



SISTEMA PARA EL ESTUDIO  
TÉCNICO EN LA  
INGENIERÍA DE  
PROYECTOS

por

Hugo Arzola Torijano

Para Obtener el Título de:

**Licenciado en Ciencias de la  
Computación**

Benemérita Universidad Autónoma  
de Puebla

Puebla, Pue. Febrero 2014

Asesor:

M.C. Meliza Contreras González

---

## ÍNDICE

<b>Agradecimientos</b> .....	iv
<b>Prefacio</b> .....	v
<b>Introducción</b> .....	vii
<b>Capítulo I: Estado del Arte</b> .....	1
1.1 Estado del Arte .....	1
1.1.1 Microsoft Dynamics ERP .....	1
1.1.2 SAP Bussines All-in-one.....	2
1.1.3 kpi.com Simple ERP .....	3
1.2 Justificación.....	4
<b>Capítulo II: Marco Teórico</b> .....	5
2.1 Paradigma Orientado a Objetos .....	5
2.1.1 Antecedentes .....	6
2.2 Lenguaje Unificado de Modelado(UML) .....	7
2.3 Diseño de la Base de Datos .....	8
2.3.1 Modelo Entidad Relación .....	8
2.3.1.1 Entidades.....	9
2.3.1.2 Atributos.....	9
2.3.1.3 Relaciones.....	10
2.3.2 Normalización .....	12
2.3.2.1 Dependencia Funcional.....	12
2.3.2.2 Formas Normales .....	13
2.4 Plataformas de Implementación.....	14
2.4.1 XAMPP .....	14
2.4.2 Mac OS X (Snow Leopard) .....	15
2.4.3 XCode.....	15
2.4.4 ADT Android .....	16
2.3.5 Phoneygap.....	16
2.3.6 Dreamweaver .....	18

2.4.7 MySQL .....	18
2.4.8 PHP .....	19
2.4.9 HTML 5 .....	19
2.4.10 JavaScript .....	20
2.4.11 CSS3 .....	21
2.4.12 JQueryMobile .....	21
<b>Capítulo III: Análisis y Diseño del Sistema .....</b>	<b>23</b>
3.1 Requisitos del Sistema.....	23
3.1.1 Diagrama general de casos de uso .....	24
3.1.2 Especificación de casos de uso .....	25
3.1.3 Escenarios .....	29
3.2 Modelo Conceptual.....	35
3.2.1 Diagrama de clases .....	35
3.2.2 Diccionario de modelo .....	36
3.3 Diseño del Sistema .....	37
3.3.1 Modelo Conceptual.....	37
3.3.2 Colección de clases de análisis.....	38
3.3.3 Diagrama de clases de análisis .....	39
3.3.4 Diagramas de secuencia.....	41
3.3.5 Diagramas de colaboración.....	43
<b>Capítulo IV: Diseño de la Base de Datos.....</b>	<b>46</b>
4.1 Modelo Entidad Relación .....	46
4.1.1 Entidades.....	46
4.1.2 Atributos.....	47
4.1.3 Identificadores Únicos.....	48
4.2 Diagrama Entidad Relación.....	51
4.2.1 Relaciones.....	52
4.3 Normalización de la Base de Datos .....	54
4.3.1 FN1 .....	55
4.3.2 FN2 .....	55
4.3.3 FN3 .....	56

4.4 Implementación de la Base de Datos .....	56
<b>Capítulo V: Implementación</b> .....	<b>58</b>
5.1 Interfaz del Sistema .....	58
5.1.1 Pantalla de Inicio.....	58
5.2.1 Página de Usuario .....	59
5.2 Pruebas del Sistema.....	62
5.1.1 Caja Blanca.....	63
5.2.1 Caja Negra.....	67
Conclusiones .....	69
Perspectivas .....	70
Bibliografía.....	71

## AGRADECIMIENTOS

**A mi Madre** Por brindarme tu amor incondicional y enseñarme a ser un hombre de bien, por ser ejemplo de esfuerzo constante y valor para sacar adelante a mis hermanos y a mí, por darme la oportunidad y la guía necesaria para culminar esta etapa, gracias Mamá.

**A mi Padre** Por tu apoyo incondicional, por ser un gran ejemplo de trabajo, responsabilidad e inquebrantable voluntad, por mostrarme el camino y las incontables enseñanzas han hecho posible que llegue a esta etapa. Tu junto con mi mamá son los pilares de esta familia. Gracias Papa.

**A mis Hermanos** Por compartir victorias, derrotas, risas, lágrimas, juegos y obligaciones. Gracias por compartir su tiempo conmigo, les deseo mucho éxito a lo largo de su camino, saben que siempre me tendrán a su lado para levantarlos o para reprender sus faltas cuando sea necesario. Gracias Hermanos.

**A mis Tías y Tíos** Por cuidarme y quererme como a uno de sus hijos, por brindarme su apoyo y su mano cuando lo necesite, por todo el ejemplo enseñado a lo largo de este camino. Gracias Tías y Tíos.

**A mis Primos** Por los buenos momentos que vivimos, por ser cómplices de secretos e incontables travesuras, por compartir risas y regaños, por compartir su tiempo conmigo. Gracias Primos.

**A mis Abuelos** Por los consejos y experiencias contadas, por brindarme a mis padres, por quererme y apoyarme. Gracias Abuelos.

## PREFACIO

El presente trabajo se encuentra enfocado a desarrollar un sistema que pueda ser utilizado en un dispositivo móvil, con el cual pueda ingresar datos para así sacar los resultados necesarios en un estudio de técnico.

Para desarrollar el sistema antes mencionado se analizaron las funciones que son más susceptibles de ser automatizadas, así como las que demandan una mayor atención dentro de los procesos de un estudio técnico.

Sustentado en una metodología para el desarrollo de software como lo es UML (Modelado de Lenguaje Unificado), se trazó un diseño que cumpliera con los requisitos y funcionalidades esperadas, que se obtuvieron a partir de casos de uso..

Así mismo, mediante la aplicación de la teoría de Bases de Datos Relaciones se elaboraron los modelos Entidad-Relación y Relacional que nos ayudan a implementar la Base de Datos, la cual permite ejecutar las funciones de integridad, persistencia y transacciones del sistema propuesto.

Por otra parte, se plantea que la naturaleza de la herramienta propuesta, sea utilizando las tecnologías de desarrollo de software más actuales, además de cuidar el aspecto de inversión económica en su elaboración, implantación y manejo, por tanto se eligió la plataforma XAMP (acrónimo de Mac OS X, Apache, MySQL, PHP) para su fabricación; Mac OS X Snow Leopard como Sistema Operativo donde se trabajará y codificara la aplicación e IOS 5 como sistema operativo para móviles en donde se desea implementar; Apache como Servidor de páginas WEB; MySQL como Administrador de la base de Datos del Sistema, PHP para la implantación de las reglas de negocio, JavaScript para el manejo de información entre cliente y el servidor, finalmente XCode se necesito para poder compilar la aplicación en IOS.

Adicionalmente se empleó HTML5 (Lenguaje de Marcado de Hipertexto), CSS3 (Hojas de Estilo en Cascada) y JQueryMobile (Framework de JQuery para móviles), para complementar el desarrollo de la interfaz y funcionalidad del usuario.

## INTRODUCCIÓN

### **Planteamiento del Problema**

Es frecuente encontrar empresas que al iniciar su actividad económica, se dan cuenta que el lugar geográfico donde se encuentran o las materias primas que utilizan no son las más óptimas, por lo que realizar un estudio preventivo resulta de vital importancia, ya que permite detectar y establecer las condiciones más adecuadas durante la construcción, realización ó finalización de un producto ó servicio, y también permite conocer a más detalle las futuras inversiones que tendrán que realizarse.

### **Objetivos**

#### **General.-**

Por lo tanto se plantea un sistema que permita administrar\_ la información obtenida en un estudio técnico de factibilidad, que particularmente obtenga el tamaño óptimo y la localización óptima de la empresa, además que pueda administrar la información sobre los equipos usados, las instalaciones y los aspectos organizativos en la realización de la producción de bienes o en la prestación de los servicios de la misma.

#### **Específicos.-**

El sistema debe realizar las siguientes acciones:

1. Administrar la información sobre el tipo de producción y el tamaño de la planta.
2. Administrar la información sobre las actividades de los obreros y las maquinas.



3. Administrar la información sobre los procesos de producción y sobre la localización óptima del proyecto.
4. Generar plantillas para agregar estudios técnicos de distintos proyectos.
5. Generar tablas de comparación de localización de plantas y actividades de obreros y máquinas.

Con esto se pretende crear un sistema que permita recopilar datos relevantes sobre el desarrollo de un proyecto para tomar la mejor decisión, si procede su desarrollo o implementación.

## ESTADO DEL ARTE

### 1.1 Estado del Arte

En la actualidad los sistemas que más se acercan a las demandas de un estudio técnico son los sistemas ERP, a continuación describiremos algunos de los que podemos encontrar en el mercado.

#### 1.1.1 Microsoft Dynamics

Es un software de planificación de recursos a nivel empresarial de Microsoft. Este tipo de sistemas ERP, son sistemas que conjuntan varios módulos para dar soluciones a los diferentes problemas y demandas que surgen en una empresa chica, mediana ó grande[3].

Microsoft permite adaptar el software a nuestras necesidades como empresa, para ello divide su software en dos tipos de licencias, la primera que está dirigida hacia grandes empresas y se llama Microsoft Dynamics AX, el cual está diseñado para adaptarse específicamente a cinco sectores: sector público, servicios industriales, manufactura y comercio minorista y distribución.

