



UNIVERSIDAD DE VALPARAÍSO  
Facultad de Ingeniería  
Escuela de Ingeniería Civil en Informática  
Ingeniería Civil Informática

# DESARROLLO DE UNA APLICACIÓN PARA REALIZAR CONSULTAS EN ÁLGEBRA RELACIONAL

Propuesta de Trabajo de Título  
Sebastián Andrés Segovia Cordero  
*sebastian.segoviac@alumnos.uv.cl*  
3 de abril de 2014

Profesor Guía: Eliana Paz Providel Godoy

## Resumen

La Escuela de Informática de la Universidad de Valparaíso cuenta con un ramo llamado Modelo de Datos. Ahí se imparten los conocimientos básicos para entender lo que es una Base de Datos y sus clasificaciones, centrándose en Bases de Datos Relacionales. Desde Modelos Relacionales hasta SQL, cada concepto relacionado es impartido y ejercitado sin mayor dificultad, pero hay uno que muestra una falencia a la hora de ser impartido, básicamente por ser algo totalmente teórico y su dificultad para medir el grado de avance en los alumnos, el Álgebra Relacional. En este Trabajo de Título se presenta una solución factible a este problema, además de un enfoque que podría revolucionar la metodología de la enseñanza del Álgebra Relacional en este ramo.

# Índice

<b>1. Introducción</b>	<b>4</b>
<b>2. Definición del Problema</b>	<b>6</b>
2.1. Problema . . . . .	6
2.2. Solución Propuesta . . . . .	7
2.3. Importancia del trabajo . . . . .	7
<b>3. Objetivos</b>	<b>8</b>
3.1. Objetivo General . . . . .	8
3.2. Objetivos Específicos . . . . .	8
<b>4. Metodología</b>	<b>9</b>
<b>5. Planificación</b>	<b>11</b>
<b>6. Recursos</b>	<b>13</b>
6.1. Recursos Humanos . . . . .	13
6.2. Recursos Materiales . . . . .	13
6.3. Recursos del Desarrollador . . . . .	13
<b>Bibliografía</b>	<b>14</b>

## Índice de tablas

1.	Explicación de metodología de desarrollo. . . . .	10
2.	Planificación desde Marco conceptual hasta Implementación. . . . .	11
3.	Planificación desde Validación del sistema hasta Análisis de resultados.	12

## Índice de figuras

1.	Metodología de desarrollo. . . . .	9
----	------------------------------------	---

# 1. Introducción

En una era donde guardar y recuperar grandes colecciones de datos se ha convertido en una necesidad, ya sea, por ejemplo para, mantener un historial de las transacciones realizadas en un banco en un determinado tiempo o para recopilar y procesar datos obtenidos durante alguna simulación, las Bases de Datos nos han otorgado una eficiente forma para ordenar, mantener y recuperar todo tipo de información.

Para dar solución de forma efectiva al almacenamiento de datos han surgido muchos tipos de Bases de Datos, tales como: jerárquicas, orientada a objetos, deductivas, espaciales, entre otros; todas ellas con diferentes objetivos y paradigmas. Pero hay una en especial que se ha instalado fuertemente en el universo de las aplicaciones de Base de Datos, de fácil acceso y con fuertes fundamentos teóricos: las Bases de Datos Relacionales [1].

Las Bases de Datos relacionales se presentan como una colección de datos y las relaciones existentes entre ellas, donde cada relación es visualizada como una *tabla* de valores denominada *relación*. En ella, cada *fila* o *tupla* representa un grupo de datos relacionados entre sí y cada *columna* se explica como un valor de algún dominio específico que pertenece a una *tupla*, comúnmente llamado *atributo*. En una Base de Datos Relacional, el nombre de las *tablas* y los nombres de los *atributos* ayudan a entender el significado de los valores que contiene en cada *tupla*, además, cada *tupla* contiene un atributo único llamado *clave primaria* que nos ayuda a identificarlas y diferenciarlas entre ellas. Una Base de Datos Relacional comúnmente contiene muchas *relaciones* y en éstas las *tuplas* están conectadas de diferentes maneras manteniendo a su vez una integridad en los datos. Esta integridad es obtenida mediante distintos tipos de restricciones que especifican cómo debe comportarse un esquema relacional y cada una de sus instancias. Estas son: *Integridad de Entidades*, que nos indica que ninguna clave primaria debe ser nula para poder así diferenciar una tupla de otra; e *integridad referencial*, que se centra en las referencias entre relaciones gracias a un atributo llamado *Clave Externa*. Éste último debe cumplir con 2 reglas: primero debe pertenecer al mismo dominio que los atributos de la tabla que se quiere referenciar y su valor debe ocurrir en alguna tupla de la tabla referenciada o bien ser nulo [1].

