que la información ha sido estructurada.



Figura 24 Modelo de Arquitectura Funcional

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

# IMPLEMENTACIÓN

## **6 IMPLEMENTACIÓN**

---

### **6.1 Introducción**

Con el fin de profundizar el entendimiento del sistema, se ha dedicado este capítulo para la implementación de aquellos módulos y servicios específicos en este proyecto. Explicando cómo se desarrollaron algunos de los componentes más importantes de la aplicación.

También se comentará en detalle algunas de las herramientas y conceptos utilizados, ya que es imprescindible tomar en cuenta la gestión del conocimiento, puesto que la tecnología es un elemento que evoluciona.

Por ello se analizará el funcionamiento de Google Maps, su uso en un aplicación web. Además, la aplicación de conceptos de usabilidad en un sitio web.

## 6.2 Api de Google Maps.

### 6.2.1 Introducción.

El API de Google Maps permite utilizar mapas con gran nivel de detalle en páginas Web, mediante la utilización de JavaScript. El API proporciona diferentes utilidades para manipular mapas y añadir contenido a los mismos mediante diversos servicios, permitiendo crear sólidas aplicaciones de mapas en sitios web.

Este servicio se encuentra disponible en diferentes lenguajes y frameworks, a continuación se presentan los servicios disponibles para la utilización de Google Maps:

- Maps API for Flash
- Maps JavaScript API (Versión 2 y 3).
- Static Maps API

### 6.2.2 Características de Google Maps.

Google Maps fue presentado el 8 de Enero de 2005. Originalmente soportaría sólo a los usuarios de Internet Explorer y Mozilla Firefox, sin embargo, el soporte para Opera y Safari fue agregado el 25 de febrero de 2005. El software estuvo en su fase beta por 6 meses antes de convertirse en parte de Google Local, el 6 de octubre de 2005.

Google Maps fue una de las primeras aplicaciones basadas en AJAX de uso masivo por parte de los usuarios. Su gran éxito ha provocado que compañías imiten y realicen nuevas propuestas, sin embargo, Google Maps continúa acaparando la preferencia de los usuarios.

Google ofrece de forma gratuita un API para el desarrollo de aplicaciones a medida basadas en su servicio de mapas, con esto se permite integrar los mapas en otras aplicaciones e incluso hacer mash-up o mezclas de Google Maps y otras aplicaciones web que también disponen de una API pública.

Sin embargo, Google impone limitaciones en el uso de su API. A continuación se presentan tales limitaciones.

- No existe limitación en el número de visitas diarias a la página que se puede generar mediante Google Maps API.
- El número de solicitudes de codificación geográfica que se pueden enviar diariamente es limitado.
- El API de Google Maps no incluye publicidad.
- Los usuarios finales deberán acceder de forma gratuita al servicio.
- No se pueden modificar ni ocultar los logotipos ni la atribución del mapa.
- Se puede utilizar el API (salvo en el caso de Static Maps API) en sitios Web o en aplicaciones de software. En el caso de sitios Web, se debe registrar la URL del servidor que soportará la aplicación que contenga los mapas.

### 6.2.3 Google Maps JavaScript API

La versión de JavaScript permite la incrustación de mapas dinámicos en páginas Web, así como la utilización de geo-localización, es un API de gran estabilidad y eficiencia, que necesita navegadores compatibles con JavaScript y acceso a Internet para su normal funcionamiento

### 6.2.4 Funcionamiento de Google Maps.

Google utiliza KML, que es un formato de archivo que se utiliza para mostrar datos geográficos en un navegador terrestre, como Google Earth, Google Maps y Google Maps para móviles.

Los archivos KML permiten identificar ubicaciones, añadir superposiciones de imágenes y presentar datos detallados de formas diferentes. KML es un estándar internacional mantenido por Open Geospatial Consortium, Inc.

Es el standard KML el que nos permite generar puntos, líneas y polígonos dentro de un sistema de mapas. Además de algunas otras capas, que no serán consideradas en este proyecto.

Por ejemplo, la Figura 25, corresponde a un segmento de código en formato *kml*, similar al xml, el uso de *tags* permiten al API la interpretación de lo que se intenta generar en el mapa.

En este caso se trata de un polígono, conformado por una serie de coordenadas, **(-36.701111,-72.436927)** es una de ellas, bajo esta notación de xml, Google Maps interpretara nuestra intención de unir cada coordenada y con ello formar un polígono, que en este caso corresponde a lo que podrían ser los límites de la Comuna de Chillán.

```

1  <kml xmlns="http://www.opengis.net/kml/2.2">
2  <Placemark>
3    <name>ComunaChillan.kml</name>
4    <Polygon>
5      <outerBoundaryIs>
6        <LinearRing>
7          <coordinates>
8            -36.701111,-72.436927
9            -36.707096,-72.383461
10           -36.693438,-72.348128
11           -36.655905,-72.352081
12           -36.631406,-72.302224
13           -36.614344,-72.139443
14           -36.668713,-72.078744
15           .
16           .
17           .
18           -36.583447,-72.224847
19           -36.578379,-72.255014
20           -36.599187,-72.28308
21           -36.590247,-72.310615
22           -36.6447,-72.465299
23         </coordinates>
24       </LinearRing>
25     </outerBoundaryIs>
26   </Polygon>
27 </Placemark>
28 </kml>

```

Figura 25 Segmento de código en formato KML

Lo anterior nos da la posibilidad de representar en un mapa zonas definidas políticamente como podrían ser ciudades, comunas, provincias o regiones, siempre que se tenga acceso a los puntos que conforman dichas zonas.

### 6.3 Sistema Integrado de Información Territorial

El Sistema Integrado de Información Territorial (SIIT) [12] es una herramienta facilitada por la Biblioteca del Congreso Nacional, que como anuncia en su página web, pone a nuestra disposición información territorial de todas las unidades político-administrativas en las que se divide Chile; regiones, provincias y comunas, como así también sus unidades electorales: circunscripciones senatoriales y distritos de diputados.

Lo anteriormente mencionado nos permite acceder a información suficiente como para que este proyecto pueda representar diferentes zonas en un mapa. Ya que la Biblioteca del Congreso Nacionales posible obtener los mapas vectoriales de las diferentes unidades en las cuales está dividido el territorio nacional.

Sin embargo, el formato en el cual se encuentran estos mapas vectoriales es **Shapefile**, de extensión *shp*. Este formato es característico de la compañía ESRI, la cual comercializa software para SIG. El formato de este es sencillo y no topológico que se utiliza para almacenar la ubicación geométrica y la información de atributos de las entidades geográficas. Las entidades geográficas de un shapefile se pueden representar por medio de puntos, líneas o polígonos (áreas) [13]. Si bien este formato posee suficiente información para generar los polígonos, no es compatible con el formato de texto plano que utiliza el API de Google Maps.

### 6.4 Conversión de SHP a KML

Con la información referente a la división política entregada por la Biblioteca del Congreso Nacional, es posible dibujar en un mapa un polígono que represente dicha unidad, facilitando de este modo la interpretación de este, dando la posibilidad de mostrar una zona demarcada dentro de un mapa.

Para utilizar la información proporcionada por la Biblioteca del Congreso Nacional, es necesario poder extraer las coordenadas que componen los diferentes polígonos que finalmente representarían las zonas. Para esto es posible utilizar el servicio de una aplicación web llamada "Shape Escape"<sup>2</sup>, el cual permite convertir los Shapefile en **Google Fusion Tables**, que consiste en una herramienta experimental de Google [14] para la visualizar y compartir tablas con datos.

Básicamente lo que Shape Escape hace es tomar la información contenida en los archivos shp y acceder mediante nuestra cuenta de Google al servicio de Google Drive y añadir los datos obtenidos a una Fusion Table.

---

<sup>2</sup> <http://www.shpescape.com/ft/upload/>

El funcionamiento de este servicio es simple, como puede observarse en la Figura 26, solo es necesario cargar el archivo comprimido obtenido desde la Biblioteca del Congreso Nacional.

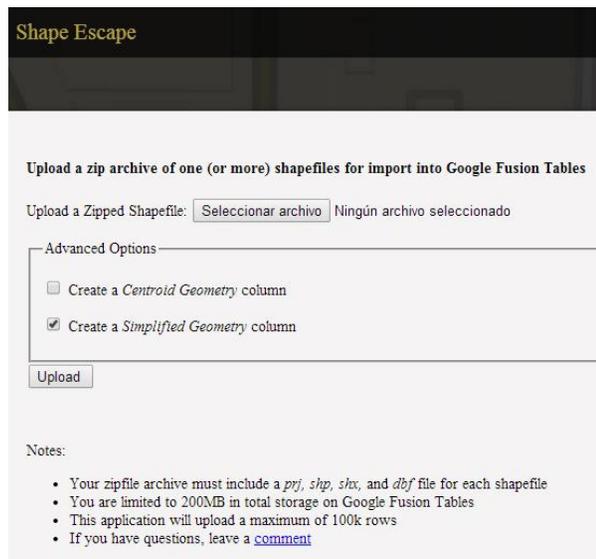


Figura 26 Shape Escape Interfaz

Una vez finalizada la conversión por parte de Shape Escape, este nos proporcionará el vínculo correspondiente a la Fusion Table, semejante a la de la Figura 27, correspondiente en nuestra cuenta de Google Drive.

division\_comunal  
 Imported at Sat May 17 15:05:48 PDT 2014 from division\_comunal.csv.  
 Edited on May 17, 2014

File Edit Tools Help Rows 1 Cards 1

Filter No filters applied

1-100 of 346

NOM_REG;NOM_PROV;NOM_COM;SHAPE_LENG;DIS_ELEC;CIR_SENA;COD_COMUNA;SHAPE_Le_1;SHAPE_Area	col1	col2	col3
Region de La Araucania;Malleco;Ercilla;170675	59543800000;48;14;9204;170675	59570600000;497902482	30900000000
Region de La Araucania;Malleco;Lonquimay;346907	19543700000;49;14;9205;346907	19589900000;3932854068	57000000000
Region de La Araucania;Malleco;Los Sauces;169844	12228600000;48;14;9206;169844	12223900000;851973016	71500000000
Region de La Araucania;Malleco;Lumaco;227705	21920400000;48;14;9207;227705	21880600000;1112642685	57000000000
Region de La Araucania;Malleco;Purén;137030	48047400000;48;14;9208;137030	48035300000;465475974	96900000000
Region de La Araucania;Malleco;Renaico;114371	36835200000;48;14;9209;114371	36861400000;265033862	14200000000
Region de La Araucania;Malleco;Traiguén;191794	77649600000;48;14;9210;191794	77648400000;897793661	86300000000
Region de La Araucania;Malleco;Victoria;195942	04971300000;49;14;9211;195942	04920500000;1262086778	32000000000
Region de Los Lagos;Chiloe;Castro;273556	69389500000;58;17;1020;273556	69509400000;469410398	86100000000
Region de Los Lagos;Chiloe;Chonchi;231548	49419500000;58;17;1020;231548	49229100000;1365844853	18000000000
Region de Los Lagos;Chiloe;Curaco de Vejez;49577	17954410000;58;17;1020;49577	17964070000;79193853	90820000000
Region de Los Lagos;Chiloe;Dalcahue;302364	03277600000;58;17;1020;302364	03279100000;1243142803	09000000000
Region de Los Lagos;Chiloe;Puqueldón;72488	11555630000;58;17;1020;72488	11552180000;95545308	35730000000
Region de Los Lagos;Chiloe;Queilén;197534	30962600000;58;17;1020;197534	30925000000;328765332	36500000000
Region de Los Lagos;Chiloe;Quellón;884977	68340600000;58;17;1020;884977	68507600000;3364376510	68000000000
Region de Los Lagos;Chiloe;Quemchi;357198	14043900000;58;17;1020;357198	13889300000;436693664	12100000000
Region de Los Lagos;Chiloe;Quinchao;216985	34494500000;58;17;1021;216985	34484000000;156021977	46700000000
Region de Los Lagos;Llanquihue;Calbuco;462933	75049900000;57;17;1010;462933	75153900000;586823716	07900000000