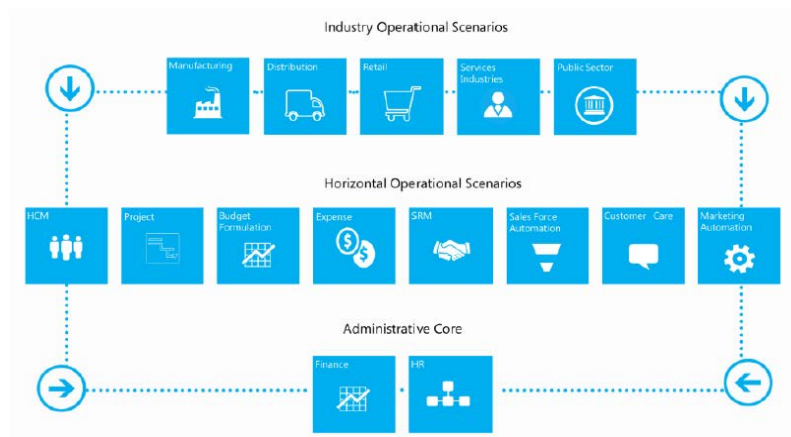


Fig 1.1 Diagrama de escenarios Microsoft Dynamics

La figura anterior muestra como están compuestos los diferentes escenarios que plantea Microsoft Dynamics AX en el quehacer diario de la industria, para ello plantea soluciones como gestión financiera, inteligencia de negocios, ventas y marketing, compras y abastecimiento, gestión de la cadena de suministros, gestión de proyectos y funcionalidad, gestión de capital humano, riesgo y cumplimiento globales, sostenimiento medio ambiental y funcionalidades específicas del país.

La segunda licencia se llama Microsoft Dynamics GP que está diseñado y enfocado para PyMEs dentro de sus ventajas se encuentra acceso a la información crítica de las empresas, integración con las herramientas de Microsoft tradicionales, es altamente personalizable y cuenta con múltiples herramientas de integración de sistemas actuales y futuros.

### **1.1.2 SAP Bussines All-in-one**

SAP es un software modular para la gestión empresarial, el cual ofrece una solución empresarial completa para la mediana empresa que garantiza el nivel de funcionalidad necesario para su negocio. Especialmente adaptadas a los procesos empresariales de la mediana empresa, las soluciones de SAP Business All-in-One permiten aprovechar las mejores prácticas del sector, proporcionando al mismo tiempo flexibilidad para adaptarse a los nuevos requisitos empresariales[1].

Dentro de los beneficios que ofrece esta la optimización de la eficacia operativa, mejorar la visibilidad de la empresa, reducir el costeo de propiedad, responder a los requisitos empresariales en constante evolución y mejorar la productividad.

### 1.1.3 kpi.com Simple ERP

Kpi es un software que completa e integra la infraestructura de apoyo empresarial con Customer Relationship Management (CRM), Gestión de proyectos (Project Management), Recursos humanos (HR), Nómina (Payroll), comercio electrónico (E-Commerce), Contabilidad básica (Basic Accounting) y reportes. Todos estos elementos de negocio se encuentran conectados a través de la nube[2].

Por lo que kpi.com Simple ERP es la versión móvil del software principal, dentro de las ventajas se encuentran que toda la información almacenada se encuentra en la nube por lo que puedes acceder a ella cuando la necesites, plugins gratuitos compatibles con Microsoft Office, engloba diferentes necesidades empresariales y como se menciona anteriormente esta su versión para móviles.



Fig 1.2 Pantalla de kpi.com Simple ERP

La figura anterior muestra la pantalla de inicio y los módulos a los que puedes acceder en la versión móvil de kpi.com.

## **1.2 Justificación**

Con los ejemplos anteriormente analizados podemos apreciar que los sistemas tipo ERP son sistemas que contemplan varios módulos para las diferentes demandas de la industria ofreciendo soluciones integrales y personalizables para cada cliente, estas soluciones van desde la administración y control del personal y su nómina hasta la optimización de la producción que se realiza en cada industria.

Si bien los sistemas mencionados anteriormente ofrecen múltiples soluciones para PyMEs y grandes empresas, aun no abarcan el estudio que antecede a la instalación y realización de un producto o servicio, sin mencionar que sus productos a pesar de que si tienen contemplados algunos módulos en su versión móvil aun no integran sus productos ERP para versiones móviles completas y además que sean multiplataforma.

Por lo anterior y mencionando que no se abarcara todo lo que un sistema ERP engloba y ofrece, pero si se desarrollara un software que ofrezca una solución móvil al estudio que antecede la instalación de una empresa.

## *Capítulo 2*

### MARCO TEORICO

En este capítulo se procederá a realizar el análisis y diseño del sistema tomando a consideración los objetivos antes mencionados y a aplicar la metodología del lenguaje de modelado unificado con lo que empezaremos describiendo los requisitos del sistema.

Por lo anterior a continuación se mostraran los procesos de diseño, desarrollo e implementación para la aplicación que desea crear.

#### **2.1 Programación Orientada a Objetos (OOP)**

La Programación Orientada a Objetos supone un cambio en la concepción del mundo de desarrollo de software, introduciendo un mayor nivel de abstracción que permite mejorar las características del código final. De manera muy básica, las aportaciones de este paradigma se pueden resumir en: **[5]**.

1. Conceptos de clase y objeto, que proporcionan una abstracción del mundo centrada en los seres y no en los verbos **[5]**.
2. Los datos aparecen encapsulados dentro del concepto de clase. El acceso a los datos se produce de manera controlada e independiente de la representación final de los mismos. Como consecuencia, se facilita el mantenimiento y la evolución de los sistemas, al desaparecer las dependencias entre distintas partes del sistema **[5]**.
3. Mediante conceptos como la composición, herencia y polimorfismo se consigue simplificar el desarrollo de sistemas. La composición y la

herencia nos permiten construir clases a partir de otras clases, aumentando en gran medida la reutilización [5].

## **2.2 Antecedentes**

1ª Etapa. Lenguajes Ensambladores. La unidad de programación es la instrucción, compuesta de un operador y los operandos. El nivel de abstracción que se aplica es muy bajo [6].

2ª Etapa. Lenguajes de Programación: Fortran, Algol, Cobol. Los objetos y operaciones del mundo real se podían modelar mediante datos y estructuras de control separadamente. En esta etapa el diseño del software se enfoca sobre la representación del detalle procedimental y en función del lenguaje elegido. Conceptos como: refinamiento progresivo, modularidad procedimientos y programación estructurada son conceptos básicos que se utilizan en esta etapa. Existe mayor abstracción de datos [6].

3ª Etapa. Se introducen en esta etapa los conceptos de abstracción y ocultación de la información [6].

4ª Etapa. A partir de los años setenta se trabaja sobre una nueva clase de lenguajes de simulación y sobre la construcción de prototipos tales como Simula-70 y basado en parte de éste, el Smalltalk. En estos lenguajes, la abstracción de datos tiene una gran importancia y los problemas del mundo real se representan mediante objetos de datos a los cuales se les añade el correspondiente conjunto de operaciones asociados a ellos. Términos como Abstracción de datos, objeto, encapsulación entre otros, son conceptos básicos sobre la que se fundamenta la POO [6].

## 2.2 Lenguaje de Unificado de Modelado (UML)

La metodología usada en el presente trabajo está fundamentada en la metodología de desarrollo de software UML.

**UML**, por sus siglas en inglés (Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados [7].

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo [7].

El UML combina lo mejor de

- Métodos centrados en datos
- Métodos centrados en escenarios de uso
- Métodos centrados en responsabilidades [6].

UML se utiliza para modelar visualmente:

- La interacción del sistema con el mundo externo
- El comportamiento del sistema
- La estructura del sistema
- La arquitectura del sistema
- Los componentes del sistema [6].



## **2.3 Diseño de la Base de Datos**

La Base de Datos diseñada en este trabajo usa la metodología Entidad-Relación que a continuación describiremos.

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos [9].

### **2.3.1 Modelo Entidad Relación**

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos. El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la representación del significado de los datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R. [10].

Hay tres nociones básicas que emplea el modelo de datos E-R: conjunto de entidades, conjunto de relaciones y atributos. [10].

### **2.3.1.1 Entidades**

Una entidad es una cosa u objeto en el mundo real que es distinguible de todos los demás objetos. Una entidad está formada por un conjunto de propiedades, y los valores para algún conjunto de propiedades pueden identificar una entidad de forma unívoca [10].

Un conjunto de entidades es un conjunto de entidades del mismo tipo que comparten las mismas propiedades, o atributos. El conjunto de todas las personas que son clientes en un banco dado, por ejemplo, se pueden definir como el conjunto de entidades cliente [10].

### **2.3.1.2 Atributos**

Los atributos describen propiedades que posee cada miembro de un conjunto de entidades. La designación de un atributo para un conjunto de entidades expresa que la base de datos almacena información similar concerniente a cada entidad del conjunto de entidades; sin embargo, cada entidad puede tener su propio valor para cada atributo. Posibles atributos del conjunto de entidades cliente son id-cliente, nombre-cliente, calle-cliente y ciudad-cliente, teléfono-cliente [10].

Los atributos pueden ser de la siguiente forma.

- Simples
- Compuestos
- Monovalorados
- Multivalorados

Los atributos simples son aquellos que no están divididos en subpartes. Los atributos compuestos, en cambio, se pueden dividir en

subpartes (es decir, en otros atributos). Por ejemplo, nombre-cliente podría estar estructurado como un atributo compuesto consistente en nombre, primer-apellido y segundo-apellido [10].

Los atributos Monovalorados son aquellos que tienen un solo valor para una entidad, tomando como ejemplo lo anterior el atributo id-cliente está ligado solo y a un único cliente, la relación es uno a uno, en cambio los atributos Multivalorados son atributos que la relación son uno a muchos por ejemplo el atributo teléfono-cliente en el cual un cliente puede tener uno o varios números de teléfonos [10].



Fig 2.1 Ejemplo de la entidad Cliente y sus atributos

### 2.3.1.1 Relaciones

Una relación es una asociación entre diferentes entidades. Por ejemplo, se puede definir una relación que asocie al cliente López con el número de préstamo P-15. Un conjunto de relaciones es un conjunto de relaciones del mismo tipo [10].

En la siguiente figura veremos cómo se expresan las relaciones en el modelo E-R. [10].