Dentro de las Bases de Datos Relacionales, existen variadas operaciones con el fin de actualizar y obtener los datos en ella. Cada una de estas operaciones debe seguir las *Restricciones de Integridad* definidas anteriormente. Dentro de las actualizaciones, hay 3 operaciones que son básicas: *insertar*, que sirve para agregar una o más *tuplas* nuevas en una relación; *eliminar*, que permite eliminar una o más *tuplas*; y *modificar* que sirve para modificar los valores de algunos *atributos*.

Para la obtención, organización y gestión de datos, existe un lenguaje que permite

la comunicación con el Sistema Gestor de Base de Datos<sup>1</sup> (SGBD) llamado Lenguaje Estructurado de Consultas (SQL). Este es un lenguaje tan popular en Bases de Datos Relacionales que muchos lenguajes de programación adicionan sentencias SQL como parte de su colección. Entre sus principales características se destaca que es un lenguaje para cualquier tipo de usuarios, desde administradores, desarrolladores e incluso usuarios normales; es un lenguaje declarativo, es decir, solamente se especifica qué se quiere, no donde ni cómo; y es utilizado tanto como para realizar consultas, como para actualizaciones, definiciones de datos y controles de acceso de usuarios y de transacciones [3].

De esta forma SQL proporciona un gran repertorio de sentencias que se usan en distintas tareas, tales como consultar datos, crear, actualizar y eliminar *tablas* en una base de datos, crear, actualizar y eliminar datos de las *tablas*, controlar el acceso a la base de datos y a *las tablas* [3]. Pero existe una mejor forma de entender las consultas en una Base de Datos Relacional previo al SQL, el Álgebra Relacional.

El Álgebra Relacional es un lenguaje que consta de una colección de operaciones que se utilizan para manipular *tablas* y especificar consultas. Estas operaciones, por ejemplo, nos permite obtener una *tupla* específica o el valor de un *atributo* de un conjunto de tuplas. El resultado de estas consultas es una nueva relación, permitiéndonos de esta forma seguir aplicando operaciones de Álgebra Relacional para depurar más una consulta. Estas operaciones se clasifican en 2 grupos: uno pertenece a las operaciones básicas de la Teoría Matemática de Conjuntos (Unión, Intersección, Diferencia, Producto Cartesiano, etc.) y el otro consta de operaciones específicas para Bases de Datos Relacionales (Seleccionar, Proyectar, Reunión, etc.) [1].

De esta forma, el Álgebra Relacional se transforma en un lenguaje potente a la hora de realizar consultas, ya que provee de simples pero poderosas operaciones para manipular de manera efectiva los datos en una Base de Datos Relacional. Su importancia es clave para entender mejor el tratamiento de las Bases de Datos Relacionales y cómo realizar consultas en él, pero con una gran falencia, es sólo teoría. A consecuencia de esta falla surgen una serie de dudas. ¿Es posible entregar una base consistente para que los alumnos de la escuela de ingeniería civil en informática que deben aprender este lenguaje puedan entender claramente de que trata y cómo funciona el Álgebra Relacional? ¿Existen herramientas para medir el grado de conocimiento que adquieren los alumnos a la hora de enseñarles este lenguaje?

Este Trabajo de Título tiene como objetivo desarrollar una herramienta para Álgebra Relacional tal que pueda aportar a los alumnos en su aprendizaje y detectar los errores que ocurren con mayor frecuencia, para así encaminar de manera personalizada la enseñanza de éste lenguaje.

---

<sup>1</sup>Sistema Gestor de Base de Datos: "Colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos" [3]

EL documento está estructurado de la siguiente manera. La sección 2 define en detalle el problema a tratar, junto con la solución propuesta y la importancia de este trabajo. La sección 3 se especifica el objetivo general y los objetivos específicos que se desean alcanzar. La sección 4 explica la metodología a utilizar. La sección 5 muestra la planificación de la ejecución de este trabajo. Y finalmente la sección 6 especifica los recursos necesarios para realizar este trabajo.

## 2. Definición del Problema

En esta sección se presenta el problema que se tratará en el presente Trabajo de Título, la solución propuesta para resolver el problema y para finalizar con la importancia de este trabajo.