Figura 27 Fusion Table creada por Shape Escape

Si bien estos datos son legibles, no es posible aun utilizarlos en los servicios de KML, aun poseen demasiada información para esto. Es necesario entonces realizar una siguiente conversión. Esta vez a un archivo de valores separados por comas (CSV por sus siglas en ingles). Estos son documentos donde es sencillo representar datos en forma de tabla, donde las columnas se separan por comas (o punto y coma en caso de que se use la coma como separador decimal) y las filas por saltos de línea [15]. Fusion Tables permite la descarga de la tabla en formato CSV, por lo que su obtención es realmente sencilla. Resultando un documento con un contenido similar al de la Figura 28.

```

1 geometry_simplified,geometry_simplified_vertex_count,geometry,geometry_vertex_count,NOM_REG,NOM_PROV,NOM_
  COM,SHAPE_LENG,DIS_ELEC,CIR_SENA,COD_COMUNA,SHAPE_Le_1,SHAPE_Area,import_notes
2 "<Polygon><outerBoundaryIs><LinearRing><coordinates>-71.351068,-33.528545 -71.316957,-33.588208
  -71.251252,-33.582507 -71.207282,-33.544785 -71.059644,-33.556035 -71.013428,-33.539942
  -71.061632,-33.492416 -71.059192,-33.430721 -71.179882,-33.45053 -71.276423,-33.404382
  -71.331112,-33.407646 -71.353134,-33.492562</coordinates></LinearRing></outerBoundaryIs></Polygon>",13,"
  <Polygon><outerBoundaryIs><LinearRing><coordinates>-71.267511,-33.409593 -71.268847,-33.409002
  -71.271101,-33.407784 -71.271919,-33.40729 </coordinates></LinearRing></outerBoundaryIs></Polygon>
  ",1593,RegiA3n Metropolitana de Santiago,Melipilla,MarAa
  Pinto,109727.643978,31,7,1350,109727.644651,394506632.672000, .....
3 "<Polygon><outerBoundaryIs><LinearRing><coordinates>-71.711538,-33.966755 -71.671625,-33.975521
  -71.66594,-33.997749 -71.615752,-34.003994 -71.58708,-34.033906 -71.56531,-33.991389
  -71.541757,-34.008889 -71.493361,-34.003283 -71.420202,-34.034232 -71.389917,-34.075656
  -71.207102,-33.990299 -71.20721,-33.96069 -71.287207,-33.908503 -71.300422,-33.864212
  -71.34204,-33.867258 -71.355764,-33.889572 -71.372579,-33.889194 -71.419903,-33.821191
  -71.422753,-33.79432 -71.536547,-33.764123 -71.555751,-33.782483 -71.5592,-33.869423
  -71.581262,-33.867125 -71.714347,-33.947103</coordinates></LinearRing></outerBoundaryIs></Polygon>",25,"
  <Polygon><outerBoundaryIs><LinearRing><coordinates>-71.539582,-33.764087 -71.5402,-33.764027
  -71.540761,-33.763968 -71.541214,-33.764053 -71.541895,-33.764227 -71.54246,-33.764309
  
```

Figura 28 Resultado de Conversión de Fusion Table a CSV

Los valores de este documento se asemejan más a los necesarios a los que utiliza el estándar KML, este contiene 14 campos, tales como los puntos que componen una comuna de manera simplificada, de manera detallada, el nombre de esta u otros.

Toda esta información puede ser simplificada utilizando una herramienta web para conversión de CSV<sup>3</sup>. Entre las conversiones que ofrece se encuentra la de CSV a KML. Para utilizar esta funcionalidad se debe cargar el CSV, la herramienta nos permitirá la columna que servirá como nombre del atributo y la columna que contenga el valor correspondiente.

En este caso el archivo csv, como puede verificarse en la Figura 29, posee 14 valores. Para este proyecto solo es necesario tener acceso al nombre de la comuna y al polígono que la represente. Ya que no es la finalidad del sistema procesar el detalle de cada comuna, con el polígono simplificado bastaría, ya que posee los suficientes puntos para que visualmente puedan diferenciarse los límites de una comuna en el mapa. Para este caso el nombre del atributo será la columna 7 que representa el nombre de la comuna y el valor del atributo la columna 1, que contiene el polígono simplificado.

```

1 geometry_simplified,geometry_simplified_vertex_count,geometry,
  geometry_vertex_count,NOM_REG,NOM_PROV,NOM_COM,SHAPE_LENG,DIS_ELEC,
  CIR_SENA,COD_COMUNA,SHAPE_Le_1,SHAPE_Area, import_notes
2 "<Polygon><outerBoundaryIs><LinearRing><coordinates>
  -71.351068,-33.528545 -71.316957,-33.588208 -71.251252,-33.582507
  
```

Figura 29 Columnas contenidas en archivo csv

<sup>3</sup> <http://www.convertcsv.com/csv-to-kml.htm>

En la Figura 30 es posible ver el proceso de conversión por parte de la aplicación.

Enter records in CSV (click 'Add condition' below)

[Add condition](#)

Sort CSV <input type="checkbox"/> Ignore Case			
	Field #	Type	Direction
First By	-Choose-	-Default-	Ascending
Then By	-Choose-	-Default-	Ascending
Then By	-Choose-	-Default-	Ascending

For each line in the CSV, identify the field position (starting at 1) for this information:

Name Field #  Description Field #

Latitude Field #  Longitude Field #

Save this file on your server and reference the URL in Google Maps to display each marker. [This is the Example KML using Google Maps.](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
<Placemark>
<name><lt;Polygon>&gt;&lt;outerBoundaryIs>&gt;&lt;LinearRing>&gt;&lt;coordinates>&gt;-71.351068,-33.5285
<description>María Pinto</description>
</Placemark>
<Placemark>
<name><lt;Polygon>&gt;&lt;outerBoundaryIs>&gt;&lt;LinearRing>&gt;&lt;coordinates>&gt;-71.711538,-33.9667
<description>San Pedro</description>
</Placemark>
<Placemark>
<name><lt;Polygon>&gt;&lt;outerBoundaryIs>&gt;&lt;LinearRing>&gt;&lt;coordinates>&gt;-70.65179,-33.43278
<description>Independencia</description>
</Placemark>
```

Copyright © 2013 Data Design Group Inc. All Rights Reserved [Contact Us](#) [Change Log](#) [Terms of Use](#)

Figura 30 Conversión de CSV a KML

Ya con el proceso de conversión terminado, tenemos disponible un archivo con el estándar KML, como el mostrado en la Figura 31, que posee el nombre de la comuna y los puntos que la componen.