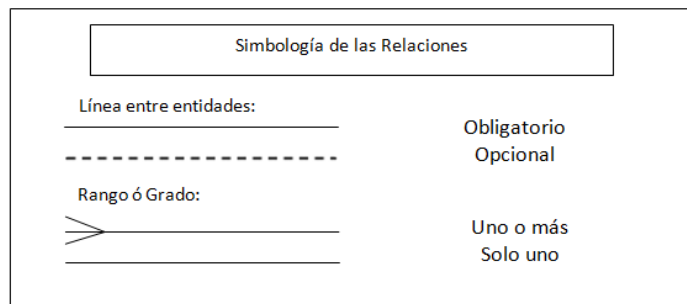


Fig 2.2 Expresión de las relaciones

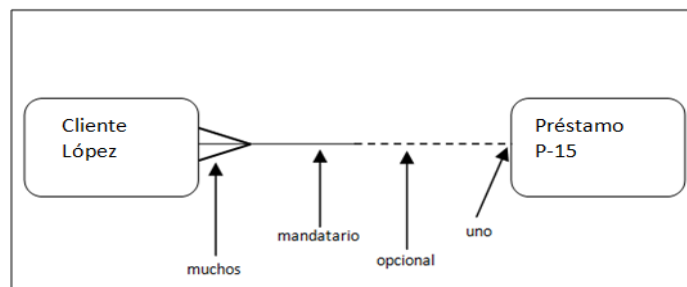


Fig 2.3 Ejemplo de relaciones entre entidades

Las relaciones pueden ser:

- Muchos-a-Uno: Estas son las más comunes y muestran que una relación tiene un grado de uno o más en una dirección y uno y solamente uno en la otra dirección [10].
- Muchos-a-Muchos: Hay un grado de uno o más en ambas direcciones y también son muy comunes. Usualmente son opcionales en ambas direcciones [10].
- Uno-a-Uno: Hay un grado de uno y sólo uno en ambas direcciones. Este tipo de relaciones es bastante raro, comúnmente indican que las dos entidades son realmente la misma entidad [10].

### 2.3.2 Normalización

La normalización es un proceso por el cual se aplica una serie de reglas a las relaciones de la Base de Datos, tras aplicar el modelo ER al modelo relacional y se utiliza para: [11].

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Para realizar el proceso de normalización, se debe determinar la forma normal de una relación a partir de su relación de atributos conocida como “dependencia funcional” [11].

#### 2.3.2.1 Dependencia Funcional

Una dependencia funcional es una conexión entre uno o más atributos (DF). Tomando el ejemplo anterior primer-apellido es dependiente del atributo nombre-cliente, existe una correlación ó dependencia entre un atributo y otro [11].

El concepto de DF, se tomó de las matemáticas elementales; y tiene tres propiedades ó también llamados axiomas de Armstrong, las cuales son: [11].

Dependencia o axioma *reflexiva*: Si "Y" está incluido en "X" entonces X implica Y [11].

El segundo axioma es la *aumentativa* y su planteamiento es el siguiente: Si “X implica Y” entonces “XZ implica YZ” [11].

El tercero y último es la *transitiva* la cual dice: Si “X implica Y implica Z” entonces “X implica Z” [11].

A partir de la dependencia funcional y sus axiomas se determina la forma normal en que se encuentra la relación, para que en acto seguido se pueda realizar el proceso de normalización [11].

### 2.3.2.1 Formas Normales

Dentro del proceso de normalización se hallan las formas normales (FN) las cuales se adecuan para cubrir las necesidades de la optimización de cualquier Base de Datos, en este trabajo solo trabajaremos hasta la tercera forma normal (3FN) y el creador de las primeras FN es Edgar F. Codd y son las siguientes: [11].

Primera Forma Normal (1FN): Una Base de Datos se encuentra en 1FN si satisface las siguientes condiciones:

1. No hay orden de arriba-abajo en las filas.
2. No hay orden de izquierda-derecha en las columnas.
3. Cada intersección fila-columna contiene exactamente un valor del dominio aplicable.
4. No hay filas duplicadas.
5. Todas las columnas son regulares.

Segunda Forma Normal (2FN): La condición para la 2FN es que la Base de Datos se encuentre en 1FN y además que ninguno de sus atributos sean funcionalmente dependientes de una clave primaria.

Tercera Forma Normal (3FN): La Base de Datos se encuentra en 3FN si antes esta en 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

El objetivo de la normalización es el adecuada agrupación y relación entre los atributos de cualquier Base de Datos; en función del software que usemos para su representación se mostraría de la siguiente forma:

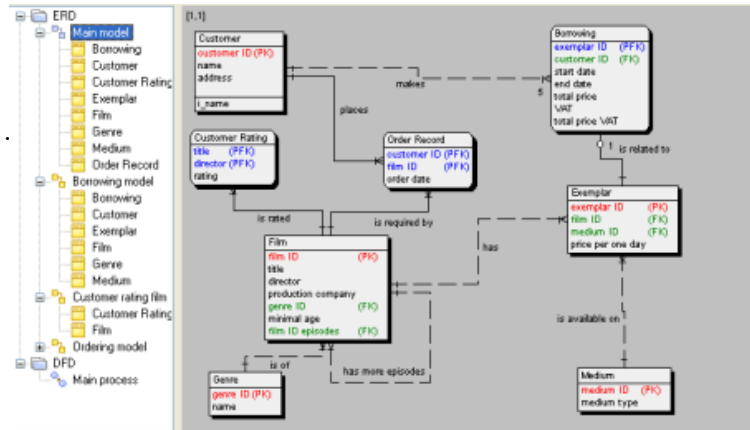


Fig 2.4 algo Diagrama del modelo entidad relación

## 2.4 Plataformas de Implementación

Las plataformas seleccionadas para este trabajo fueron necesarias para dar solución al problema planteado y además para integrar la aplicación a diferentes plataformas móviles, además de la necesidad de trabajar con tecnologías actualizadas y que a su vez brinden sencillez y transparencia en el desarrollo.

### 2.4.1 XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X. [12].

#### **2.4.2 Mac OS X (Snow Leopard)**

OS X, antes llamado Mac OS X, es una serie de sistemas operativos basados en Unix desarrollados, comercializados y vendidos por Apple Inc [13].

Mac OS X 10.6 Snow Leopard, fue lanzado públicamente el 28 de agosto de 2009 y es la séptima versión del sistema operativo de Apple. Diseñado para los poseedores de un ordenador con Mac OS X anteriores (Tiger) y procesador Intel. Es más rápido que la versión 10.5 Leopard y está disponible en el Apple Store [14].

Dentro de los requisitos generales para la instalación de Mac OS están:

- Ordenador Mac con procesador Intel
- 1 GB de memoria RAM
- 5 GB de espacio en disco disponible
- Unidad DVD para la instalación [14].

#### **2.4.4 Xcode**

Es el entorno de desarrollo integrado (IDE) de Apple Inc. El cual se suministra gratuitamente en AppStore, este incluye una colección de compiladores del proyecto GNU y puede compilar código C, C++, Objective C, Objective C++, Java y AppleScript. [15].



El Xcode podríamos considerarlo el equivalente a VisualStudio en Windows.

#### **2.4.4 ADT Android**

Son las herramientas para el desarrollo en Android (ADT) distribuido por Google Inc. Con el cual provee de los componentes necesarios para la construcción, prueba y depuración de aplicaciones para Android **[16]**.

Lo que provee el ADT de Android se encuentra:

- SDK Android Tools.
- Eclipse + ADT Plugin.
- Android Platforms Tools.
- The Latest Android Platforms.
- The Latest System Image Emulator **[16]**.

Dentro de los requisitos de sistema necesarios para su uso encontramos:

- Windows (xp o 7), Linux(Ubuntu o Lucid) ó Mac OS X (10.5.8)
- Sistemas operativos de 32-64 bit excepto en Mac OS X
- Instalación del Eclipse IDE
- Instalación del JDK 6
- Instalación de Apache Ant. 1.8 **[16]**.

El uso del ADT Android en este proyecto es para que el desarrollo de la aplicación también puede ser usado tanto en iPhone como en plataformas Android es por ello que particularmente utilizaremos el SDK de Android **[16]**.

### 2.4.5 Phonegap

Phonegap es un framework de código abierto que sirve para crear aplicaciones (apps) usando tecnologías como HTML 5, JavaScript y CSS. Un framework es una estructura digital de tecnologías de soporte definida o módulos de software concretos [17].

Phonegap comienza siendo desarrollado por la empresa canadiense Nitobi y posteriormente comprado por Adobe Systems para finalmente ser distribuido por Apache Cordova bajo la licencia open-source [17].

Dentro de las características que puede utilizar el API de Phonegap en los dispositivos móviles son:

- Acelerómetro.
- Cámara.
- Brújula.
- Contactos.
- Geolocalización.
- Medios de Comunicación.
- Red.
- Notificaciones (alertas, sonido y vibraciones).
- Almacenamiento [17].

Phonegap es compatible con sistemas operativos como IOS, Android, Blackberry OS, WebOS, Windows Phone, Symbian y Bada, particularmente en este trabajo toma total relevancia ya que con esta herramienta podemos compilar el mismo código para diferentes plataformas [17].

#### **2.4.6 Dreamweaver Cs5.5**

Es una aplicación de estudio creada por Adobe Systems la cual está destinada a la creación, diseño y edición de sitios, videos y aplicaciones web basados en estándares [18].

Esta aplicación es compatible con plataformas Mac OS X y Windows, su tipo de licencia es comercial por lo tanto adquirirlo tiene un costo dependiendo de la versión [18].

A partir de la versión MX se incorpora herramientas para la creación contenido dinámico como MySQL, Microsoft Acces, ASP, ASP.NET, ColdFusion, JSP, JavaScript y PHP, ect. En este trabajo utilizamos la versión Cs5.5 que incorpora tecnologías actualizadas como HTML 5, CSS3, JQueryMobile, Phonegap y Adobe Air [18].