### 2.1. Problema

La Escuela de Informática de la Universidad de Valparaíso dentro de su malla curricular contiene un curso que se especializa en el conocimiento de las Bases de Datos Relacionales, llamado INC402 - Modelo de Datos. En este curso se enseña los distintos enfoques de modelos de bases de datos, centrándose principalmente en el modelo relacional, cómo generar modelos relacionales y comprender el tratamiento de consultas mediante el Álgebra Relacional. Es un curso totalmente teórico y es la base para entender los Sistemas Gestores de Bases de Datos que utilizan Base de Datos Relacionales. Si bien el curso presenta un temario suficiente para comenzar a trabajar directamente con Bases de Datos Relacionales, presenta falencias importantísimas a la hora de enseñar y medir el conocimiento de los alumnos, sobretodo en Álgebra Relacional.

Tal como el pseudo-código nos ayuda a entender como trabajar con distintos lenguajes de programación, el Álgebra Relacional es la base de todo Lenguaje Estructurado de Consultas (SQL) y su comprensión es de suma importancia al momento de realizar consultas en cualquier Sistema Gestor de Base de Datos.

La metodología utilizada para enseñar el lenguaje se centra en clases presenciales donde se explica qué es el Álgebra Relacional, lectura complementaria que explica cómo trabajar con él, ejemplos hechos desde el pizarrón y guías en papel con ejercicios para practicarlo. Esto acarrea una serie de problemas tanto para los alumnos como para el profesor. Uno de ellos es que cuando se realizan ejercicios en papel los alumnos no tienen forma de saber si sus consultas funcionan correctamente, es decir, no saben si su resultado es el deseado o si la sintaxis utilizada es la correcta, obligándolos a esperar hasta encontrarse con algún Sistema Gestor de Base de Datos para recién probar sus consultas. El profesor por su parte puede formular muchos problemas, guiar el trabajo en clases y entregar una manera correcta de resolverlo, pero los

ejercicios realizados por los alumnos fuera de la clase no pueden ser revisados en su totalidad. Herramientas automáticas para trabajar con Álgebra Relacional ya están implementadas, pero ninguna se adecua a las necesidades de la asignatura, por ende no son utilizados. Junto con esto, y dado la situación anterior, no existe forma de saber cómo y en qué se equivocan los alumnos a la hora de enfrentarse a un problema donde se utilice el Álgebra Relacional fuera del aula. Además, no existe un registro de los errores más cometidos ni la frecuencia de los mismos.

## **2.2. Solución Propuesta**

Es por esto que este Trabajo de Título tiene como objetivo crear una aplicación que permita trabajar directamente con consultas en Álgebra Relacional, de tal manera que los alumnos de Modelo de Datos de la Escuela de Informática puedan aprender y ejercitar de forma más didáctica sus consultas en Bases de Datos Relacionales. A su vez, y para darle un valor agregado a esta aplicación, se implementará un módulo de extracción de datos con el fin de recopilar información de entrada ingresada por los alumnos que utilicen la aplicación para así obtener un feedback más representativo y directo.

## **2.3. Importancia del trabajo**

Gracias a esto, el alumno podrá detectar de manera inmediata y automática si sus consultas tienen una sintaxis adecuada o si estas entregan los resultados esperados. De esta forma, el alumno se enfocará directamente en trabajar con Álgebra Relacional y no tendrá la necesidad de enfrentarse directamente con un Sistema Gestor de Base de Datos, evitando la tarea de aprender la sintaxis de algún lenguaje SQL.

Por otra parte, y gracias al sistema de extracción de datos, se podrá hacer un seguimiento del trabajo de cada alumno al momento de realizar sus consultas. De esta forma el profesor tendrá la posibilidad de realizar un análisis sobre los datos obtenidos con el fin de saber, por ejemplo, cuáles son los errores más comunes, la frecuencia de ellos o la cantidad de alumnos que no comenten ningún error. Esto resulta de mucha ayuda para el profesor, ya que el feedback obtenido gracias al estudio de estos datos puede ser utilizado para reforzar las áreas donde los alumnos presenten más problemas, personalizando aún más el aprendizaje de los alumnos.

### 3. Objetivos

En esta sección se da a conocer el Objetivo General del Trabajo de Título, junto con los Objetivos Específicos que se desprenden de él.

#### 3.1. Objetivo General

El objetivo principal de este Trabajo de Título es desarrollar una aplicación que ejecute consultas de Álgebra Relacional con el fin de entregar una herramienta didáctica para fomentar el aprendizaje de este lenguaje, además de construir un módulo de extracción de datos para el posterior estudio del comportamiento de los alumnos mientras usan la aplicación con la finalidad ser un apoyo en la enseñanza del Álgebra Relacional.