```
1 <name>Chilla!n</name>
2 </Placemark>
3 <Placemark>
4 <description>&lt;Polygon>&gt;&lt;outerBoundaryIs>&gt;&lt;LinearRing>&gt;&lt;coordinates>&gt;-72.217962,-36.738374
-72.123387,-36.730556 -72.057501,-36.746841 -72.096654,-36.632396 -72.138385,-36.614355 -72.264775,-36.637119
-72.301997,-36.629556 -72.357133,-36.654269 -72.359077,-36.689787&lt;/coordinates>&gt;&lt;/LinearRing>&gt;&lt;/
outerBoundaryIs>&gt;&lt;/Polygon>&gt;</description>
5 <name>Chilla!n Viejo</name>
6 </Placemark>
7 <name>San Nicola!s</name>
8 </Placemark>
9 <Placemark>
10 <description>&lt;Polygon>&gt;&lt;outerBoundaryIs>&gt;&lt;LinearRing>&gt;&lt;coordinates>&gt;-72.823243,-36.382604
-72.737268,-36.405756 -72.698233,-36.436021 -72.68949,-36.485622 -72.635329,-36.524227 -72.557437,-36.534616
-72.507869,-36.468759 -72.50842,-36.43546 -72.565756,-36.41611 -72.585112,-36.386694 -72.636029,-36.404185
-72.661467,-36.327213 -72.741183,-36.353402 -72.789633,-36.330822 -72.858276,-36.360303 -72.871102,-36.388605&lt;/
coordinates>&gt;&lt;/LinearRing>&gt;&lt;/outerBoundaryIs>&gt;&lt;/Polygon>&gt;</description>
11 <Placemark>
```

Figura 31 Archivo en estándar KML

El proceso de conversión pudiera terminar aquí, salvo que las coordenadas finales no están en el orden correcto, es decir, los valores de la longitud y latitud están intercambiados, por lo que el procedimiento final sería intercambiar estos valores, ya sea de manera manual o automatizada mediante un script en Java, solo por dar un ejemplo. Este último paso nos permitirá conseguir un

archivo similar a mostrado en la Figura 25, el cual podrá visualizarse en el mapa del sistema como se muestra en la Figura 32.

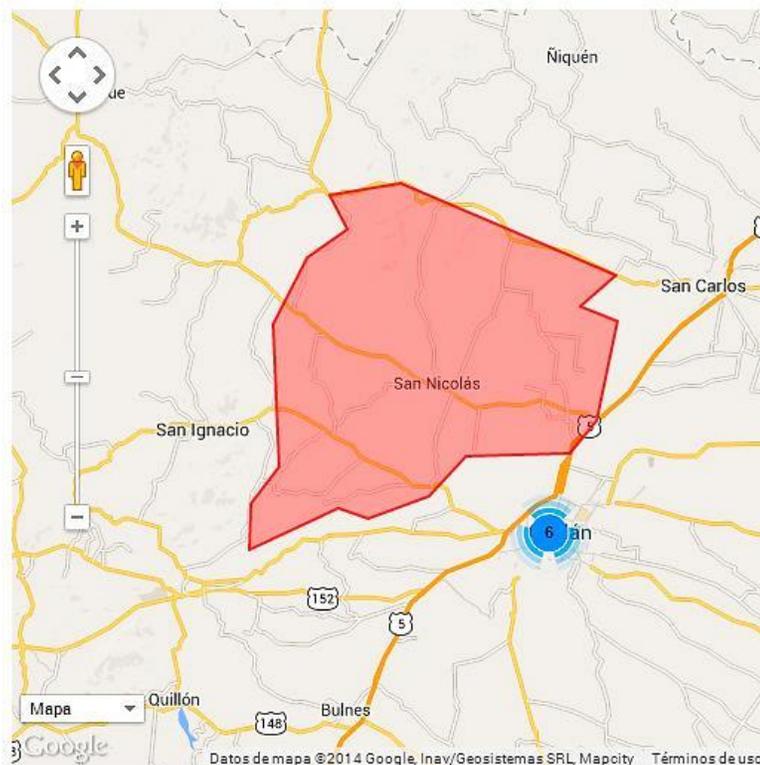


Figura 32 Visualización de un polígono en un mapa

## 6.5 Uso de Mapas.

Para la representación y el uso del API de Google Maps en un sistema web, existen varias alternativas, la mayor utiliza es el uso de JavaScript. Donde la inserción del mapa en la página web es mediante esta tecnología. Otra alternativa es el uso de extensión, que simplifican la construcción y configuración de estos.

Para este proyecto se hará uso tanto de JavaScript como de una extensión de Yii llamada EGMap.

### 6.5.1 EGMap

EGMap es una extensión para Yii [16] inspirada en Google Maps y creada por Fabrice Bernhard y que ha sufrido diversos cambios por diferentes desarrolladores. Una de las ventajas de esta librería es que provee métodos limpios para la construcción y manipulación de un mapa. La decisión de usar esta extensión radica en la facilidad para la configuración de los marcadores, globos de información, despliegue de los polígonos en el mapa.

## 6.5.2 Google Maps y JavaScript

El API de JavaScript de Google Maps permite insertar Google Maps en las páginas web. Actualmente la versión 3 es la versión oficial del API, dejando obsoleta la versión 2 [17]. Esta versión está especialmente diseñada para proporcionar una mayor velocidad y que se pueda aplicar más fácilmente tanto a móviles como a las aplicaciones de navegador de escritorio tradicionales.

El API proporciona diversas utilidades para manipular mapas y para añadir contenido al mapa mediante diversos servicios, permitiéndote crear sólidas aplicaciones de mapas en tu sitio web.

Si bien, como se vio en el punto anterior, la extensión EGMap es suficiente para proveer un mapa con la información de los estudiantes, además de poder dibujar los límites de las comunas, por lo que podría parecer redundante utilizar otro servicio de mapas, la razón de esto se debe a que para mejorar la experiencia del usuario durante el ingreso de datos, el API de JavaScript nos permite poder desplazar el marcador para ubicar el domicilio del estudiante, capturando las coordenadas para posteriormente almacenar estas.

## 6.6 Usabilidad

A continuación se presentan los conceptos de usabilidad aplicados al sitio web.

### 6.6.1 Migajas de Pan

Este concepto se refiere a que el usuario pueda saber en qué parte del sitio se encuentra y poder regresar a cualquier página en el inicio y la actual. En la Figura 33 se ilustra el concepto, una vez el usuario a navegado hasta el módulo de agregar estudiante, fácilmente puede volver a la vista anterior o a la página de inicio con solo haciendo clic en el vínculo que desee.



Figura 33 Migajas de Pan

### 6.6.2 Visibilidad del Estado del Sistema

El usuario estará constantemente informado acerca de que está haciendo en el sistema. En la Figura 34 es posible apreciar el concepto, en cada página en la cual se encuentre, el título de esta dará al usuario una noción de la finalidad del módulo. También es posible, como se ve en la Figura 35, informar al usuario de las acciones que puede realizar.



Figura 34 Visibilidad del estado del sistema

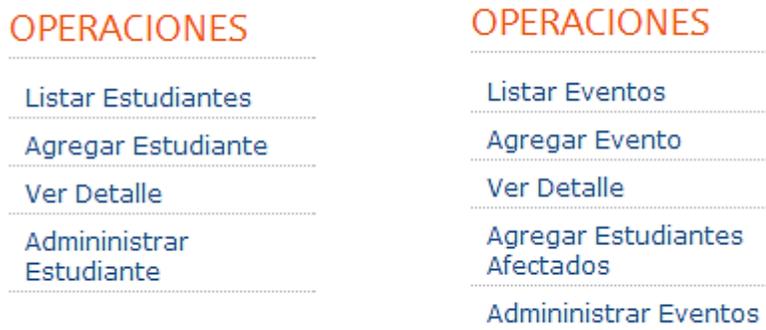


Figura 35 Visibilidad de las posibles acciones

### 6.6.3 Control por parte del Usuario.

Dado que los usuarios frecuentemente eligen funciones por error, es necesario proveer un mecanismo para que este pueda enmendar tales acciones. En la Figura 36 se muestra un dialogo de confirmación con el propósito de evitar que el usuario elimine por error información del sistema, en este caso, si usuario no deseaba elegir dicha opción, tiene la posibilidad de no eliminar el dato.

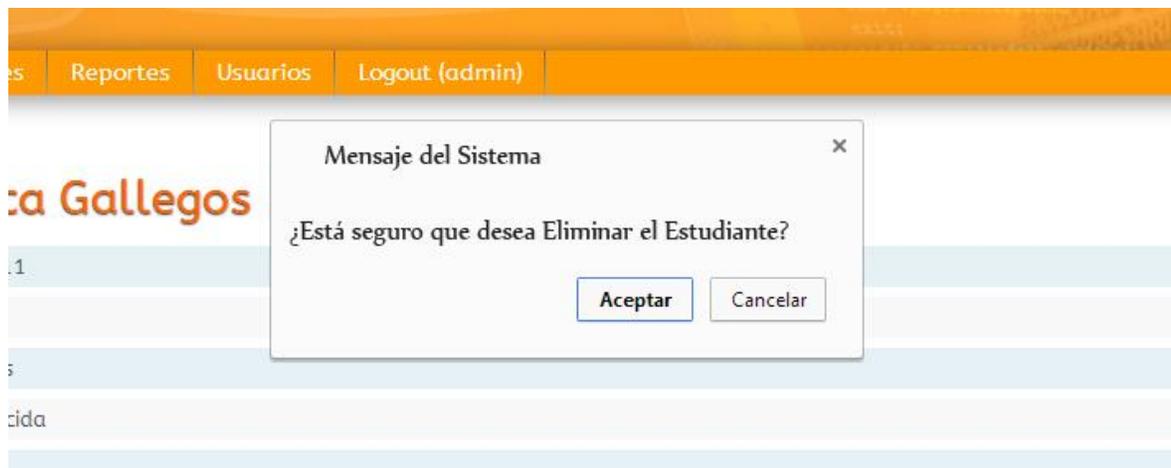


Figura 36 Confirmación de Eliminación de un Estudiante

#### 6.6.4 Similitud entre el sistema y el mundo real.

El sistema debe usar palabras y conceptos familiares para el usuario, siguiendo las convenciones usadas en el mundo real, logrando que la información parezca lógica y natural. En este caso, los conceptos que utiliza la aplicación son de conocimiento tanto de los docentes como de la administración y jefatura.

El sistema otorga la posibilidad de ubicar la dirección de un estudiante utilizando un mapa como el visto en la Figura 5, el cual posee una vista satelital, lo que provee un método bastante cercano con la realidad al momento de localizar una dirección.

#### 6.6.5 Consistencia y Estandarización

En el sistema no se utilizan nombres distintos para una misma acción, lo cual evita que los usuarios se confundan al momento de realizar un proceso o solicitar un servicio.

#### 6.6.6 Reconocer antes de Memorizar

Los usuarios no tendrán que recordar detalles de páginas anteriores para la realización de una acción, debido que esto facilita la interacción del usuario con el sistema, evitando que estos abandonen un procedimiento en el cual no estén dispuestos a recordar.

#### 6.6.7 Estética y minimalismo

En cuanto a la estética del proyecto, este fue asesorado por una diseñadora gráfica experimentada en el desarrollo web, cuyas recomendaciones se enfocaron en la presentación de la información. Los conceptos ilustrados en la Figura 14 y Figura 15, fueron resultado de las reuniones con ella. En la Figura 37 es posible apreciar el diseño de dos de los módulos previa asesoría de la diseñadora gráfica.

Ahora, dentro de las páginas no existe información que salga del contexto de las acciones a seguir, evitando así la distracción del usuario.



Figura 37 Diseños conceptuales previos a la asesoría de diseñadora gráfica.

# SEGURIDAD Y PRUEBAS

---

## 7 SEGURIDAD Y PRUEBAS

---

### 7.1 Introducción

En sus comienzos, la seguridad consistía en el uso de medios físicos, tales como cajas fuertes o armarios con cierre de seguridad. Poco después, aparecieron los medios administrativos, como lo son los contratos de empleados para salvaguardar información confidencial de la empresa.

Con la llegada de la computación, hicieron falta nuevas herramientas de seguridad diseñadas para proteger los datos y evitar la intrusión de los hackers. Estas herramientas se conocen con el nombre de “seguridad informática”.

Finalmente, con la aparición de Internet, las redes locales y los sistemas distribuidos, surgió la necesidad de disponer de seguridad durante la transmisión. Las herramientas que satisfacen estas nuevas necesidades se engloban bajo la denominación “seguridad en redes”.

Si bien la seguridad es un factor importante, que deben ser abordadas, existen, además, las pruebas de software, que son un elemento crítico para garantizar la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

La calidad de un sistema software es algo subjetivo que depende del contexto y del objeto que se pretende conseguir. Para determinar dicho nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

Por lo anterior, ambos conceptos serán revisados en el presente capítulo, a fin de que todo el trabajo invertido en el proyecto culmine en un producto de calidad y seguro para el usuario.

## 7.2 Seguridad

Debido al tipo de información que el sistema maneja sobre los estudiantes, es absolutamente necesario dedicar un tiempo para evaluar la necesidad de tener un sistema seguro, no solo referente a la estabilidad del mismo, sino a la protección de los datos que este alojará.

Según palabras de Kevin Mitnick [21], solo se necesitan unas pocas deficiencias de seguridad en una empresa para hacerse con el dominio absoluto de los sistemas informáticos.

Por lo anterior, es necesario realizar un análisis que permita disminuir al mínimo las brechas de seguridad.

### 7.2.1 Seguridad mediante Oscuridad

Si bien no se trata de un método de seguridad propiamente tal, algunas entidades suelen utilizarlos en sus sistemas. Básicamente trata de utilizar el desconocimiento de un posible atacante sobre el diseño o arquitectura del sistema. Como ejemplo de esto, podríamos suponer que un sistema web compuesto de diferentes módulos, y que para acceder a ellos se necesita una url específica. En este caso el diseño del sistema se basa en la suposición de que cualquiera que supiera las direcciones de llamada a estas partes del sitio Web sería un empleado o una persona autorizada. Otro ejemplo de una situación muy común para este tipo de seguridad, son aquellos sistemas que utilizan para su conexión puertos superiores o poco frecuentes, en los que su seguridad se basa en la baja probabilidad de que un atacante sea capaz de encontrar dicho puerto de acceso.

Como contramedida a posibles situaciones como las anteriormente mencionadas, cada módulo del sistema debe ser autenticado. Esta simple medida solo permitirá a usuarios con sesión iniciada acceder a los diferentes módulos.

### 7.2.2 Seguridad mediante Contraseñas

Un método bastante común y prácticamente conocido por todos, el uso de contraseñas es usado básicamente como método de autenticación al momento acceder a un sistema.

Este sistema presenta algunas debilidades, las contraseñas, al ser caracteres alfanuméricos en su mayoría, pueden ser descubiertas mediante diversos mecanismos.

Una medida para mejorar la seguridad mediante contraseñas, es el uso de contraseñas más robustas, ya sea mediante combinaciones de caracteres alfanuméricos, mayúsculas, minúsculas y caracteres especiales. Todo esto con la finalidad de imposibilitar a un atacante que trate de adivinar la contraseña.

También existe el uso de contraseñas dinámicas, como por ejemplo un token de seguridad, que genera de manera automática contraseña o que responde a la solicitud de combinación de caracteres específicos, como lo son las tarjetas de coordenadas, las cuales poseen una matriz con diferentes caracteres ordenados de manera aleatoria, donde el sistema solicita el ingreso de algunos de estos.

### 7.2.3 Encriptación

Si bien, hay una necesidad de bloquear todo ingreso no autorizado a un sistema, siempre existe la posibilidad de que una atacante tenga éxito. Para ello existe un nivel de seguridad cuyo fin es la protección de los datos almacenados. La encriptación busca volver ilegible toda información almacenada.

De este modo, si un atacante logra acceder a la información, no podrá hacer uso de esta, a no ser que cuente con la clave para desencriptarla.

En el caso de este proyecto, la contraseña de los usuarios será encriptada mediante MD5, que es un método de una sola vía, es decir, no existe un método inverso que permita desde el valor encriptado obtener el valor original. En su manera más simple, usando MD5, se pasa un parámetro, que podría ser la contraseña, y este genera un nuevo valor de un largo X, almacenando este valor encriptado, luego, cuando el usuario ingrese nuevamente su contraseña, esta vuelve a procesarse mediante MD5 y lo que el sistema validará es si el valor generado corresponde al almacenado.

### 7.2.4 Niveles de Acceso

Los niveles de acceso, buscan garantizar el correcto uso del sistema. Para ello generalmente se utilizan roles, los cuales diferencian los permisos que tiene cada usuario dentro del sistema. Jeffrey Winesett ejemplifica utilizando 3 niveles o tipos de rol [22]

1. Dueño del proyecto. Conocido también como súper usuario o administrador del sistema. Es aquel que tiene completo acceso total a este.
2. Miembro del proyecto. Conocido como usuario. Es aquel que en menor medida tiene acceso al sistema o su control sobre este es limitado.
3. Lector del proyecto. A veces conocido como invitado. Es aquel que puede o no tener acceso al sistema y que cuando lo tiene no puede hacer cambio alguno en este.