Los requerimientos básicos de sistema para su utilización son:

- Windows (XP, Vista o 7)
- Mac OS X (10.5.7 o 10.6)
- 512 RAM
- 1.8 GB de Disco Duro
- Resolución de pantalla 1280x800
- Unidad de DVD-ROM. [18].

#### **2.4.7 MySQL**

Es un sistema de Base de Datos relacional multiusuario utilizado por gran parte usuarios y plataformas más comunes (Mac Os X, Windows y Linux) [19].

Dentro de sus principales características se encuentran:

- Portabilidad

- Sistema de privilegios y contraseñas que es flexible y seguro.
- Escritorio en C y C++
- Esta probado en varios compiladores
- Maneja un sistema de reserva de memoria rápido y eficaz [19].

MySQL es idea para proyectos de tipo web ya que hay baja concurrencia en la modificación y lectura de datos lo que lo convierte en una herramienta ideal para el desarrollo de este proyecto [19].

#### **2.4.8 PHP**

Es un lenguaje de programación diseñado para contenido web dinámico de lado del servidor el cual fue uno de los primeros que se podía incorporar directamente en los documentos HTML, el código es interpretado por un servidor web con un modulo de procesador PHP que genera la página web resultante [20].

PHP es usado por la gran mayoría de los servidores web al igual que todos lo sistema operativos, su tipo de licencia en open-source. Su gran parecido con lenguajes de programación como C ó Perl, permite que la mayoría de los programadores se adapten rápidamente en uso y control [20].

#### **2.4.9 HTML 5**

Por sus siglas en ingles (HyperText Markup Leguage), es un lenguaje de marcado para la elaboración de sitios web, “5” ya que es la quinta revisión para este leguaje estándar en la Wolrd Wide Web y es regulado por la W3C [22].

Las mejoras o novedades que incorpora este nuevo estándar son:

- Incorpora etiquetas multimedia (canvas 2D y 3D, audio, video)

- Incorpora etiquetas para gran contenido de datos (Datagrid, Details, Menu y Command)
- Mejoras en los formularios
- Nueva funcionalidad Drag & Drop (arrastrar objetos)
- Emplea etiquetas para manejar la web semántica.
- Incorpora nuevas API'S [22].

A pesar de que es un estándar a prueba la utilización de este dentro del proyecto es relevante ya que incorporamos otra plataforma dentro HTML 5 lo cual se convierte en el bosquejo principal de la aplicación [22].

#### **2.4.10 JavaScript**

Es un lenguaje de programación interpretado y estandarizado, es un lenguaje orientado a objetos su uso es principalmente del lado del cliente e implementado en navegadores web permitiendo sitios web más dinámicos y mejoras en la interfaz de usuario.

JavaScript se diseño con una sintaxis similar a C y adopta nombres y convenciones del lenguaje Java a pesar de no tener ninguna relación. Todos los navegadores modernos utilizan e interpretan el código JavaScript integrado dentro de las páginas web.

Sus características principales son:

- Imperativo y Estructurado
- Dinámico
- Funcional
- Prototípico

El uso más común de este lenguaje es forma funciones embebidas en las páginas HTML, para este proyecto su integración es para el manejo de datos por parte del cliente.

### 2.4.11 CSS3

Es un lenguaje de hojas de estilo usado para describir la presentación semántica de las páginas HTML y así proporcionarle un mejor aspecto y formato a las mismas, su integración también puede ser en documentos tipo XML, SVG y XUL.

Su sintaxis es muy sencilla basta con conformar unas palabras clave en inglés para especificar los nombres de varias propiedades de estilo y se emplean tres tipos de formas distintas:

- Estilo en línea (dentro de una etiqueta en la página HTML)
- Hoja de estilo interna (dentro el documento HTML)
- Hoja de estilo externa (se encuentra almacenada en un archivo diferente al archivo donde se almacena el código HTML)

Algunas de las ventajas que proporciona este lenguaje es el control y centralización del documento HTML, mejora la accesibilidad del documento y la optimización del mismo [3].

### 2.4.12 JQueryMobile

Es un framework basado en HTML 5 y desarrollado por JQuery para la creación de sitios web móviles, el cual nos permite generar aplicaciones cuya apariencia será siempre la misma independientemente del dispositivo móvil para que usemos siempre y cuando el dispositivo acepte HTML 5.

Este framework nos provee de herramientas que facilitan el desarrollo de aplicaciones móviles y su uso de igual forma es sencillo ya que incorpora componentes a los atributos de HTML 5. Dentro de sus características encontramos:

- Compatibilidad con varias plataformas móviles.
- Configuración basada en HTML 5

- Configuración Ajax-Powered
- Tematización en el marco de trabajo.
- La misma base de código se escala en cualquier pantalla.
- Esta construido bajo el núcleo de JQuery por lo que su aprendizaje es más fácil.
- Optimización de funciones nativas

Tal vez sus mayores atractivos son la optimización de aplicaciones nativas que mejoran su rendimiento en dispositivos móviles y la generación de elementos gráficos modificando el documento HTML que acelera sustancialmente la velocidad de desarrollo [24]. .

## *Capítulo 3*

### ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se procederá a realizar el análisis y diseño del sistema tomando a consideración los objetivos antes mencionados y a aplicar la metodología del lenguaje de modelado unificado con lo que empezaremos describiendo los requisitos del sistema.

#### **3.1 Requisitos del Sistema**

El sistema debe permitir la administración del mismo por medio de sesiones de usuario, en el cual se accederá por medio de un usuario y un password, solo los ingenieros que sean dados de alta en la Base de Datos podrán acceder al sistema.

Una vez ingresado a su sesión el sistema debe permitir la capturar el nombre del estudio técnico y el tipo de producción que se realizará, así mismo el sistema podrá obtener el tamaño óptimo del proyecto por medio de comparaciones de datos como los siguientes: tipo de zona, condiciones socioeconómicas, aspectos culturales, medio ambiente, comercio, industria y agricultura e institucionalidad.

De igual forma el sistema debe realizar la comparación de los datos ingresados por el usuario para determinar el tamaño óptimo; por medio de la asignación de datos como son: la superficie, capacidad de producción, capacidad de proceso, potencia instalada, población servida, área de influencia, personas atendidas, longitud y volumen.



### 3.1.1 Diagrama General de Casos de Uso

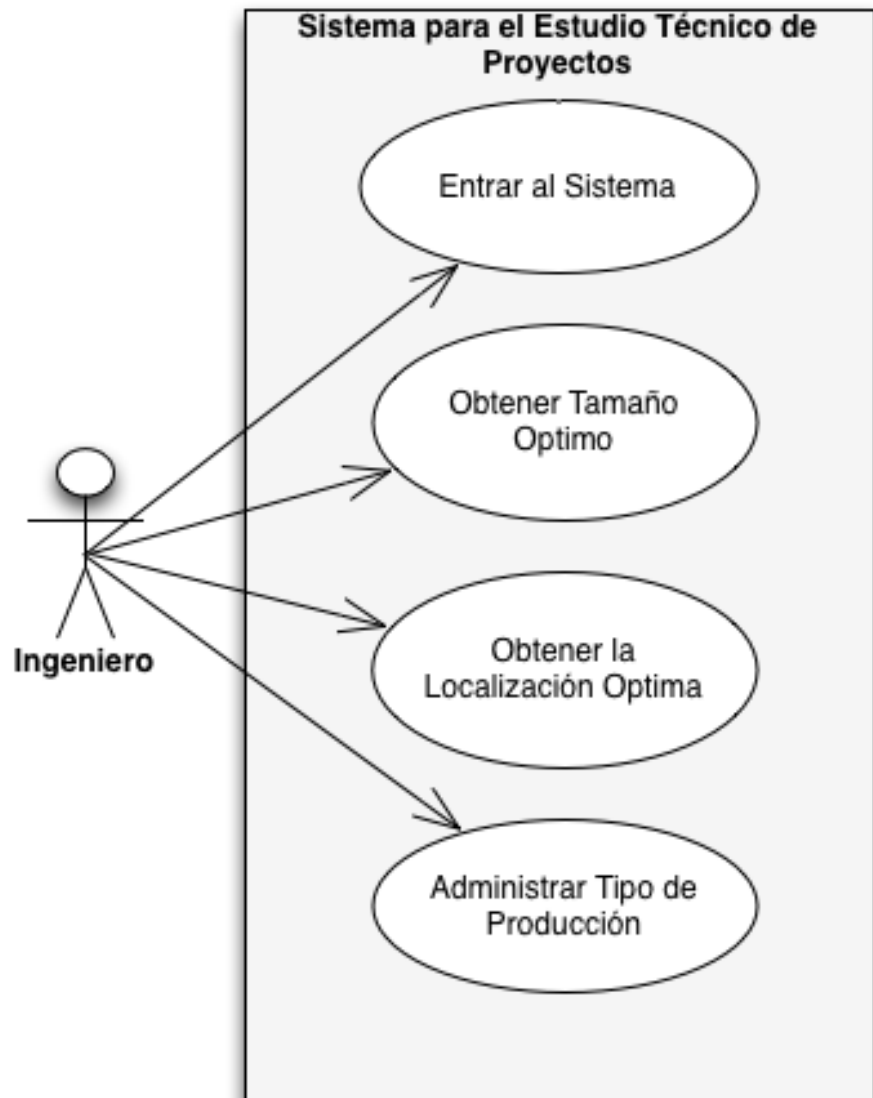


Fig 3.1 Diagrama General de Casos de Uso

En la figura 3.1 mostramos el diagrama general de casos de uso para el Sistema de Estudio Técnico de Proyectos, donde podremos apreciar el comportamiento que tendrá este sistema y quien interactúa con el sistema.

### 3.1.1 Especificación de Casos de Uso

La especificación de los casos de uso describe el modo en que el actor interactúa con el sistema, es una narración que describe el rol desempeñado por los actores en su interacción con el sistema [23].