#### 3.2. Objetivos Específicos

- Realizar una toma de requerimientos con el fin de establecer el dominio de trabajo, delimitar la frontera y analizar las necesidades fundamentales que la aplicación debe satisfacer.
- Diseñar, codificar y testear la aplicación para consultas de Álgebra Relacional.
- Desarrollar una Base de Datos que recopile los datos de las consultas realizadas y conductas de los sujetos de prueba en tiempo de ejecución.
- Crear una Base de Datos de ejemplo para importar a la aplicación a desarrollar.
- Desarrollar un set de ejercicios de ejemplos.
- Diseñar, codificar y testear un módulo para la captura de datos de las consultas realizadas de los sujetos de prueba en tiempo de ejecución, es decir, errores cometidos, sintaxis, entre otros.
- Utilizando técnicas de Minería de Datos, analizar e interpretar los datos obtenidos de las pruebas desde la Base de Datos de recolección de información.



## 4. Metodología

Para el desarrollo de la aplicación se usará un Modelo Incremental, ya que este modelo *"entrega una serie de lanzamientos, llamados incrementos, que proporcionan en forma progresiva más funcionalidad para los clientes a medida que se entrega cada uno de los incrementos"* [2]. De esta forma, es posible sacar provecho de la aplicación sin la necesidad de esperar hasta que esté completo, entregando en primeras instancias un prototipo que satisfaga los requerimientos más críticos [4].

Para el resto del Trabajo de Título, es decir, para las etapas previas y posteriores al desarrollo, se utilizará un Modelo en Cascada. La representación gráfica de la metodología en la realización del presente Trabajo de Título se presenta en la figura 1, y luego en la Tabla 1 se explica cada una de las etapas.

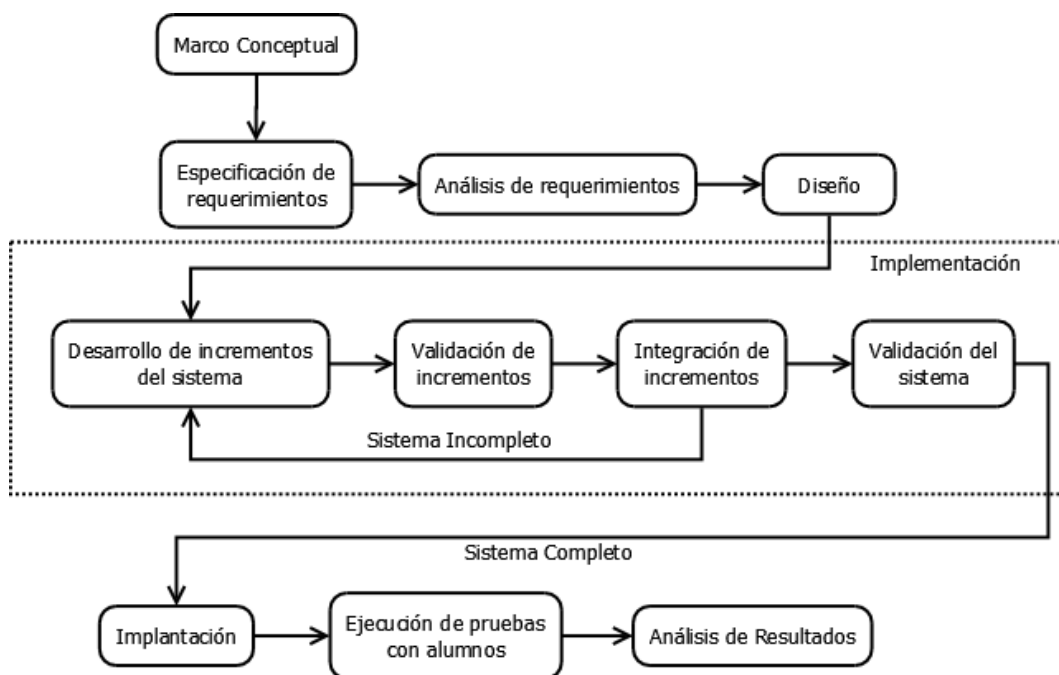


Figura 1: Metodología de desarrollo.