Esto presenta varias ventajas:

- Obliga al usuario a iniciar sesión antes de acceder al sistema.
- Permite la creación de diferentes roles y asociar estos a los permisos de cada usuario.
- Autoriza o niega el acceso a diferentes módulos del sistema o a diferentes funcionalidades de este.

### 7.2.5 Tipos de Ataque

En un sistema pueden sufrir diferentes tipos de ataque, algunos de los cuales se definen a continuación.

- **COMMAND EXECUTION:** Esta vulnerabilidad nos permite ejecutar comandos de consola desde la web, permitiendo así ver, modificar y eliminar archivos y directorios del servidor, las posibilidades son infinitas, su única limitación es el usuario que usa la consola para ejecutar los comandos, así dependería de los permisos de ese usuario para realizar ciertas acciones.
- **XSS (CROSS-SITE SCRIPTING):** Una de las vulnerabilidades más comunes en estos tiempos, consiste en "ejecutar" código HTML o Javascript en un sitio web permitiendo así cambiar

totalmente la interfaz de un sitio web (defacement). Al ejecutar scripts maliciosos puede ser perjudicial para los usuarios que accedan a la web. Se pueden encontrar dos formas de explotar esta vulnerabilidad, la primera es de forma persistente que comúnmente es cuando el código insertado es guardado en una base de datos y luego al realizar una consulta a esa base de datos se ejecuta el código. La otra es una forma indirecta conocida como “reflejada”, comúnmente se ve en sitios que pasan parámetros por vía GET (también puede ser POST). De esta forma el código insertado no se mostraría de forma persistente, pero aun así alguien podría crear una URL que ejecute el código malicioso y luego enviárselo a una persona para robarle su cookie o incluso su contraseña (Pishing).

- RFI (REMOTE FILE INCLUSION): Esta vulnerabilidad permite incluir archivos externos al sitio, donde normalmente ese archivo externo es una Shell PHP. Estructura común de una URL llamando otro archivo.
- SQL INJECTION: Como dice su nombre se trata de inyectar código SQL “intruso” en una consulta no validada correctamente, la idea principal al hacer una inyección es extraer la mayor información posible de una base de datos, aunque claro tiene otra cantidad de usos como por ejemplo iniciar sesión con la cuenta de un usuario o subir una shell al servidor.
- UNRESTRICTED FILE UPLOAD: Consiste en subir código malicioso como por ejemplo una Shell PHP por medio de un formulario que originalmente es creado para solo subir imágenes pero no filtra el tipo de archivos
- BRUTE FORCE: Un ataque por fuerza bruta se trata de enviar las combinaciones posibles para loguearse como un usuario ya sea usando un diccionario con palabras habituales o generando palabras aleatoriamente.

## 7.3 Casos de Prueba

### 7.3.1 Elementos de Prueba

#### 7.3.1.1 Módulo Estudiantes

El módulo estudiantes es aquel donde se visualiza e ingresa la información respecto a los estudiantes de la carrera. Todos los usuarios del sistema tienen acceso a este módulo, ya sea para la creación y la edición de datos. Solo el rol de administrador y jefatura de carrera tiene el permiso para la eliminación de estudiantes.

#### 7.3.1.2 Módulo Eventos

El módulo eventos es aquel donde se visualizará e ingresará información referente a los diferentes eventos que pudieran afectar a un sector del país, pudiendo ser estos incendios o catástrofes naturales. Todos los usuarios del sistema tienen acceso a este módulo, ya sea para la creación y la edición de datos. Solo el rol de administrador y jefatura de carrera tiene el permiso para la eliminación de estudiantes.

#### 7.3.1.3 Módulo Reportes

El módulo de reportes será el responsable de emitir los diferentes reportes que generará el sistema.

## 7.3.2 Especificación de las Pruebas

### 7.3.2.1 Características a Probar

En esta etapa de prueba se medirá el nivel de usabilidad, desempeño y navegación de los módulos previamente mencionados. Además, se medirá el comportamiento de seguridad del sitio frente a situaciones de vulnerabilidad

### 7.3.2.2 Nivel de Pruebas

Para las pruebas se aplicarán los niveles de unidad y aceptación de los módulos. Por parte de la seguridad se realizarán ataques al sitio de tipo Sql injection, Command execution, cross-site scripting y remote file inclusión.

### 7.3.2.3 Objetivo de la Prueba

El objetivo de las pruebas es poder detectar cualquier eventualidad que afecte el normal funcionamiento de los módulos, problemas de usabilidad, nivel de aceptación o rechazo por parte del usuario y errores respecto de las reglas del negocio que se aplicaron para el desarrollo de los módulos.

### 7.3.2.4 Enfoque

Para la definición de los casos de prueba se realizarán pruebas de caja negra.

### 7.3.2.5 Actividades de Prueba

Para llevar a cabo el plan de pruebas se debe primeramente cumplir con las condiciones adecuadas de hardware, software y conectividad, es decir, utilizar un computador (ya sea móvil o de escritorio) que cuente con un navegador de internet (en su versión más actualizada) y que tenga acceso a la red de internet.

Las actividades para las pruebas:

- Iniciar sesión (Véase Tabla 24)
- Ingresar un nuevo estudiante (Véase Tabla 25)
- Ingresar un nuevo evento (Véase Tabla 26 )
- Agregar estudiante a un evento (Véase Tabla 27)
- Modificar un estudiante (Véase Tabla 28)
- Modificar un evento (Véase Tabla 29)
- Obtener reporte (Véase Tabla 30)

Para las pruebas de seguridad se usará una herramienta llamada Damn Vulnerable Web Application (DVWA), preparada para detectar vulnerabilidades en aplicaciones y sitios web.

## 7.3.3 Responsable de las Pruebas

Para la realización de las pruebas el responsable será el mismo desarrollador del sistema, ya que el equipo de trabajo está compuesto por sólo una persona y por lo mismo tener mayor disposición a encontrar errores en el componente o ambigüedades en su especificación.

### 7.3.4 Detalles de las Pruebas

#### 7.3.4.1 Pruebas de Unidad

Definición del Caso de Prueba	
<b>ID</b>	CP-1
<b>Descripción</b>	Un usuario inicia sesión
<b>Prerrequisito</b>	Ninguno
<b>Datos de Prueba</b>	Datos de Administrativo <ul style="list-style-type: none"> <li>Rut y contraseña de prueba</li> </ul> Datos de Jefatura de Carrera <ul style="list-style-type: none"> <li>Rut y contraseña de prueba</li> </ul> Datos de Administrador <ul style="list-style-type: none"> <li>Rut y contraseña de prueba</li> </ul>
<b>Resultados esperados</b>	En caso de que el usuario ingresado no pertenezca al sistema o que los datos ingresados será incorrectos, el sistema debe alertar de esto. En caso contrario, debe reconocer el tipo de usuario e iniciar la sesión
<b>Resultados obtenidos</b>	<ul style="list-style-type: none"> <li>Se inicia sesión con datos válidos, el sistema reconoce el rol y muestra los elementos correspondientes</li> <li>Se inicia sesión con datos erróneos, el sistema muestra el mensaje "Usuario y/o Contraseña incorrectos"</li> </ul>
<b>Evaluación de la prueba</b>	El sitio responde de manera satisfactoria a las diferentes situaciones al que fue sometido

Tabla 24 Caso de Prueba Inicio de sesión

Definición del Caso de Prueba	
<b>ID</b>	CP-2
<b>Descripción</b>	Se ingresa un nuevo estudiante
<b>Prerrequisito</b>	Se ha iniciado sesión
<b>Datos de Prueba</b>	Datos de un alumno <ul style="list-style-type: none"> <li>Rut</li> <li>Nombre y Apellidos</li> <li>Dirección</li> <li>Carrera</li> <li>Sede</li> <li>Email</li> <li>Teléfono de contacto</li> <li>Coordenadas</li> </ul>
<b>Resultados esperados</b>	Si el usuario ingresa correctamente los datos requeridos, estos serán visualizados.
<b>Resultados obtenidos</b>	<ul style="list-style-type: none"> <li>Se ingresan algunos datos incorrectos, como el rut, el sistema solicita corregirlo.</li> <li>Se ingresa nombre y apellidos con caracteres inválidos, el sistema solicita corrección</li> <li>Se ingresa email y coordenadas no ingresando el formato correcto, el sistema solicita corrección</li> </ul>

	<ul style="list-style-type: none"> <li>Se ingresan los datos correctamente, el sistema muestra una nueva página con la información ingresada.</li> </ul>
<b>Evaluación de la prueba</b>	El sitio responde de manera satisfactoria a las diferentes situaciones al que fue sometido

Tabla 25 Caso de Prueba Ingresar Nuevo Estudiante

<b>Definición del Caso de Prueba</b>	
<b>ID</b>	CP-3
<b>Descripción</b>	Se ingresa un nuevo evento
<b>Prerrequisito</b>	Se ha iniciado sesión
<b>Datos de Prueba</b>	Datos de un evento <ul style="list-style-type: none"> <li>Titulo</li> <li>Comuna</li> <li>Descripción</li> </ul>
<b>Resultados esperados</b>	Si el usuario ingresa correctamente los datos requeridos, estos serán visualizados.
<b>Resultados obtenidos</b>	<ul style="list-style-type: none"> <li>Se omiten algunos campos, el sistema solicita ingresar información para proceder</li> <li>Se ingresan los datos correctamente, el sistema muestra una nueva página con la información ingresada.</li> </ul>
<b>Evaluación de la prueba</b>	El sitio responde de manera satisfactoria a las diferentes situaciones al que fue sometido

Tabla 26 Caso de Prueba Ingresar Nuevo Evento

<b>Definición del Caso de Prueba</b>	
<b>ID</b>	CP-4
<b>Descripción</b>	Agregar un estudiante a un evento
<b>Prerrequisito</b>	Seleccionar un evento creado
<b>Datos de Prueba</b>	Estudiante seleccionado y sus datos relacionados con el evento <ul style="list-style-type: none"> <li>Observaciones</li> <li>Prioridad</li> </ul>
<b>Resultados esperados</b>	Si el usuario ingresa correctamente los datos requeridos, estos serán visualizados relacionados con el evento.
<b>Resultados obtenidos</b>	<ul style="list-style-type: none"> <li>Se omiten el campo de observaciones, el sistema solicita ingresar información para proceder.</li> <li>Se ingresan los datos correctamente, el sistema muestra una nueva página con la información ingresada.</li> </ul>
<b>Evaluación de la prueba</b>	El sitio responde de manera satisfactoria a las diferentes situaciones al que fue sometido

Tabla 27 Caso de Prueba Agregar estudiante a un evento

<b>Definición del Caso de Prueba</b>	
<b>ID</b>	CP-5
<b>Descripción</b>	Modificar Estudiante
<b>Prerrequisito</b>	Seleccionar un Estudiante
<b>Datos de Prueba</b>	Estudiante seleccionado y sus nuevos datos

<b>Resultados esperados</b>	Si el usuario ingresa correctamente los datos requeridos, estos serán visualizados relacionados con el evento.
<b>Resultados obtenidos</b>	<ul style="list-style-type: none"> <li>• Se edita nombre y apellidos con caracteres inválidos, el sistema solicita corrección</li> <li>• Se edita email y coordenadas no ingresando el formato correcto, el sistema solicita corrección</li> <li>• Se editan los datos correctamente, el sistema muestra una nueva página con la información ingresada.</li> </ul>
<b>Evaluación de la prueba</b>	El sitio responde de manera satisfactoria a las diferentes situaciones al que fue sometido

Tabla 28 Caso de Prueba Modificar Estudiante

<b>Definición del Caso de Prueba</b>	
<b>ID</b>	CP-6
<b>Descripción</b>	Se modifica un evento
<b>Prerrequisito</b>	Se ha seleccionado un evento
<b>Datos de Prueba</b>	Evento seleccionado y nuevos datos
<b>Resultados esperados</b>	Si el usuario ingresa correctamente los datos requeridos, estos serán visualizados.
<b>Resultados obtenidos</b>	<ul style="list-style-type: none"> <li>• Se eliminan algunos campos, el sistema solicita ingresar información para proceder</li> <li>• Se editan los datos correctamente, el sistema muestra una nueva página con la información ingresada.</li> </ul>
<b>Evaluación de la prueba</b>	El sitio responde de manera satisfactoria a las diferentes situaciones al que fue sometido

Tabla 29 Caso de Prueba Modificar Evento

<b>Definición del Caso de Prueba</b>	
<b>ID</b>	CP-7
<b>Descripción</b>	Obtener un reporte
<b>Prerrequisito</b>	Se ha iniciado sesión
<b>Datos de Prueba</b>	Rangos de fechas inválidos
<b>Resultados esperados</b>	Si el usuario ingresa correctamente los datos requeridos, se visualizara un reporte.
<b>Resultados obtenidos</b>	<ul style="list-style-type: none"> <li>• Se ingresan fechas futuras, el sistema solicita corregir las fechas</li> <li>• Se ingresan fechas en formato inválido, el sistema solicita corregir</li> <li>• Se ingresan fechas correctas, el sistema muestra un reporte</li> </ul>
<b>Evaluación de la prueba</b>	El sitio responde de manera satisfactoria a las diferentes situaciones al que fue sometido

Tabla 30 Obtener un Reporte

### 7.3.4.2 Pruebas de Seguridad

Utilizando las opciones de la herramienta DVWA para el ataque al sitio web se puede observar en la Tabla 31:

Ataque	Resultado		
	Insatisfactorio	Regular	Satisfactorio
COMMAND EXECUTION			<b>X</b>
XSS		<b>X</b>	
RFI			<b>X</b>
SQL INJECTION			<b>X</b>
UNRESTRICTED FILE UPLOAD			<b>X</b>
BRUTE FORCE			<b>X</b>

Tabla 31 Resultado Prueba de Seguridad

### 7.3.5 Conclusiones de Pruebas

Las pruebas se han realizado bajo circunstancias normales, es decir, el sistema no se ha sometido a stress, ni ha sido instalado en un servidor de explotación adecuado, pero a pesar de ello se ha logrado obtener una respuesta satisfactoria de las pruebas realizadas, obteniendo resultados visibles y tiempos de respuesta óptimos.