<b>Especificación del Caso de Uso: Entrar al Sistema</b>			
Nombre:	Entrar al Sistema		
Descripción:	En este caso de uso el sistema permite el acceso a los usuarios mediante un usuario y un password.		
Autor:			
Fecha de Creación:	26/09/2013	Fecha de modificación:	26/09/2013
Actores:	Ingeniero		
Precondiciones:	El usuario debe estar ya dado de alta previamente		
Pos condiciones:	Deben haber registrado al usuario en la base de datos.		
Flujo Normal de Eventos			
<ol style="list-style-type: none"><li>1. El sistema muestra un formulario en la página inicial el cual solicita ingrese su usuario y su password.</li><li>2. El usuario ingresa su username y su password en el formulario y envía los datos para su verificación.</li><li>3. El sistema recibe los datos enviados y verifica la existencia de los mismos en la BD.</li><li>4. Si no existe el username o el password el sistema manda mensaje de error y pide se vuelvan a ingresar los datos.</li><li>5. Si existe el username y el password, el sistema da acceso al usuario y muestra la pantalla de inicio.</li></ol>			

Flujo Alternativo:
<p>Excepciones:</p> <ol style="list-style-type: none"> <li>1. Que el usuario no exista ó este incorrecto el sistema manda mensaje de error.</li> <li>2. Que el password no exista ó este incorrecto el sistema manda mensaje de error.</li> <li>3. No exista la información con los criterios de búsqueda</li> </ol>

<b>Especificación del Caso de Uso: Obtener el Tamaño Óptimo</b>			
Nombre:	Obtener el Tamaño Óptimo		
Descripción:	En este caso de uso, el sistema calcula y verifica cual es el tamaño óptimo mediante una escala.		
Autor:			
Fecha de Creación:	26/09/2013	Fecha de modificación:	26/09/2013
Actores:	Ingeniero		
Precondiciones:	El usuario debe haber ingresado a su sesión y haber seleccionado el proyecto a trabajar.		
Pos condiciones:	Finalización de la comparación obteniendo el tamaño óptimo.		
Flujo Normal de Eventos			
<ol style="list-style-type: none"> <li>1. El usuario ingresa a la pestaña de "Tamaño óptimo".</li> <li>2. El sistema muestra un formulario.</li> <li>3. El usuario elige el proyecto a trabajar.</li> <li>4. El usuario inserta los datos dentro del formulario y envía los datos para ser</li> </ol>			

<p>guardados.</p> <ol style="list-style-type: none"> <li>El sistema recibe los datos y los inserta dentro de la BD.</li> <li>El usuario presiona el botón de “validar” para comparar las dos zonas.</li> <li>El sistema pondera los valores ingresados y devuelve el resultado de acuerdo a la cantidad de producción y cantidad de demanda.</li> </ol>
<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> <li>El usuario seleccione el botón agregar y el sistema pregunta “Desea agregar nuevos datos”</li> <li>Si el usuario selecciona “cancelar” el sistema regresa a la página anterior.</li> <li>Si el usuario selecciona “si” el sistema limpia los formularios para el ingreso de nuevos datos.</li> </ol>
<p>Excepciones:</p> <ol style="list-style-type: none"> <li>Que el usuario no escoja ningún proyecto a trabajar.</li> <li>Que el usuario no inserte datos dentro del formulario.</li> </ol>

<b>Especificación del Caso de Uso: Obtener la Localización Óptima</b>			
Nombre:	Obtener la Localización Óptima		
Descripción:	En este caso de usos el sistema calcula y verifica cual es la localización óptima.		
Autor:	Ingeniero		
Fecha de Creación:	26/09/2013	Fecha de modificación:	26/09/2013
Actores:	Ingeniero		
Precondiciones:	El usuario debe haber ingresado a su sesión y haber seleccionado el proyecto a trabajar.		
Pos condiciones:	Finalización con éxito del ingreso de los datos de la zona1		

	y zona 2.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> <li>1. El usuario ingresa a la pestaña de “Localización óptima”.</li> <li>2. El sistema muestra dos pestañas “zona 1” y “zona 2” con su respectivo formulario.</li> <li>3. El usuario elige el proyecto a trabajar.</li> <li>4. El usuario inserta los datos dentro del formulario de “zona 1” y envía los datos para ser guardados.</li> <li>5. El sistema recibe los datos y los inserta dentro de la BD.</li> <li>6. El usuario inserta los datos dentro del formulario de “zona 2” y envía nuevamente los datos para ser guardados.</li> <li>7. El sistema recibe los datos del formulario de “zona 2” y los inserta dentro de la BD.</li> <li>8. El usuario presiona el botón de “validar” para comparar las dos zonas.</li> <li>9. El sistema pondera las dos zonas y muestra la de mejor resultado.</li> </ol>	
Flujo Alternativo:	
<ol style="list-style-type: none"> <li>1. El usuario seleccione el botón agregar y el sistema pregunta “Desea agregar nuevos datos”</li> <li>2. Si el usuario selecciona “cancelar” el sistema regresa a la página anterior.</li> <li>3. Si el usuario selecciona “si” el sistema limpia los formularios para el ingreso de nuevos datos.</li> </ol>	
Excepciones:	
<ol style="list-style-type: none"> <li>1. Que el usuario no escoja ningún proyecto a trabajar.</li> <li>2. Que el usuario no inserte datos dentro del formulario.</li> </ol>	

<b>Especificación del Caso de Uso: Administrar el Tipo de Producción</b>	
Nombre:	Administrar el Tipo de Producción
Descripción:	En este caso de usos el sistema da a escoger al usuario el tipo de producción que requiere “manufactura ó servicios”

	y lo guarda en la BD.		
Autor:			
Fecha de Creación:	26/09/2013	Fecha de modificación:	26/09/2013
Actores:	Ingeniero		
Precondiciones:	El usuario debe haber ingresado a su sesión.		
Pos condiciones:	Guardar los datos que ingreso el usuario respecto a tipo de producción que requiere.		
Flujo Normal de Eventos			
<ol style="list-style-type: none"> <li>3. El sistema muestra al usuario un formulario para que ingrese el nombre del estudio técnico, y el tipo de producción que necesita.</li> <li>4. El usuario ingresa el nombre del estudio técnico y el tipo de producción y envía los datos.</li> <li>5. El sistema ingresa los datos insertados en el formulario a la BD.</li> <li>6. El sistema muestra una lista con la selección de tipos de producción “Manufactura ó Servicios”</li> <li>7. El usuario selecciona el tipo de producción y envía datos.</li> <li>8. El sistema ingresa los datos al sistema.</li> </ol>			
Flujo Alternativo:			
<ol style="list-style-type: none"> <li>1. El usuario seleccione el botón agregar y el sistema pregunta “Desea agregar un nuevo proyecto”</li> <li>2. Si el usuario selecciona “cancelar” el sistema regresa a la página anterior.</li> <li>3. Si el usuario selecciona “sí” el sistema limpia el formulario para el ingreso de nuevos datos.</li> </ol>			
Excepciones:			
<ol style="list-style-type: none"> <li>1. Que el usuario no ingrese el nombre del estudio técnico.</li> <li>2. Que el usuario no seleccione el tipo de producción.</li> </ol>			

### 3.1.1 Escenarios

Los escenarios nos permiten tener un ejemplo claro de las diferentes situaciones en la que se encuentra el sistema por lo que cada caso de uso tendrá diferentes escenarios [23].

En cualquier sistema nos encontraremos con dos tipos de escenarios los de primarios o de flujo básico, los secundarios: alternos o de excepción y se elaboran los necesarios para entender el problema [23].

Considerando lo anterior a continuación de describirán los escenarios más representativos de los casos de usos ya mencionados.

Caso de Uso: Entrar al Sistema

#### Escenario 1

1. Bernardo debe haber sido dado de alta en la Base de Datos para su ingreso e instalado la aplicación.
2. Bernardo ingresa a la aplicación “MyPruebaApp” instalada en su Smartphone.
3. Bernardo selecciona el campo “nombre” y escribe su nombre de usuario.
4. Bernardo selecciona el segundo campo “contraseña” y escribe su contraseña en el campo.
5. El sistema valida el nombre de usuario y su contraseña en la Base de Datos.
6. El sistema al encontrar coincidencia con los datos insertados permite el acceso a Bernardo.

#### Escenario 2

1. Isaac debe haber sido dado de alta en la Base de Datos para su ingreso e instalado la aplicación.
2. Isaac ingresa a la aplicación “MyPruebaApp” instalada en su Smartphone.
3. Isaac selecciona el campo “nombre” y escribe su nombre de usuario.
4. Isaac selecciona el segundo campo “contraseña” y escribe su contraseña en el campo.

5. El sistema valida los datos insertados por Isaac y no encuentra coincidencia con los datos guardados en la Base de Datos.
6. El sistema manda un mensaje de error “usuario o contraseña incorrectos”.

#### Caso de Uso: Administrar tipo de producción

##### Escenario 1

1. Bernardo realizo el caso de uso “Ingresar al sistema”.
2. El sistema muestra una pantalla de bienvenida a Bernardo.
3. Bernardo da clic al botón de continuar.
4. El sistema muestra el formulario para ingresar el nombre del proyecto y el tipo de producción.
5. Bernardo selecciona el campo “Nombre de proyecto” en ingresa el nombre deseado.
6. Bernardo selecciona el tipo de producción que requiere “Manufactura o Servicios”.
7. El sistema recibe los datos ingresado en el formulario y los inserta dentro de la Base Datos.

##### Escenario 2

1. Isaac realizo el caso de uso “Ingresar al sistema”.
2. El sistema muestra una pantalla de bienvenida a Isaac.
3. Isaac da clic al botón de continuar.
4. El sistema muestra el formulario para ingresar el nombre del proyecto y el tipo de producción.
5. Isaac no inserta datos en el campo “nombre de proyecto” y no selecciona ningún tipo de producción “Manufactura o Servicios”.
6. El sistema verifica que no hay datos insertados en el formulario y manda un mensaje de error “No hay datos en el formulario”.