Etapa	Descripción	Productos
Marco conceptual.	Se realizará una búsqueda en la WEB de aplicaciones que trabajen con consultas en Álgebra Relacional, además de publicaciones y libros relacionados para complementar el trabajo.	Documento estado del arte.
Especificación de requerimientos.	Se recopilan y especifican los requerimientos necesarios para la creación de la aplicación.	Requerimientos funcionales y no funcionales.
Análisis de requerimientos.	Se analizan los requerimientos funcionales y no funcionales para la generación de diagramas y modelos.	Actores del sistema, diagrama de casos de usos, modelo conceptual, modelo entidad/relación.
Diseño.	Diseño de la arquitectura del software, diseño lógico de la aplicación, diseño de pruebas, diseño de datos, diseño de interfaz de usuario.	Diagrama de arquitectura del sistema, diagrama de interacción, diagrama de clases, modelo interno lógico, esquemas de navegación, diseño de plan de testing, diseño de pruebas con alumnos.
Desarrollo de incrementos del sistema.	A través de 3 iteraciones, se implementa la aplicación.	Módulo de consultas, módulo de gestión de tablas, módulo de obtención de datos.
Validación de incrementos.	Se aplica testing unitario a cada incremento implementado.	Resultados de testing, códigos corregidos.
Integración de incrementos.	Se integra el incremento testeado y aprobado a la aplicación.	Sistema anterior más nuevo incremento.
Validación del sistema.	Por medio de testing de aceptación, se prueba y acepta el sistema completo.	Resultados de testing, códigos corregidos.
Implantación.	La aplicación es instalada y se encuentra lista para ser utilizada.	Instalación de la aplicación funcional.
Ejecución de pruebas con alumnos.	Se seleccionan un grupo de alumnos y se realizan pruebas con ellos para la recopilación de datos.	Datos de pruebas.
Análisis de resultados.	Mediante minería de datos, se analizan los datos para observar patrones en el comportamiento de los alumnos.	Informe de resultados de pruebas.

Tabla 1: Explicación de metodología de desarrollo.

## 5. Planificación

La Tabla 2 muestra la planificación detalla del desarrollo desde el marco conceptual hasta la implementación de la aplicación y la Tabla 3 muestra la planificación desde la etapa de validación del sistema hasta el análisis de resultados. Es importante recalcar que semanalmente habrá una reunión de 1 hora con el profesor guía para aclarar dudas y mostrar avances.

<b>Id</b>	<b>Etapas</b>	<b>Días</b>	<b>Fecha inicio</b>	<b>Fecha fin</b>	<b>Actividad antecesora</b>
1.0.0	Marco conceptual	5	07/04/2014	11/04/2014	-
1.1.0	Búsqueda y lectura de material de apoyo	3	07/04/2014	09/04/2014	-
1.2.0	Documentación	2	10/04/2014	11/04/2014	1.1.0
2.0.0	Especificación de requerimientos	5	14/04/2014	18/04/2014	1.0.0
2.1.0	Especificación de requerimientos funcionales	2	14/04/2014	15/04/2014	1.0.0
2.2.0	Especificación de requerimientos no funcionales	2	16/04/2014	17/04/2014	1.0.0
2.3.0	Documentación	1	18/04/2014	18/04/2014	2.2.0
3.0.0	Análisis de requerimientos	5	21/04/2014	25/04/2014	2.0.0
3.1.0	Actores del sistema y diagrama de casos de usos	1	21/04/2014	21/04/2014	2.0.0
3.2.0	Modelo conceptual	1	22/04/2014	22/04/2014	3.1.0
3.3.0	Modelo entidad relacion	2	23/04/2014	24/04/2014	3.2.0
3.4.0	Documentación	1	25/04/2014	25/04/2014	3.3.0
4.0.0	Diseño	15	28/04/2014	16/05/2014	3.0.0
4.1.0	Diagrama de arquitectura	2	28/04/2014	29/04/2014	3.0.0
4.2.0	Esquemas de navegación	2	30/04/2014	01/05/2014	4.1.0
4.3.0	Diagrama de interacción	2	02/05/2014	05/05/2014	4.2.0
4.4.0	Diagrama de clases	2	06/05/2014	07/05/2014	4.3.0
4.5.0	Modelo interno lógico	2	08/05/2014	09/05/2014	4.4.0
4.6.0	Diseño plan de Ttesting	2	12/05/2014	13/05/2014	4.5.0
4.7.0	Diseño prueba con alumnos	2	14/05/2014	15/05/2014	4.6.0
4.8.0	Documentación	1	16/05/2014	16/05/2014	4.7.0
5.0.0	Desarrollo y Validación de incrementos del sistema	26	19/05/2014	23/06/2014	4.0.0
5.1.0	Iteración 1	8	19/05/2014	28/05/2014	4.0.0
5.1.1	Implementación módulo de consultas	5	19/05/2014	23/05/2014	4.0.0
5.1.2	Testing unitario	2	26/05/2014	27/05/2014	5.1.1
5.1.3	Documentación	1	28/05/2014	28/05/2014	5.1.2
5.2.0	Iteración 2	9	29/05/2014	10/06/2014	5.1.0
5.2.1	Implementación módulo de gestión de tablas	6	29/05/2014	05/06/2014	5.1.0
5.2.2	Testing unitario	2	06/06/2014	09/06/2014	5.2.1
5.2.3	Documentación	1	10/06/2014	10/06/2014	5.2.2
5.3.0	Iteración 3	9	11/06/2014	23/06/2014	5.2.0
5.3.1	Implementación módulo de obtención de datos	6	11/06/2014	18/06/2014	5.2.0
5.3.2	Testing unitario	2	19/06/2014	20/06/2014	5.3.1
5.3.3	Documentación	1	23/06/2014	23/06/2014	5.3.2