El hecho de que las pruebas no se hayan realizado en un ambiente adecuado, no significa que no sea capaz de responder a una alta tasa de exigencia, ello dependerá de la configuración que posea el servidor en que finalmente se montara la aplicación. El sitio fue diseñado utilizando un framework que está preparado para una alta exigencia considerando tanto aspectos de seguridad como de lógica.

Respecto de la seguridad del sitio, es importante mencionar que las pruebas realizadas solo contemplan aspectos sencillos y básicos de seguridad, ningún sistema es impenetrable. En el caso de ataque XSS, aunque el sistema no fue vulnerado, ya que no se logró ejecutar ningún script, se considera un resultado regular ya que fue capaz de aceptar el script considerándolo una cadena de caracteres y no código ejecutable. Un resultado satisfactorio sería que se el sitio identificara la estructura de código de ejecución y no aceptará la cadena de entrada.

A pesar de ello se obtuvieron satisfactorios resultados de los distintos ataques realizados.

Concretamente el sitio en sí ha respondido adecuadamente gracias a la robustez del framework utilizado y a su estructura.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

# CONCLUSIONES Y TRABAJABOS FUTUROS

---

## 8 CONCLUSIONES

---

### 8.1 Conclusiones Finales

- Éste proyecto fue una propuesta hacia la Jefatura de Carrera, con el fin generar un nuevo proceso que permita un rápido acceso a la información referente a la residencia y contacto de un estudiante en situaciones de catástrofe. Otorgando una mejor interpretación de los datos mediante el uso de mapas con la ubicación de los estudiantes y zonas demarcadas, dando una percepción de la información más cercana a la realidad.
- Considerando los objetivos mencionados al principio del proyecto, se pueden considerar cumplidos con gran éxito. Señalando que el producto final es mayor a lo propuesto inicialmente.
- La etapa de diseño junto con la captura de requerimientos se convirtió en fundamental, ya que tras cada reunión en que se presentaba el diseño de los requisitos obtenidos previamente, fue posible añadir mejoras al proyecto, entregándole a este más valor. Recalco en esta etapa el proceso de mejora continua que tuvo el sistema, otorgado por las reuniones con Jefatura de Carrera y la diseñadora gráfica que asesoró el proyecto.
- En diseño del sistema web fue completamente necesario aplicar los conceptos de usabilidad, dado que la preocupación por la experiencia del usuario ayuda a que el producto tenga una aceptación y un futuro. Recalcando en este punto que una página debe ser lo más simple de utilizar posible, utilizando las palabras de Nielsen, ojalá que esta no necesite documentación para su uso.
- En cuanto a las tecnologías utilizadas, el framework Yii permitió facilitar la etapa de implementación, principalmente en la construcción de los módulos de mantenedores. Su enfoque en el MVC y la posibilidad de utilizar extensiones ayudan a la construcción de un gran producto en poco tiempo.
- Si bien para la utilización de mapas se optó por Google Maps, API que ha mantenido una mejora continua, la mayor parte de la documentación existente y que podría ayudar al proyecto data de hace 2 o más años, por lo que algunas funcionalidades necesitaron la combinación de tecnologías.
- Aunque gran parte de la implementación presentó un desafío al desconocer la tecnología, fue la conversión de los mapas vectoriales con las divisiones geo-políticas de Chile, lo que produjo más de algún contratiempo, al no encontrarse toda la información en un solo sitio.
- Este proyecto constituye un aporte en cuanto estimula y fortalece el desarrollo de software, integrando distintas tecnologías, buscando diferentes métodos para una solución, desafiando a que el desarrollador piense en cual mejora que de aun más valor al producto.

## 8.2 Trabajos Futuros

De los resultados y experiencia obtenidos por el trabajo realizado se abren sin lugar a dudas un conjunto amplio ideas. Entre ellas se mencionan las que parecen ser de mayor interés y proyección:

- Automatización del ingreso de datos de nuevos alumnos.
- Incorporación de carga masiva de datos.
- Incorporar información respecto al estado actual del estudiante (Egresado, Terminal, etc)
- Implementación para direcciones en el extranjero
- Implementación de servicios para otros sistemas de la Carrera
- Incorporar otras carreras

La tecnología y la implementación del proyecto da lugar a dos posibles áreas de explotación, las cuales se listan a continuación:

1. Implementación de un sistema similar enfocado al área de Bienestar Estudiantil de la universidad, con la finalidad de apoyar a las asistentes sociales en la planificación de visitas a los estudiantes, las cuales realizan al menos una vez durante los primeros 2 años y que actualmente realizan mediante planillas que contienen las direcciones y que ellas revisan una por una en Google Maps.
2. El desarrollo un sistema que utilice tecnología similar, que permita a los dueños de PYMES indicar en un mapa común la ubicación de sus locales o negocios, generando así una aplicación pública que contenga información de muchos sectores comerciales y que pueda ser filtrado y accedido mediante la web o aplicación móvil.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

# BIBLIOGRAFÍA

---

## BIBLIOGRAFÍA

---

1. Arauque Ibañez, Antonio.  
Sistema de Información Geográfica para la Mejora de la Gestión y la Toma de Decisiones Difusa en Entornos Oleícolas.  
Universidad de Jaén.  
2012.
2. Utilidad de los SIGs [En línea]  
<[http://sig.cea.es/utilidad\\_SIG](http://sig.cea.es/utilidad_SIG)>  
[Consulta: 15 de Marzo de 2014]
3. ¿Qué es un SIG? [En línea]  
<<http://langleruben.wordpress.com/%C2%BFque-es-un-sig/>>  
[Consulta: 15 de Marzo de 2014]
4. Neumann, Andreas.  
Web Mapping and Web Cartography.  
2008.
5. Google Maps. [En línea]  
<<http://maps.google.com/support/bin/static.py?hl=es&page=guide.cs&guide=21670&from=21670&rd=1.>>  
[Consulta: 24 de Marzo de 2014.]
6. Canepa, Juan Pablo; Recabarren, Mattías.  
Patrones de diseño Modelo-Vista-Controlador. (MVC), IIC1222 – Programación Avanzada. [En línea]  
Pontificia Universidad Católica de Chile. 2008.  
<[http://svn.assembla.com/svn/ImageShockPro/1213061556\\_aecheve\\_sec1\\_pos0.pdf](http://svn.assembla.com/svn/ImageShockPro/1213061556_aecheve_sec1_pos0.pdf)>  
[Consulta: 10 de Marzo de 2014.]
7. Gilfillan, Ian.  
La Biblia de MySQL.  
Anaya Multimedia.  
2003.
8. Martínez, Rafael.  
Manual de PHP. 2008  
<<http://www.matematica.ciens.ucv.ve/files/Manuales/Manuales/Programacion%20Web%20-%20Manual%20de%20PHP.pdf>>  
[Consulta: 15 de Marzo de 2014]

9. Yii. [En línea]  
<<http://www.yiiframework.com/about/>>  
[Consulta: 15 de Marzo de 2014]
10. Tang, Agata; Clark, Kristin  
Geocoding Rule Base Developer Guide. 2003  
<[webhelp.esri.com/arcgisdesktop/9.1/pdf/Geocoding\\_Rule\\_Base\\_Developer\\_Guide.pdf](http://webhelp.esri.com/arcgisdesktop/9.1/pdf/Geocoding_Rule_Base_Developer_Guide.pdf)>  
[Consulta: 19 de Marzo de 2014]
11. P. Christen, T. Churches y J. X. Zhu  
“Probabilistic name and address cleaning and standardization”. The Australasian Data Mining Conference  
2002.
12. SIIT [En línea]  
<<http://siit2.bcn.cl/acercade>>  
[Consultado: 5 de Mayo de 2014]
13. ShapeFile [En línea]  
<<http://help.arcgis.com/es/arcgisdesktop/10.0/help/index.html#//005600000002000000>>  
[Consultado: 15 de Mayo de 2014]
14. Fusion Tables [En línea]  
<<https://support.google.com/fusiontables/answer/2571232>>  
[Consultado: 6 de Mayo de 2014]
15. CSV [En línea]  
<<http://es.wikipedia.org/wiki/CSV>>  
[Consultado: 15 de Mayo de 2014]
16. EGMap [En línea]  
<<http://www.yiiframework.com/extension/egmap/#hh0>>  
[Consultado: 12 de Abril de 2014]
17. JavaScript para Google Maps [En línea]  
<<https://developers.google.com/maps/documentation/javascript/?hl=es>>  
[Consultada: 8 de Abril de 2014]
18. Carvallo, Yusneyi  
Programación Orientada a Objetos  
Recopilación Prof Yusneyi Carvallo  
2007
19. Sommerville, Iam  
Ingeniería del Software  
Séptima Edición

20. Larman, Craig  
UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado  
Segunda Edición: Prentice Hall  
2003
21. Mitnik, Kevin; Simon, William L.  
El Arte de la intrusión  
Primera Edición: AlfaOmega Grupo Editor  
2007
22. Winesett, Jeffrey  
Web Application Development with Yii and PHP  
Segunda Edición: Prentice Hall  
2012
23. Pérez Ruiz, Jorge Marcelo  
Mejoramiento en el nivel de usabilidad del sitio web DAI  
2004
24. Nielsen, Jakob  
Usabilidad: Diseño de Sitios Web  
Prentice Hall  
2002
25. Elmasri, Ramez  
Fundamentos de sistemas de bases de datos.  
Addison-Wesley  
2002
26. Curtis, Gregory.  
The Cave Painters: Probing the Mysteries of the World's First Artists  
Anchor  
2007
27. Cerda Lorca, Jaime; Valdivia C, Gonzalo  
John Snow, la epidemia de cólera y el nacimiento de la epidemiología moderna  
<<http://www.scielo.cl/pdf/rci/v24n4/art14.pdf>>  
2007

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

## ANEXO A

# El API de codificación geográfica de Google

## Solicitudes de codificación geográfica

Las solicitudes del API de codificación geográfica deben tener el siguiente formato:

<http://maps.googleapis.com/maps/api/geocode/output?parameters>

En esta solicitud, *output* debe ser uno de los valores que se indican a continuación:

- *json* (recomendado): indica el formato de salida en Notación de objetos JavaScript (JavaScript Object Notation, JSON).
- *xml*: indica el formato de salida como un archivo XML.

Para acceder al API de codificación geográfica de Google a través de HTTPS, se utiliza la siguiente dirección:

<https://maps.googleapis.com/maps/api/geocode/output?parameters>

Se recomienda utilizar HTTPS para aplicaciones que incluyan datos de usuario sensibles, como una ubicación de usuario, en las solicitudes.

En cualquier caso, hay algunos parámetros que son obligatorios y otros opcionales. Como en las URL estándar, todos los parámetros se separan con el carácter &. A continuación, se indican los parámetros admitidos y sus posibles valores.

### Parámetros obligatorios

- *address*: es la dirección que quieres codificar de forma geográfica.
  - o *latlng*: es el valor textual de latitud y longitud para el que deseas obtener la dirección interpretable por humanos más cercana.
  - o *components*: un filtro de componente para el que quieres obtener un código geográfico. Para obtener más información. El filtro de componentes también se acepta como un parámetro opcional si se proporciona *address*.
- *sensor*: indica si la solicitud de codificación geográfica procede de un dispositivo con un sensor de ubicación. Este valor debe ser true o false.

Los usuarios del API de Google Maps for Business deben incluir parámetros *client* y *signature* válidos junto con sus solicitudes de codificación geográfica.

### Parámetros opcionales

- *bounds*: es el cuadro delimitador de la ventana gráfica en el que quieres predeterminar los resultados de codificación geográfica con mayor importancia. Este parámetro solo afectará a los resultados del geocoder (aunque puede que no se limite exclusivamente a este elemento).
- *language*: indica el idioma en el que se deben mostrar los resultados. Tener en cuenta que esta lista puede no ser exhaustiva, ya que se actualizan los idiomas admitidos con

frecuencia. Si no se proporciona *language*, el geocoder intentará utilizar el idioma nativo del dominio desde el que se envió la solicitud (siempre que sea posible).

- *region*: es el código de país, especificado como un valor de dos caracteres *ccTLD* ("dominio de nivel superior"). Este parámetro solo afectará a los resultados del *geocoder* (aunque puede que no se limite exclusivamente a este elemento)
- *components*: los filtros de componente separados por una pleca (|). Cada filtro de componente está compuesto por un par de *component:value* y solo afectan a los resultados del geocoder.

## Respuestas de codificación geográfica

Las respuestas de codificación geográfica se devuelven en el formato que indica la marca de output en la ruta de la solicitud de la URL.

### Formatos de salida JSON

En este ejemplo, el API de codificación geográfica solicita una respuesta en formato json para una consulta sobre la dirección "1600 Amphitheatre Parkway, Mountain View, CA":

*http://maps.googleapis.com/maps/api/geocode/json?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=true\_or\_false*

Se ha dejado el parámetro *sensor* como una variable *true\_or\_false* para hacer hincapié en que el usuario debe definir este valor en *true* o en *false* de forma explícita.

A continuación, se muestra la respuesta en formato JSON que ha devuelto la solicitud. Tener en cuenta que el formato JSON actual puede contener menos espacios en blanco. No debes hacer suposiciones sobre la cantidad o el formato de espacios en blanco existente entre las solicitudes.