##### Escenario 3

1. Hugo realizo el caso de uso “Ingresar al sistema”.
2. El sistema muestra una pantalla de bienvenida a Hugo.
3. Hugo da clic al botón de continuar.
4. El sistema muestra el formulario para ingresar el nombre del proyecto y el tipo de producción.



5. Hugo ingresa el nombre del proyecto en el campo “nombre de proyecto” y selecciona el tipo de producción.
6. Hugo requiere otro proyecto nuevo y selecciona el botón de “agregar”.
7. El sistema muestra una nueva ventana con las opciones de “cancelar” y “si, nuevo proyecto”.
8. Hugo selecciona “si, nuevo proyecto”.
9. El sistema hace una limpieza del formulario para insertar nuevamente los datos.

#### Escenario 4

1. Hugo realiza el caso de uso “Ingresar al sistema”.
2. El sistema muestra una pantalla de bienvenida a Hugo.
3. Hugo da clic al botón de continuar.
4. El sistema muestra el formulario para ingresar el nombre del proyecto y el tipo de producción.
5. Hugo ingresa el nombre del proyecto en el campo “nombre de proyecto” y selecciona el tipo de producción.
6. Hugo requiere otro proyecto nuevo y selecciona el botón de “agregar”.
7. El sistema muestra una nueva ventana con las opciones de “cancelar” y “si, nuevo proyecto”.
8. Hugo selecciona “cancelar”.
9. El sistema no realiza ninguna acción y regresa a la página del formulario con datos anteriormente ingresados.

#### Caso de Uso: Obtener el tamaño óptimo

##### Escenario 1

1. Bernardo realiza el caso de uso “Ingresar al sistema” y “Administrar tipo de producción”.
2. El sistema muestra una pantalla con dos pestañas “zona 1” y “zona 2” con un formulario en cada pestaña.
3. Bernardo selecciona el “proyecto a trabajar”.
4. Bernardo llena el formulario de “zona 1” y envía los datos para que sea guardados.
5. El sistema recibe los datos del primer formulario y los inserta en la Base de Datos.
6. Bernardo selecciona la pestaña “zona 2” y llena el segundo formulario.
7. El sistema recibe los datos del segundo formulario y los inserta en la Base de Datos.

8. Bernardo da clic al botón de “validar” para comparar las zonas.
9. El sistema valida los datos de los dos formularios y muestra el resultado.

#### Escenario 2

1. Isaac realizo el caso de uso “Ingresar al sistema” y “Administrar tipo de producción”.
2. El sistema muestra una pantalla con dos pestañas “zona 1”y “zona 2” con un formulario en cada pestaña.
3. Isaac selecciona el “proyecto a trabajar”.
4. Isaac llena el formulario de “zona 1” y envía los datos para que sea guardados.
5. El sistema recibe los datos del primer formulario y los inserta en la Base de Datos.
6. Isaac selecciona la pestaña “zona 2” y deja en blanco el formulario.
7. Isaac da clic al botón de “validar” para comparar las zonas.
8. El sistema valida los datos y verifica que el segundo formulario no tiene datos, por lo que manda un mensaje de error “no hay datos en el formulario”.

#### Escenario 3

1. Hugo realizo el caso de uso “Ingresar al sistema” y “Administrar tipo de producción”.
2. El sistema muestra una pantalla con dos pestañas “zona 1”y “zona 2” con un formulario en cada pestaña.
3. Hugo selecciona el “proyecto a trabajar”.
4. Hugo llena el formulario de “zona 1” y “zona 2”, envía los datos para que sean guardados.
5. El sistema recibe los datos y los inserta en la Base de Datos.
6. Hugo da clic en el boto “validar”.
7. El sistema valida las zonas y muestra el resultado.
8. Hugo quiere revisar mas zonas y da clic en el botón “Agregar”.
9. El sistema muestra una pantalla donde pregunta “Desea validar mas zonas”.
10. El usuario selecciona la opción “si, deseo agregar”.
11. El sistema limpia los campos de los formularios.

Caso de Uso: Obtener la localización óptima

### Escenario 1

10. Bernardo realizo el caso de uso “Ingresar al sistema” y “Administrar tipo de producción”.
11. El sistema muestra una pantalla con dos pestañas “zona 1”y “zona 2” con un formulario en cada pestaña.
12. Bernardo selecciona el “proyecto a trabajar”.
13. Bernardo llena el formulario de “zona 1” y envía los datos para que sea guardados.
14. El sistema recibe los datos del primer formulario y los inserta en la Base de Datos.
15. Bernardo selecciona la pestaña “zona 2” y llena el segundo formulario.
16. El sistema recibe los datos del segundo formulario y los inserta en la Base de Datos.
17. Bernardo da clic al botón de “validar” para comparar las zonas.
18. El sistema valida los datos de los dos formularios y muestra el resultado.

### Escenario 2

9. Isaac realizo el caso de uso “Ingresar al sistema” y “Administrar tipo de producción”.
10. El sistema muestra una pantalla con dos pestañas “zona 1”y “zona 2” con un formulario en cada pestaña.
11. Isaac selecciona el “proyecto a trabajar”.
12. Isaac llena el formulario de “zona 1” y envía los datos para que sea guardados.
13. El sistema recibe los datos del primer formulario y los inserta en la Base de Datos.
14. Isaac selecciona la pestaña “zona 2” y deja en blanco el formulario.
15. Isaac da clic al botón de “validar” para comparar las zonas.
16. El sistema valida los datos y verifica que el segundo formulario no tiene datos, por lo que manda un mensaje de error “no hay datos en el formulario”.

### Escenario 3

12. Hugo realizo el caso de uso “Ingresar al sistema” y “Administrar tipo de producción”.
13. El sistema muestra una pantalla con dos pestañas “zona 1”y “zona 2” con un formulario en cada pestaña.

14. Hugo selecciona el “proyecto a trabajar”.
15. Hugo llena el formulario de “zona 1” y “zona 2”, envía los datos para que sean guardados.
16. El sistema recibe los datos y los inserta en la Base de Datos.
17. Hugo da clic en el boto “validar”.
18. El sistema valida las zonas y muestra el resultado.
19. Hugo quiere revisar mas zonas y da clic en el botón “Agregar”.
20. El sistema muestra una pantalla donde pregunta “Desea validar mas zonas”.
21. El usuario selecciona la opción “cancelar”.
22. El sistema regresa al usuario a la página anterior sin realizar acción alguna.

### **3.2 Modelo Conceptual**

El sistema debe permitir la administración del mismo por medio de sesiones de usuario, en el cual se accederá por medio de un usuario y un password, solo los ingenieros que sean dados de alta en la Base de Datos podrán acceder al sistema.

Una vez ingresado a su sesión el sistema debe permitir la capturar el nombre del estudio técnico y el tipo de producción que se realizará, así mismo el sistema podrá obtener el tamaño óptimo del proyecto por medio de comparaciones de datos como los siguientes: tipo de zona, condiciones socioeconómicas, aspectos culturales, medio ambiente, comercio, industria y agricultura e institucionalidad.

#### **3.2.1 Diagrama de Clases**

A continuación procedemos a aplicar el concepto del paradigma orientado a objetos el cual permite detectar los objetos y abstraerlo a clases.

Tomando en cuenta los requerimientos del sistema pudimos plantear un modelo para identificar las clases que integran el sistema, en

la siguiente figura el diagrama de clases de nuestro sistema y de igual forma procederemos a describir cada clase del mismo.

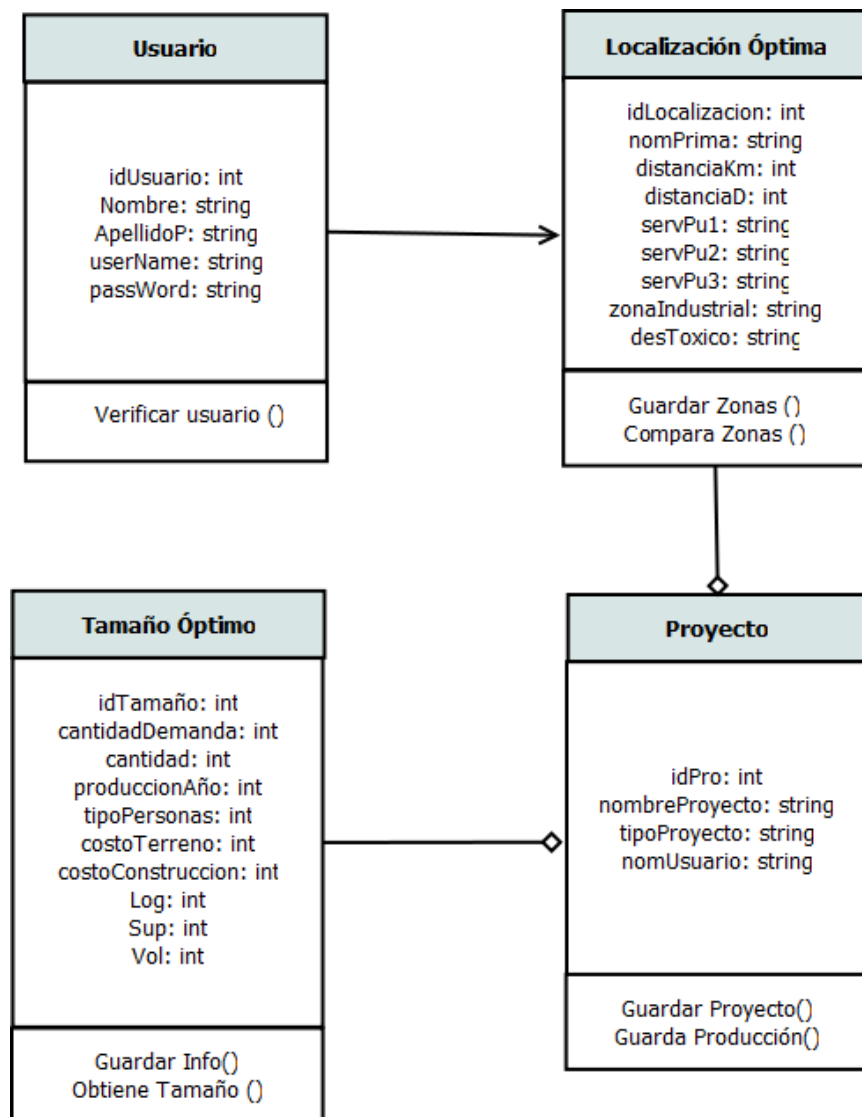


Fig 3.2 Diagrama de Clases (Modelo Conceptual)

### 3.2.2 Diccionario de Modelo

El diccionario de modelo describe las clases anteriormente mencionadas.