Tabla 2: Planificación desde Marco conceptual hasta Implementación.

<b>Id</b>	<b>Etapas</b>	<b>Días</b>	<b>Fecha inicio</b>	<b>Fecha fin</b>	<b>Actividad antecesora</b>
6.0.0	Validación del sistema	4	24/06/2014	27/06/2014	5.0.0
6.1.0	Testing de integración	3	24/06/2014	26/06/2014	5.0.0
6.2.0	Documentación	1	27/06/2014	27/07/2014	6.1.0
7.0.0	Implatación	5	30/06/2014	04/07/2014	6.0.0
7.1.0	Testing de aceptación	3	30/06/2014	02/07/2014	6.0.0
7.2.0	Instalación de la aplicación	1	03/07/2014	03/07/2014	7.1.0
7.3.0	Documentación	1	04/07/2014	04/07/2014	7.2.0
8.0.0	Ejecución de pruebas con alumnos	25	28/07/2014	29/08/2014	7.0.0
8.1.0	Diseño set de ejercicios para pruebas con alumnos	5	28/07/2014	01/08/2014	7.0.0
8.2.0	Pruebas con alumnos	15	04/08/2014	22/08/2014	8.1.0
8.3.0	Documentación	5	25/08/2014	29/08/2014	8.2.0
9.0.0	Análisis de resultados	45	01/09/2014	31/10/2014	8.0.0
9.1.0	Selección de técnica de análisis para minería de datos	10	01/09/2014	12/09/2014	8.0.0
9.2.0	Aplicación de técnica de análisis de minería de datos	10	15/09/2014	26/09/2014	9.1.0
9.3.0	Análisis de datos	20	29/09/2014	24/10/2014	9.2.0
9.4.0	Documentación	5	27/09/2014	31/10/2014	9.3.0

Tabla 3: Planificación desde Validación del sistema hasta Análisis de resultados.

## **6. Recursos**

En esta sección se especifican los recursos mínimos necesarios para llevar a cabo el presente Trabajo de Título.

### **6.1. Recursos Humanos**

- Profesor Guía.
- Alumnos que estudien en la Escuela de Informática de la Universidad de Valparaíso para implementar las pruebas.

### **6.2. Recursos Materiales**

- Computador, Procesador: Intel(R) Core(TM) Duo, 2.00 GHz, Memoria Ram: 3 GB y Disco Duro: 320 GB.
- Laboratorio con la aplicación instalada para implementar pruebas.

### **6.3. Recursos del Desarrollador**

- Procesador de Textos ( $\text{\LaTeX}$  ).
- Libros y Publicaciones acordes al presente Trabajo de Título.

## Bibliografía

- [1] Ramez Elmasri, Shamkant B Navathe, Verónica Canivell Castillo, Beatriz Galán Espiga, and Gloria Zaballa Pérez. *Fundamentos de sistemas de bases de datos*. Addison-Wesley, 2002.
- [2] Roger S PRESSMAN. *Ingeniería de Software, Sexta Edición, Ed.* Mc Graw Hill, México, 2005.
- [3] Montero Rodríguez Fernando Ramos Martín María Jesús, Ramos Martín Alicia. *Sistemas Gestores de Base de Datos*. Mc Graw Hill, México, 2006.
- [4] Ian Sommerville. *Ingeniería del software*. Pearson Educación, 2005.