```
{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "1600",
          "short_name" : "1600",
          "types" : [ "street_number" ]
        },
        {
          "long_name" : "Amphitheatre Pkwy",
          "short_name" : "Amphitheatre Pkwy",
          "types" : [ "route" ]
        },
        {
          "long_name" : "California",
          "short_name" : "CA",
          "types" : [ "administrative_area_level_1", "political" ]
        },
        {
          "long_name" : "United States",
          "short_name" : "US",
```

```

        "types" : [ "country", "political" ]
      },
      {
        "long_name" : "94043",
        "short_name" : "94043",
        "types" : [ "postal_code" ]
      }
    ],
    "formatted_address" : "1600 Amphitheatre Pkwy, Mountain View, CA 94043,
USA",
    "geometry" : {
      "location" : {
        "lat" : 37.42291810,
        "lng" : -122.08542120
      },
      "location_type" : "ROOFTOP",
      "viewport" : {
        "northeast" : {
          "lat" : 37.42426708029149,
          "lng" : -122.0840722197085
        },
        "southwest" : {
          "lat" : 37.42156911970850,
          "lng" : -122.0867701802915
        }
      }
    },
    "types" : [ "street_address" ]
  }
],
"status" : "OK"
}

```

Debemos tener en cuenta que la respuesta JSON contiene dos elementos raíz:

- **"status"**: contiene los metadatos de la solicitud.
- **"results"** contiene una matriz de información geométrica y de direcciones codificadas geográficamente.

Normalmente, solo se devuelve una entrada de la matriz **"results"** para las búsquedas de direcciones, aunque el *geocoder* puede devolver varios resultados cuando las consultas de direcciones son ambiguas.

Tener en cuenta que, normalmente, estos resultados se deben analizar si queremos extraer valores de ellos. El análisis de JSON es relativamente sencillo.

### Formatos de salida XML

En este ejemplo, el API de codificación geográfica solicita una respuesta xml para la misma consulta indicada anteriormente de "1600 Amphitheatre Parkway, Mountain View, CA":

[http://maps.googleapis.com/maps/api/geocode/xml?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=true\\_or\\_false](http://maps.googleapis.com/maps/api/geocode/xml?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=true_or_false)

A continuación, se muestra la respuesta en formato XML a esta solicitud.

Sistema Web de Geolocalización de Direcciones de Estudiantes carrera  
Ingeniería Civil en Informática

```

<GeocodeResponse>
  <status>OK</status>
  <result>
    <type>street_address</type>
    <formatted_address>1600 Amphitheatre Pkwy, Mountain View, CA 94043,
USA</formatted_address>
    <address_component>
      <long_name>1600</long_name>
      <short_name>1600</short_name>
      <type>street_number</type>
    </address_component>
    <address_component>
      <long_name>Amphitheatre Pkwy</long_name>
      <short_name>Amphitheatre Pkwy</short_name>
      <type>route</type>
    </address_component>
    <address_component>
      <long_name>Mountain View</long_name>
      <short_name>Mountain View</short_name>
      <type>locality</type>
      <type>political</type>
    </address_component>
    <address_component>
      <long_name>San Jose</long_name>
      <short_name>San Jose</short_name>
      <type>administrative_area_level_3</type>
      <type>political</type>
    </address_component>
    <address_component>
      <long_name>Santa Clara</long_name>
      <short_name>Santa Clara</short_name>
      <type>administrative_area_level_2</type>
      <type>political</type>
    </address_component>
    <address_component>
      <long_name>California</long_name>
      <short_name>CA</short_name>
      <type>administrative_area_level_1</type>
      <type>political</type>
    </address_component>
    <geometry>
      <location>
        <lat>37.4217550</lat>
        <lng>-122.0846330</lng>
      </location>
      <location_type>ROOFTOP</location_type>
      <viewport>
        <southwest>
          <lat>37.4188514</lat>
          <lng>-122.0874526</lng>
        </southwest>
        <northeast>
          <lat>37.4251466</lat>
          <lng>-122.0811574</lng>
        </northeast>
      </viewport>
    </geometry>
  </result>
</GeocodeResponse>

```

Tener en cuenta que la respuesta XML está formada por un único `<GeocodeResponse>` y por dos elementos de nivel superior:

- `<status>`: contiene los metadatos de la solicitud. Consulta la sección sobre códigos de estado que aparece a continuación.
- Cero o más elementos `<result>`, que contienen un único conjunto de información geométrica y de direcciones codificadas geográficamente.
- 

Considerar que esta respuesta es mucho más larga que la respuesta JSON. Por ello, se recomienda utilizar `json` como la marca de salida preferida (a menos que el servicio deba utilizar `xml` por cualquier motivo). Además, debemos prestar atención al procesar árboles XML para asegurarnos de hacer referencia a los elementos y nodos adecuados.

- Los resultados XML se incluyen en un elemento de respuestas `<GeocodeResponse>` raíz.
- Los resultados JSON indican entradas con varios elementos mediante varios conjuntos (`types`), mientras que los resultados XML lo hacen mediante varios elementos en singular (`<type>`).
- Los elementos en blanco se muestran en forma de conjuntos vacíos en los resultados JSON, mientras que en los resultados XML están ausentes. Por ejemplo, una respuesta que no genera ningún resultado, devolverá una matriz `results` vacía en JSON y ningún elemento `<result>` en XML.

## Códigos de estado

El campo `"status"` dentro del objeto de respuesta de codificación geográfica contiene el estado de la solicitud y puede incluir información de depuración para ayudar a descubrir el motivo por el que no funciona la codificación geográfica. El campo `"status"` puede contener los siguientes valores:

- `"OK"` indica que no se ha producido ningún error; la dirección se ha analizado correctamente y se ha devuelto al menos un código geográfico.
- `"ZERO_RESULTS"` indica que la codificación geográfica se ha realizado correctamente pero no ha devuelto ningún resultado. Esto puede ocurrir si en la codificación geográfica se incluye una dirección (`address`) inexistente o un valor `latlng` en una ubicación remota.
- `"OVER_QUERY_LIMIT"` indica que se ha excedido el cupo de solicitudes.
- `"REQUEST_DENIED"` indica que la solicitud se ha denegado; normalmente se debe a la ausencia de un parámetro sensor.
- `"INVALID_REQUEST"` normalmente indica que no se ha especificado la solicitud (`address` o `latlng`).

## Resultados

Cuando el `geocoder` devuelve resultados, los coloca dentro de una matriz `results` (JSON). Aunque el `geocoder` no devuelva ningún resultado (por ejemplo, si no existe la dirección) seguirá devolviendo una matriz `results` vacía (las respuestas XML están formadas por cero o varios elementos `<result>`).

A continuación, se indican los campos de los que se suele componer un resultado:

- La matriz *types*[] indica el tipo del resultado devuelto. Este conjunto incluye una o más etiquetas que identifican el tipo de elemento que se ha obtenido en el resultado. Por ejemplo, un código geográfico de "Chicago" devuelve "*locality*", lo que indica que "Chicago" es una ciudad y además devuelve "*political*", lo que significa que es una entidad política.
- *formatted\_address* es una cadena que contiene la dirección interpretable por humanos de la ubicación. Esta dirección suele corresponder con la "dirección postal", que a veces difiere de un país a otro (tener en cuenta que algunos países, como Reino Unido, no permiten la distribución de direcciones postales reales debido a restricciones de la licencia). Esta suele estar formada por uno o varios componentes de la dirección. Por ejemplo, la dirección "111 8<sup>th</sup> Avenue, New York, NY" contiene componentes independientes para "111" (el número de la calle), para "8<sup>th</sup> Avenue" (una ruta), para "New York" (la ciudad) y para "NY" (el estado de EE.UU.). Estos componentes de la dirección contienen información adicional, tal como se indica a continuación.
- *address\_components*[] es un conjunto que incluye los diferentes componentes de la dirección, tal como se ha explicado anteriormente. A continuación, se indica lo que suele contener cada *address\_component*.
  - *types*[] es un conjunto que indica el tipo de componente de la dirección.
  - *long\_name* es la descripción completa o el nombre completo del componente de la dirección tal como lo ha devuelto el geocoder.
  - *short\_name* es un nombre textual abreviado del componente de la dirección (si está disponible). Por ejemplo, un componente de la dirección del estado de Alaska puede incluir un *long\_name* de "Alaska" y un *short\_name* de "AK" mediante la abreviación postal de dos letras.

Debemos considerar que *address\_components*[] puede contener más componentes de la dirección (además de los indicados en *formatted\_address*).

- *geometry* contiene la siguiente información:
  - *location* contiene el valor de latitud y longitud codificado de forma geográfica. Este campo suele ser el más importante en las búsquedas de direcciones normales.
  - *location\_type* almacena datos adicionales sobre la ubicación especificada. Actualmente, se admiten los siguientes valores:
    - "*ROOFTOP*" indica que el resultado devuelto es un código geográfico preciso para el que disponemos de información de ubicación correcta hasta la precisión de la dirección.
    - "*RANGE\_INTERPOLATED*" indica que el resultado obtenido refleja una aproximación (normalmente en una carretera) interpolada entre dos puntos precisos (por ejemplo, intersecciones). Los resultados interpolados se suelen obtener cuando los códigos geográficos de la parte superior no están disponibles para una dirección postal.

- *"GEOMETRIC\_CENTER"* indica que el resultado obtenido corresponde al centro geométrico de un resultado del tipo polilínea (por ejemplo, una calle) o polígono (una región).
  - *"APPROXIMATE"* indica que el resultado obtenido es aproximado.
- *viewport* contiene la ventana gráfica recomendada para mostrar el resultado que se devuelve, especificado como dos valores de latitud y de longitud que definen los extremos *southwest* y *northeast* del cuadro delimitador de la ventana gráfica. Normalmente, la ventana gráfica se utiliza para enmarcar un resultado que se muestra a un usuario.
- *bounds* (se devuelve de forma opcional) almacena el cuadro delimitador que pueden contener por completo el resultado obtenido. Considerar que estos límites pueden no coincidir con la ventana gráfica recomendada (por ejemplo, San Francisco incluye Los Farallones, que técnicamente forman parte de la ciudad pero no se muestran en la ventana gráfica).
- *partial\_match* indica que el *geocoder* no proporciona un resultado que coincida exactamente con la solicitud original, aunque coincide parcialmente con la dirección solicitada. En estos casos se recomienda revisar la solicitud original por si hubiera errores de ortografía o una dirección incompleta. Las coincidencias parciales se suelen producir en el caso de calles que no existen en la localidad transmitida en la solicitud.

### Tipos de componentes de la dirección

El conjunto *types[]* del resultado obtenido indica el tipo de dirección. Estos tipos también pueden aparecer dentro de los *conjuntos address\_components[]* para indicar el tipo del componente de la dirección en concreto. Las direcciones del *geocoder* pueden tener diversos tipos (los tipos se pueden considerar "etiquetas"). Por ejemplo, muchas ciudades pueden estar etiquetadas con los tipos *political* y *locality*.

A continuación, se indican los tipos que admite y muestra el *geocoder* de HTTP.

- *street\_address* indica una dirección postal precisa.
- *route* indica una carretera identificada (por ejemplo, "US 101").
- *intersection* indica una intersección importante, normalmente de dos carreteras importantes.
- *political* indica una entidad política. Este tipo suele indicar un polígono de alguna administración civil.
- *country* indica la entidad política nacional y, normalmente, es el tipo de mayor escala que devuelve el *geocoder*.
- *administrative\_area\_level\_1* indica una entidad civil de primer nivel por debajo del nacional. En Estados Unidos, las entidades que corresponden a este nivel administrativo son los estados. No todos los países cuentan con este nivel administrativo.
- *administrative\_area\_level\_2* indica una entidad civil de segundo nivel por debajo del nacional. En Estados Unidos, las entidades que corresponden a este nivel administrativo son los condados. No todos los países cuentan con este nivel administrativo.
- *administrative\_area\_level\_3* indica una entidad civil de tercer nivel por debajo del nacional. Este tipo indica una división administrativa de menor tamaño. No todos los países cuentan con este nivel administrativo.
- *colloquial\_area* indica un nombre alternativo usado con frecuencia para designar la entidad.

- *locality* indica una entidad política equivalente al municipio.
- *sublocality* indica una entidad política de primer nivel por debajo del municipal.
- *neighborhood* indica un barrio identificado.
- *premise* indica una ubicación identificada. Normalmente se trata de un edificio o de un complejo de edificios con un nombre común.
- *subpremise* indica una entidad de primer nivel por debajo del de ubicación identificada. Normalmente se trata de un edificio individual dentro de un complejo de edificios con un nombre común.
- *postal\_code* indica un código postal utilizado para enviar correo postal dentro del país.
- *natural\_feature* indica un paraje natural destacado.
- *airport* indica un aeropuerto.
- *park* indica un parque identificado.
- *point\_of\_interest* indica un lugar de interés identificado. Normalmente, estos lugares de interés son entidades locales importantes que no encajan con facilidad en otras categorías, como el "Edificio Empire State" o la "Estatua de la Libertad".

Además de los tipos anteriores, los componentes de la dirección pueden presentar los siguientes tipos:

- *post\_box* indica un apartado postal específico.
- *street\_number* indica la calle y el número exacto del edificio.
- *floor* indica la planta del edificio.
- *room* indica la puerta dentro del edificio.

## Codificación geográfica inversa (búsqueda de direcciones)

El término codificación geográfica suele hacer referencia al proceso de traducir una dirección interpretable por humanos en una ubicación de un mapa. El proceso de conversión o traducción de una ubicación en una dirección interpretable por humanos se conoce como codificación geográfica inversa.

El API de codificación geográfica admite la codificación geográfica inversa directamente mediante el parámetro *latlng*. Por ejemplo, la siguiente consulta contiene el valor de latitud/longitud de una ubicación de Brooklyn:

[http://maps.googleapis.com/maps/api/geocode/json?latlng=40.714224,-73.961452&sensor=true\\_or\\_false](http://maps.googleapis.com/maps/api/geocode/json?latlng=40.714224,-73.961452&sensor=true_or_false)

Un segmento de la respuesta es el siguiente:

```
{
  "status": "OK",
  "results": [ {
    "types": street_address,
    "formatted_address": "275-291 Bedford Ave, Brooklyn, NY 11211, USA",
    "address_components": [ {
      "long_name": "275-291",
      "short_name": "275-291",
      "types": street_number
    },
  ],
```

Las "*formatted\_addresses*" del resultado no son solo direcciones postales, sino cualquier forma de definir de forma geográfica una ubicación. Por ejemplo, al codificar de forma geográfica un punto de la ciudad de Chicago, el punto codificado se puede etiquetar como la dirección de una calle, como la ciudad (Chicago), como su estado (Illinois) o como un país (Estados Unidos). Todas ellas son "direcciones" para el geocoder. El geocoder inverso devuelve cualquiera de estos tipos como resultados válidos.

El geocoder inverso encuentra coincidencias con entidades políticas (países, provincias, ciudades y barrios), con direcciones y con códigos postales.

A continuación, se indica la lista completa de los valores *formatted\_address* que ha devuelto la consulta anterior.

```
"formatted_address": "275-291 Bedford Ave, Brooklyn, NY 11211, USA",
"formatted_address": "Williamsburg, NY, USA",
"formatted_address": "New York 11211, USA",
"formatted_address": "Kings, New York, USA",
"formatted_address": "Brooklyn, NY, USA",
"formatted_address": "New York, NY, USA",
"formatted_address": "New York, USA",
"formatted_address": "United States"
```

Generalmente, las direcciones se devuelven en un orden que va de las más específicas a las menos específicas; la dirección más exacta es el resultado más importante, como se ve en este caso. Hay que tener en cuenta que se devuelven diferentes tipos de direcciones, desde la dirección postal más específica hasta entidades políticas menos específicas como barrios, ciudades, países, estados, etc. Si queremos encontrar coincidencias para una dirección más general, debemos inspeccionar el campo "*types*" de las Placemark devueltas.

## Cómo especificar la ventana gráfica

También se puede indicar al servicio de codificación geográfica que establezca una ventana gráfica determinada (expresada como cuadro delimitador) como preferencia. Para hacerlo, se debe definir el parámetro *bounds* dentro de la URL de solicitud. Hay que tener en cuenta que la especificación solo tiene preferencia por los resultados dentro de los límites, por lo que si existen resultados más relevantes fuera de estos límites, es posible que se incluyan.

El parámetro *bounds* define las coordenadas de latitud/longitud de las esquinas sudoeste y noreste del cuadro delimitador mediante una pleca (|) para separar las coordenadas.

Por ejemplo, si se añade un argumento *bounds* en el que se defina un cuadro delimitador para el Valle de San Fernando de Los Ángeles, los resultados de la codificación geográfica devolverán el barrio llamado "Winnetka" de esa ubicación:

```
http://maps.googleapis.com/maps/api/geocode/json?address=Winnetka&bounds=34.172684,-118.604794|34.236144,-118.500938&sensor=false
```

Retornará:

```
{
  "status": "OK",
  "results": [ {
    "types": [ "sublocality", "political" ],
    "formatted_address": "Winnetka, California, USA",
    "address_components": [ {
      "long_name": "Winnetka",
      "short_name": "Winnetka",
      "types": [ "sublocality", "political" ]
    }, {
      "long_name": "Los Angeles",
      "short_name": "Los Angeles",
      "types": [ "administrative_area_level_3", "political" ]
    }, {
      "long_name": "Los Angeles",
      "short_name": "Los Angeles",
      "types": [ "administrative_area_level_2", "political" ]
    }, {
      "long_name": "California",
      "short_name": "CA",
      "types": [ "administrative_area_level_1", "political" ]
    }, {
      "long_name": "United States",
      "short_name": "US",
      "types": [ "country", "political" ]
    } ],
    "geometry": {
      "location": {
        "lat": 34.2131710,
        "lng": -118.5710220
      },
      "location_type": "APPROXIMATE",
      "viewport": {
        "southwest": {
          "lat": 34.1947148,
          "lng": -118.6030368
        },
        "northeast": {
          "lat": 34.2316232,
          "lng": -118.5390072
        }
      }
    },
    "bounds": {
      "southwest": {
        "lat": 34.1791050,
        "lng": -118.5883200
      },
      "northeast": {
        "lat": 34.2353090,
        "lng": -118.5534191
      }
    }
  } ]
}
```

En esta respuesta se puede apreciar que el segmento "bounds" contiene:

```

"bounds": {
  "southwest": {
    "lat": 34.1791050,
    "lng": -118.5883200
  },
  "northeast": {
    "lat": 34.2353090,
    "lng": -118.5534191
  }
}

```

Donde los valores retornados para la latitud y longitud se acercan a los seteados en la petición,  
 bounds=34.172684,-118.604794|34.236144,-118.500938

Por el contrario, si se hubiese omitido dichos parámetros, el segmento “*bounds*” de la respuesta hubiera sido:

```

"bounds": {
  "southwest": {
    "lat": 42.0885320,
    "lng": -87.7715480
  },
  "northeast": {
    "lat": 42.1284090,
    "lng": -87.7110160
  }
}

```

Claramente se distingue la diferencia en el tamaño de la ventana grafica de retorno.

## Cómo filtrar componentes

El API de codificación geográfica de Google puede devolver resultados de direcciones de un área específica. Esta restricción se especifica con el filtro *components*. Este filtro está compuesto por una lista de pares de *component:value* separados por una pleca (|). Solo se devuelven los resultados que coinciden con todos los filtros. Los valores del filtro admiten los mismos métodos de corrección ortográfica y de coincidencia parcial que otras solicitudes de codificación geográfica. Si un resultado de codificación geográfica coincide parcialmente con un filtro de componente, incluirá un campo *partial\_match* en la respuesta.

A continuación se indican los componentes (*components*) que se pueden filtrar.

- *route* coincide con el nombre corto o largo de una ruta.
- *locality* coincide con los tipos *locality* y *sublocality*.
- *administrative\_area* coincide con todos los niveles de *administrative\_area*.
- *postal\_code* coincide con *postal\_code* y *postal\_code\_prefix*.
- *country* coincide con el nombre de un país o con un código de país ISO 3166-1 de dos letras.

Una consulta que incluya un filtro de componente solo devolverá los resultados de codificación geográfica que coincidan con el filtro. Si no existe ninguna coincidencia, el *geocoder* devolverá un resultado que coincida con el filtro en sí.

## ANEXO B

# Servicio de asignación de identificadores geográficos

## Descripción general

La codificación geográfica es el proceso de transformar direcciones (como "1600 Amphitheatre Parkway, Mountain View, CA") en coordenadas geográficas (como 37.423021 de latitud y -122.083739 de longitud), que se pueden utilizar para colocar marcadores o situar el mapa.

El API de Google Maps proporciona una clase *geocoder* que permite codificar geográficamente las direcciones de forma dinámica a partir de los datos introducidos por el usuario. Estas solicitudes son limitadas para evitar un abuso del servicio.

## Solicitudes de codificación geográfica

El acceso al servicio de asignación de identificadores geográficos tiene lugar de forma asíncrona, dado que el API de Google Maps necesita realizar una llamada a un servidor externo. Por ese motivo, se debe incluir un método de devolución de llamada para que se ejecute al completar la solicitud. Este método de devolución de llamada procesará los resultados. Tener en cuenta que el *geocoder* puede devolver más de un resultado.

Debemos acceder al servicio de asignación de identificadores geográficos del API de Google Maps desde un código a través del objeto *google.maps.Geocoder*. El método *Geocoder.geocode()* inicia la solicitud al servicio de asignación de identificadores geográficos y transmite un objeto literal *GeocodeRequest* que contiene los términos de entrada y un método de devolución de llamada que se ejecutará al recibir la respuesta.

El objeto literal *GeocodeRequest* contiene los siguientes campos:

```
{
  address: string,
  latLng: LatLng,
  bounds: LatLngBounds,
  region: string
}
```

A continuación, se explican estos campos.

- *address* (obligatorio\*): es la dirección que se quiere codificar de forma geográfica.
- *latLng* (obligatorio\*): es la latitud y la longitud (LatLng) para las que se quiere obtener la dirección interpretable por humanos más cercana.
- *bounds* (opcional): son los límites de latitud y de longitud (LatLngBounds) dentro de los que se quiere predeterminar los resultados de codificación geográfica con mayor importancia.
- *region* (opcional): es el código de región, especificado como una subetiqueta region del lenguaje IANA. En la mayoría de los casos, estas etiquetas se asignan directamente a valores de dos caracteres ccTLD ("dominio de nivel superior") ya conocidos
- 

\*Nota: para la búsqueda puedes incluir un campo *address* o un campo *latLng* (si transmites un campo *latLng*, el *geocoder* realizará el proceso conocido como codificación geográfica inversa.

Los parámetros *bounds* y *region* solo influirán en los resultados del geocoder, no los restringirán por completo.

## Respuestas de codificación geográfica

Para el servicio de asignación de identificadores geográficos se necesita un método de devolución de llamada que se ejecute al recuperar los resultados del *geocoder*. Esta devolución de llamada debe transmitir dos parámetros que alojen resultados (*results*) y un código de estado (*status*), en ese orden. Puesto que el *geocoder* puede devolver más de una entrada, el objeto literal *GeocoderResults* es un conjunto.

### Resultados de codificación geográfica

El objeto literal *GeocoderResults* representa un único resultado de codificación geográfica y presenta la siguiente forma:

```
results[]: {
  types[]: string,
  formatted_address: string,
  address_components[]: {
    short_name: string,
    long_name: string,
    types[]: string
  },
  geometry: {
    location: LatLng,
    location_type: GeocoderLocationType,
    viewport: LatLngBounds,
    bounds: LatLngBounds
  }
}
```

A continuación, se explican estos campos.

- *types[]* es un conjunto que indica de qué tipo son los resultados obtenidos. Este conjunto incluye una o más etiquetas que identifican el tipo de elemento que se ha obtenido en el resultado. Por ejemplo, un código geográfico de "Chicago" devuelve "*locality*", lo que indica que "Chicago" es una ciudad y además devuelve "*political*", lo que significa que es una entidad política.
- *formatted\_address* es una cadena que contiene la dirección interpretable por humanos de la ubicación. Esta dirección suele corresponder con la "dirección postal", que a veces difiere de un país a otro (teniendo en cuenta que algunos países, como Gran Bretaña, no permiten la distribución de direcciones postales reales debido a restricciones de la licencia). Esta suele estar formada por uno o varios componentes de la dirección. Por ejemplo, la dirección "111 8th Avenue, New York, NY" contiene componentes independientes para "111 8th Avenue" (una dirección postal), "New York" (la ciudad) y "NY" (el estado de EE.UU.).
- *address\_component[]* es un conjunto que incluye los diferentes componentes de la dirección, tal como se ha explicado anteriormente.

- *geometry* contiene la siguiente información:
  - *location* contiene el valor de latitud y longitud codificado de forma geográfica. Tener en cuenta que este valor se devuelve como un objeto *LatLng*, no como una cadena con formato.
  - *location\_type* almacena datos adicionales sobre la ubicación especificada. Actualmente, se admiten los siguientes valores:
    - *google.maps.GeocoderLocationType.ROOFTOP* indica que el resultado obtenido refleja un código geográfico preciso.
    - *google.maps.GeocoderLocationType.RANGE\_INTERPOLATED* indica que el resultado obtenido refleja una aproximación (normalmente en una carretera) interpolada entre dos puntos precisos (por ejemplo, intersecciones). Los resultados interpolados se suelen obtener cuando los códigos geográficos de la parte superior no están disponibles para una dirección postal.
    - *google.maps.GeocoderLocationType.GEOMETRIC\_CENTER* indica que el resultado obtenido corresponde al centro geométrico de un resultado del tipo polilínea (por ejemplo, una calle) o polígono (una región).
    - *google.maps.GeocoderLocationType.APPROXIMATE* indica que el resultado obtenido es aproximado.
  - *viewport* almacena la ventana gráfica recomendada para el resultado obtenido.
  - *bounds* (se devuelve de forma opcional) almacena los límites de latitud y de longitud (*LatLngBounds*) que pueden contener por completo el resultado obtenido. Tener en cuenta que estos límites pueden no coincidir con la ventana gráfica recomendada (por ejemplo, San Francisco incluye Los Farallones, que técnicamente forman parte de la ciudad pero no se muestran en la ventana gráfica).

El *geocoder* devuelve las direcciones con la configuración de idioma preferida del navegador, o en el idioma especificado al cargar el código JavaScript del API mediante el parámetro *language*.

### Tipos de componentes de la dirección

El conjunto *types[]* del resultado obtenido indica el tipo de dirección. Estos tipos también pueden aparecer dentro de los conjuntos *address\_components[]* para indicar el tipo del componente de la dirección en concreto. Las direcciones del *geocoder* pueden tener diversos tipos (los tipos se pueden considerar "etiquetas"). Por ejemplo, muchas ciudades pueden estar etiquetadas con los tipos *political* y *locality*.

A continuación, se indican los tipos de direcciones que admite y muestra el *geocoder* de HTTP.

- *street\_address* indica una dirección postal precisa.
- *route* indica una carretera identificada (por ejemplo, "US 101").
- *intersection* indica una intersección importante, normalmente de dos carreteras importantes.
- *political* indica una entidad política. Este tipo suele indicar un área poligonal de alguna administración civil.

- *country* indica la entidad política nacional y normalmente, es el tipo de mayor escala que devuelve el geocoder.
- *administrative\_area\_level\_1* indica una entidad política de primer nivel por debajo del nacional. En Estados Unidos, las entidades que corresponden a este nivel administrativo son los estados. No todos los países cuentan con este nivel administrativo.
- *administrative\_area\_level\_2* indica una entidad política de segundo nivel por debajo del nacional. En Estados Unidos, las entidades que corresponden a este nivel administrativo son los condados. No todos los países cuentan con este nivel administrativo.
- *administrative\_area\_level\_3* indica una entidad política de tercer nivel por debajo del nacional. Este tipo indica una división administrativa de menor tamaño. No todos los países cuentan con este nivel administrativo.
- *colloquial\_area* indica un nombre alternativo usado con frecuencia para designar la entidad.
- *locality* indica una entidad política equivalente al municipio.
- *sublocality* indica una entidad política de primer nivel por debajo del municipal.
- *neighborhood* indica un barrio identificado.
- *premise* indica una ubicación identificada. Normalmente se trata de un edificio o de un complejo de edificios con un nombre común.
- *subpremise* indica una entidad de primer nivel por debajo del de ubicación identificada. Normalmente se trata de un edificio individual dentro de un complejo de edificios con un nombre común.
- *postal\_code* indica un código postal utilizado para enviar correo postal dentro del país.
- *natural\_feature* indica un paraje natural destacado.
- *airport* indica un aeropuerto.
- *park* indica un parque identificado.

A continuación, se indican otros posibles tipos de componentes de dirección.

- *post\_box* indica un apartado postal específico.
- *street\_number* indica la calle y el número exacto del edificio.
- *floor* indica la planta del edificio.
- *room* indica la puerta dentro del edificio.

## Códigos de estado

El código status puede devolver uno de los siguientes valores:

- *google.maps.GeocoderStatus.OK* indica que la codificación geográfica se ha realizado correctamente.
- *google.maps.GeocoderStatus.ZERO\_RESULTS* indica que la codificación geográfica se ha realizado correctamente pero no ha devuelto ningún resultado. Esto puede ocurrir si en la codificación geográfica se incluye una dirección (*address*) inexistente o un valor *latng* en una ubicación remota.

- *google.maps.GeocoderStatus.OVER\_QUERY\_LIMIT* indica que se ha excedido el cupo de solicitudes.
- *google.maps.GeocoderStatus.REQUEST\_DENIED* indica que se ha denegado la solicitud por algún motivo.
- *google.maps.GeocoderStatus.INVALID\_REQUEST* normalmente indica que no se ha especificado la solicitud (*address* o *latLng*).

En el siguiente ejemplo se codifica de forma geográfica una dirección y se coloca un marcador en los valores de latitud y de longitud obtenidos. Hay que tener en cuenta que el controlador se transmite como una función literal anónima.

```

var geocoder;
var map;
function initialize() {
  geocoder = new google.maps.Geocoder();
  var latlng = new google.maps.LatLng(-34.397, 150.644);
  var mapOptions = {
    zoom: 8,
    center: latlng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  }
  map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
}

function codeAddress() {
  var address = document.getElementById("address").value;
  geocoder.geocode( { 'address': address}, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
      map.setCenter(results[0].geometry.location);
      var marker = new google.maps.Marker({
        map: map,
        position: results[0].geometry.location
      });
    } else {
      alert("Geocode was not successful for the following reason: " + status);
    }
  });
}

<body onload="initialize()">
  <div id="map_canvas" style="width: 320px; height: 480px;"></div>
  <div>
    <input id="address" type="text" value="Sydney, NSW">
    <input type="button" value="Encode" onclick="codeAddress()">
  </div>
</body>

```

## Codificación geográfica inversa (búsqueda de direcciones)

El término codificación geográfica suele hacer referencia al proceso de traducir una dirección interpretable por humanos en una ubicación de un mapa. El proceso contrario, la traducción de una ubicación a una dirección interpretable por humanos, se conoce como codificación geográfica inversa.

*Geocoder* permite realizar directamente el proceso de codificación geográfica inversa. En lugar de transmitir una dirección (*address*) textual, hay que especificar un par latitud/longitud separado por comas en el parámetro *latLng*.

El siguiente ejemplo codifica de forma geográfica un valor de latitud y de longitud y centra el mapa en esa ubicación, a la vez que muestra una ventana de información en la que se indica la dirección con su formato. Se devuelve el segundo resultado, que es menos específico que el primero (en este caso es el nombre de un barrio):

```
var geocoder;
var map;
var infowindow = new google.maps.InfoWindow();
var marker;
function initialize() {
    geocoder = new google.maps.Geocoder();
    var latlng = new google.maps.LatLng(40.730885, -73.997383);
    var mapOptions = {
        zoom: 8,
        center: latlng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }
    map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
}

function codeLatLng() {
    var input = document.getElementById("latlng").value;
    var latlngStr = input.split(",",2);
    var lat = parseFloat(latlngStr[0]);
    var lng = parseFloat(latlngStr[1]);
    var latlng = new google.maps.LatLng(lat, lng);
    geocoder.geocode({'latLng': latlng}, function(results, status) {
        if (status == google.maps.GeocoderStatus.OK) {
            if (results[1]) {
                map.setZoom(11);
                marker = new google.maps.Marker({
                    position: latlng,
                    map: map
                });
                infowindow.setContent(results[1].formatted_address);
                infowindow.open(map, marker);
            }
        } else {
            alert("Geocoder failed due to: " + status);
        }
    });
}
```

Tener en cuenta que en el ejemplo anterior se muestra el segundo resultado (al especificar *results[1]*). El *geocoder* inverso a menudo devuelve más de un resultado. Las "direcciones" de codificación geográfica no son solo direcciones postales, sino cualquier forma de definir de forma geográfica una ubicación. Por ejemplo, al codificar de forma geográfica un punto de la ciudad de Chicago, el punto codificado se puede etiquetar como una dirección postal, como la ciudad (Chicago), como su estado (Illinois) o como un país (Estados Unidos). Todas ellas son direcciones para el *geocoder*. El *geocoder* inverso devuelve todos esos resultados.

El *geocoder* inverso encuentra coincidencias con entidades políticas (países, provincias, ciudades y barrios), con direcciones y con códigos postales.

A continuación se indica la lista completa de las direcciones que ha devuelto la consulta anterior.

```
results[0].formatted_address: "275-291 Bedford Ave, Brooklyn, NY 11211, USA",
results[1].formatted_address: "Williamsburg, NY, USA",
results[2].formatted_address: "New York 11211, USA",
results[3].formatted_address: "Kings, New York, USA",
results[4].formatted_address: "Brooklyn, New York, USA",
results[5].formatted_address: "New York, New York, USA",
results[6].formatted_address: "New York, USA",
results[7].formatted_address: "United States"
```

Las direcciones se devuelven ordenadas de mayor a menor coincidencia. Normalmente, la dirección más exacta es el resultado más destacado, como en este caso. Ten en cuenta que se devuelven diferentes tipos de direcciones, desde la dirección postal más específica hasta entidades políticas menos específicas como barrios, ciudades, países, estados, etc. Si quieres obtener resultados de direcciones más generales, puedes examinar el campo *results[].types*.

Nota: la codificación geográfica inversa no es una ciencia exacta. El *geocoder* intentará obtener la ubicación localizable mediante la dirección más cercana dentro de un determinado margen.