Nombre: Usuarios

Información acerca de las personas que tienen acceso, goce y uso del sistema (Realizar comparaciones entre zonas).

Nombre: Tipo de Producción

Información acerca del tipo de producción que desea o pretende realizar, y el nombre que quiere proporcionar al proyecto.

Nombre: Tamaño Óptimo

Información acerca del tamaño de la producción, estudio de mercado, tecnología utilizada, etc. Necesaria para poder sacar un promedio entre zonas y obtener la zona con mejor resultado.

Nombre: Localización Óptima

Información acerca de la distancia de la materia prima, del mercado meta, tipo de desecho, etc. Necesarios para hacer una comparación entre dos zonas obteniendo el mejor porcentaje de ellas.

## 3.3 Diseño del Sistema

### 3.2.1 Modelo Conceptual

Entraremos a la conceptualización del entorno de desarrollo del sistema que se divide en dos partes:

El modelo estático que es aquel que permite la estructura del modelo de análisis y el modelo dinámico que muestra las responsabilidades e interacciones dentro del sistema.

El inicio de la construcción de este modelo se realiza con un análisis de los casos de uso y de los requerimientos funcionales, para este proyecto es suficiente con la identificación de las clases: entidad, limite y control tratadas anteriormente.

### 3.2.2 Colección de Clases de Análisis

Nombre: Entrar al Sistema

Entidad	Limite	Control
Usuario	PantallaIngresoUsuario	IngresarAlSistema
	PantallaInicioAlSistema	
	VentanaMensajeError	

Nombre: Administrar Tipo de Producción

Entidad	Limite	Control
Usuario	PantallaBienvenidaSistema	IngresarNombreProyec
TipodeProducción	CampoCapturaNomPro	SeleccionarTipoProyec
	CampoSelecciónTipoPro	AgregarNuevoProyec
	PantallaAgregarNuevo	
	PantallaCancelarOpción	

Nombre: Obtener Tamaño Óptimo

Entidad	Limite	Control
Usuario	PantallaFormZona1	SeleccionarNombreProyec
TipodeProducción	PanatallaFormzona2	IngresarDatosForm

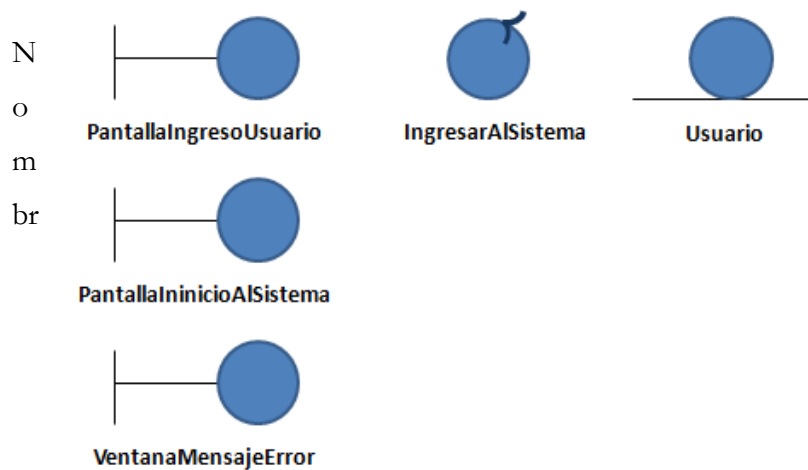
ón		
TamañoÓptimo	VentanaValidaZonas	AgregarNuevoZona
	VentanaResultCompara	ValidarZonas
	PantallaAgregarNuevo	
	PantallaCancelarOpción	

Nombre: Obtener Localización Óptima

Entidad	Limite	Control
Usuario	PantallaFormZona1	SeleccionarNombreProyec
TípodeProducción	PanatallaFormzona2	IngresarDatosForm
LocalizaciónÓptima	VentanaValidaZonas	AgregarNuevoZona
	VentanaResultCompara	ValidarZonas
	PantallaAgregarNuevo	
	PantallaCancelarOpción	

### 3.2.3 Diagramas de clases de análisis

Ya identificado las clases de análisis procedemos a elaborar los diagramas de clases.





e: Entrar al Sistema

Fig 3.3 Diagrama de Clases Entrar al Sistema

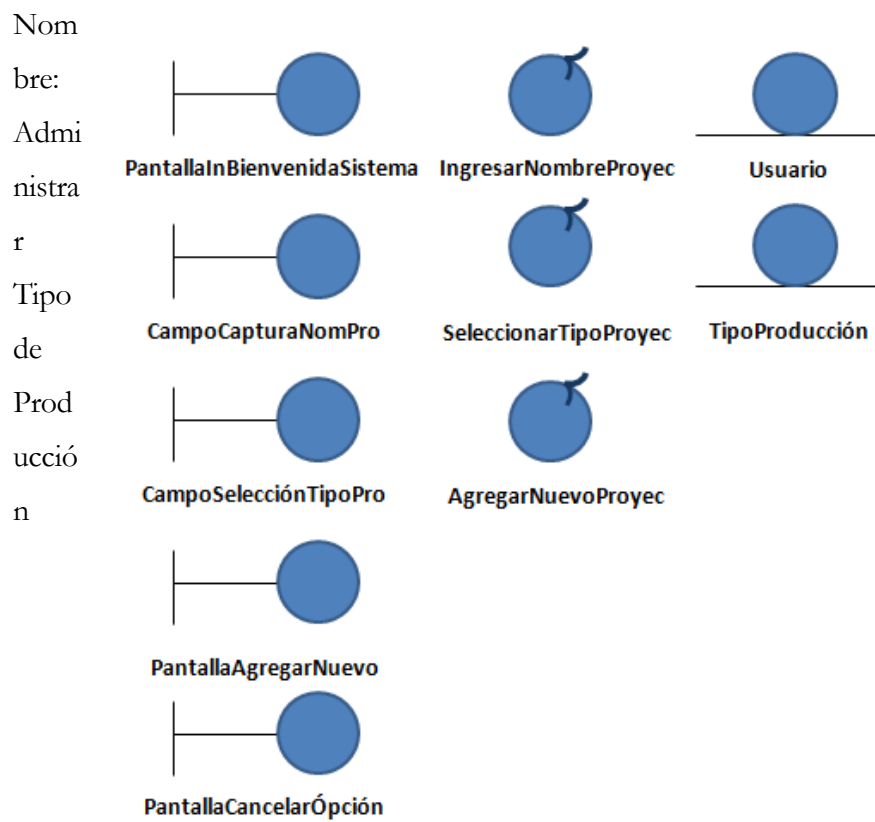
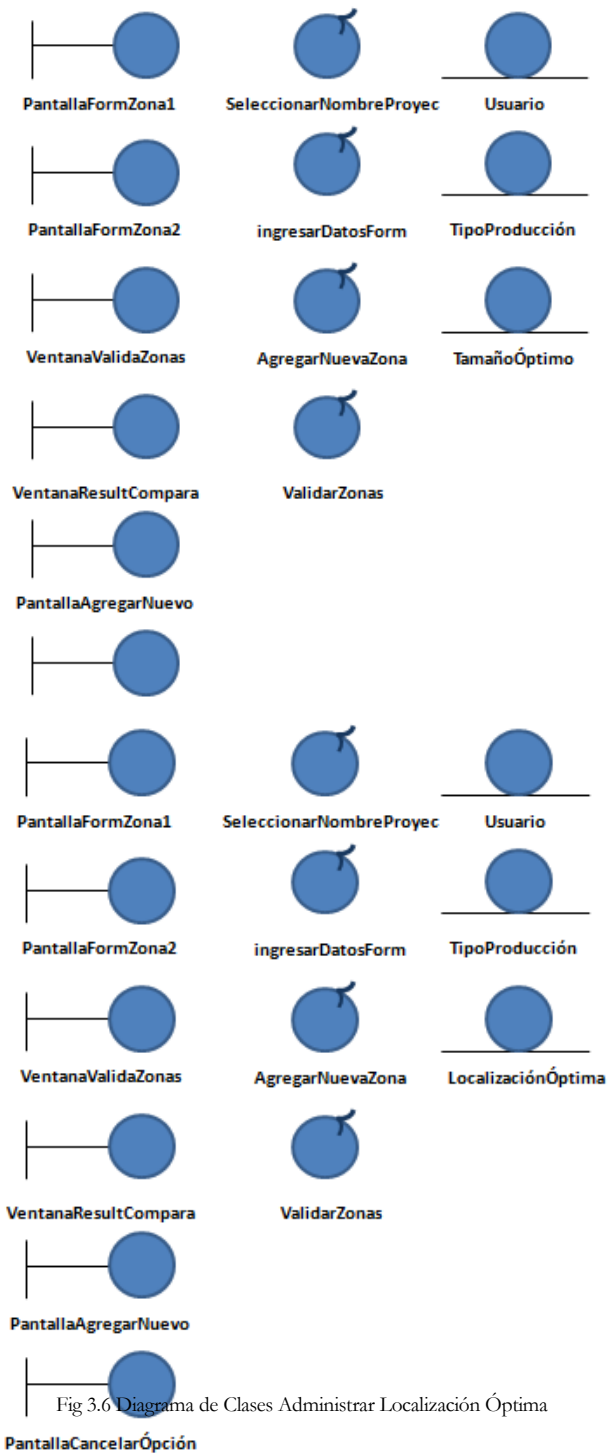


Fig 3.4 Diagrama de Clases Administrar Tipo de Producción

Nombre: Administrar Tamaño Óptimo



Óptima

Nombre:  
Administrar  
Localización

Fig 3.6 Diagrama de Clases Administrar Localización Óptima

### **3.3.4 Diagrama de Secuencia**

A continuación vamos a realizar los diagramas de secuencia para cada caso de uso los cuales nos sirven para modelar las interacciones entre los objetos ordenados y su respectiva los cuales incluyen:

- Los objetos participan en la secuencia con su línea de vida.
- Los mensajes representaran el flujo de eventos, y
- El enfoque de control sobre los eventos.

Vamos a proceder con el primer caso de uso:

Nombre: Entrar al Sistema

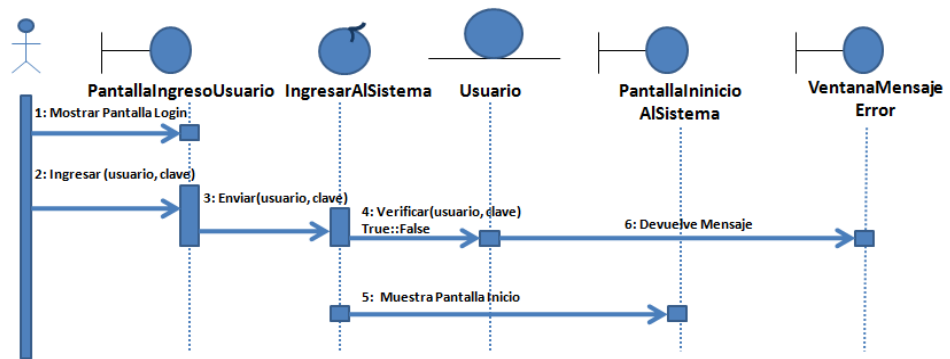
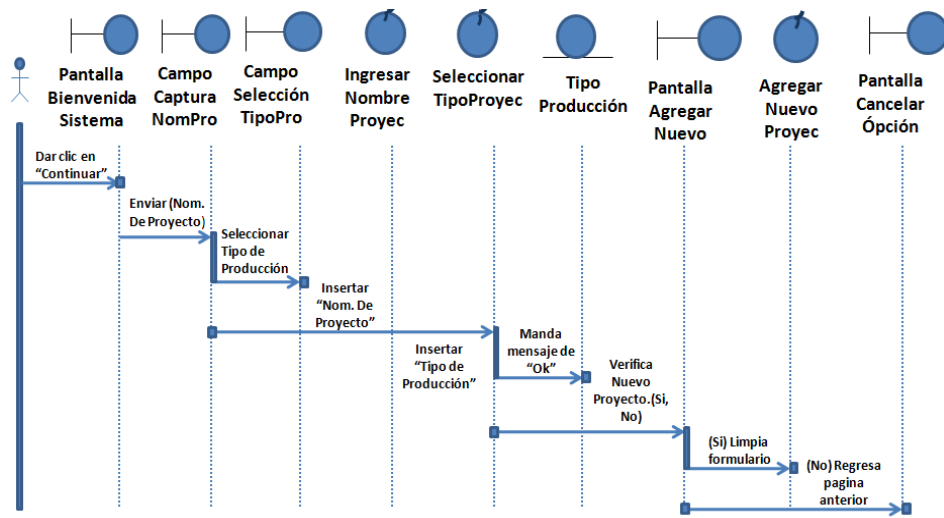


Fig 3.7 Diagrama de Secuencia Entrar al Sistema

Nombre: Administrar Tipo de Producción



Nombre:

Fig 3.8 Diagrama de Secuencia Tipo de Producción

Administrar Tamaño Óptimo

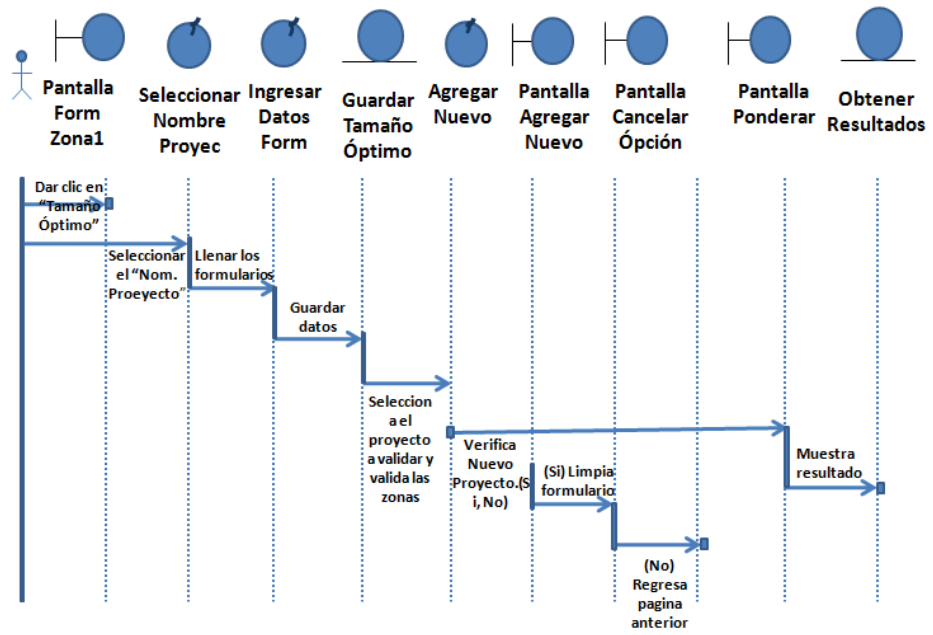


Fig 3.9 Diagrama de Secuencia Tamaño Óptimo

### Nombre: Administrar Localización Óptima

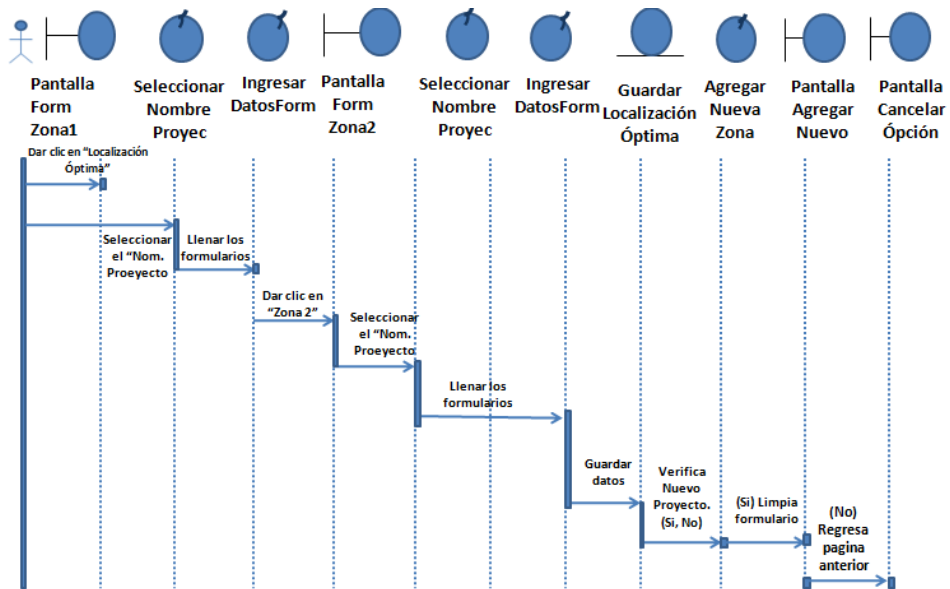


Fig 3.10 Diagrama de Secuencia Localización Óptima

### 3.3.4 Diagrama de Colaboración

Son una forma de representar los mensajes intercambiados por un conjunto de objetos, en estos diagramas se muestran los objetos con los enlaces entre ellos cuando hay más de una interacción y se señalan como dibujos entre los objetos.

Este tipo de diagramas incluye:

- Objetos
- Enlaces entre ellos
- Mensajes intercambiados entre objetos
- Flujo de datos entre objetos

A continuación mostraremos dichos diagramas:

Nombre: Entrar al Sistema

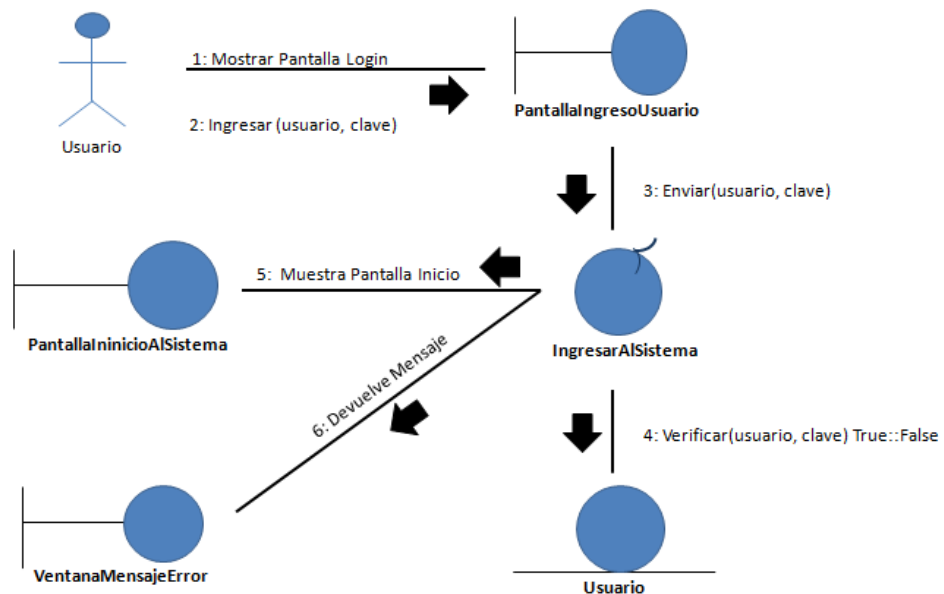


Fig 3.11 Diagrama de Colaboración Entrar al Sistema

Nombre: Tipo de Producción

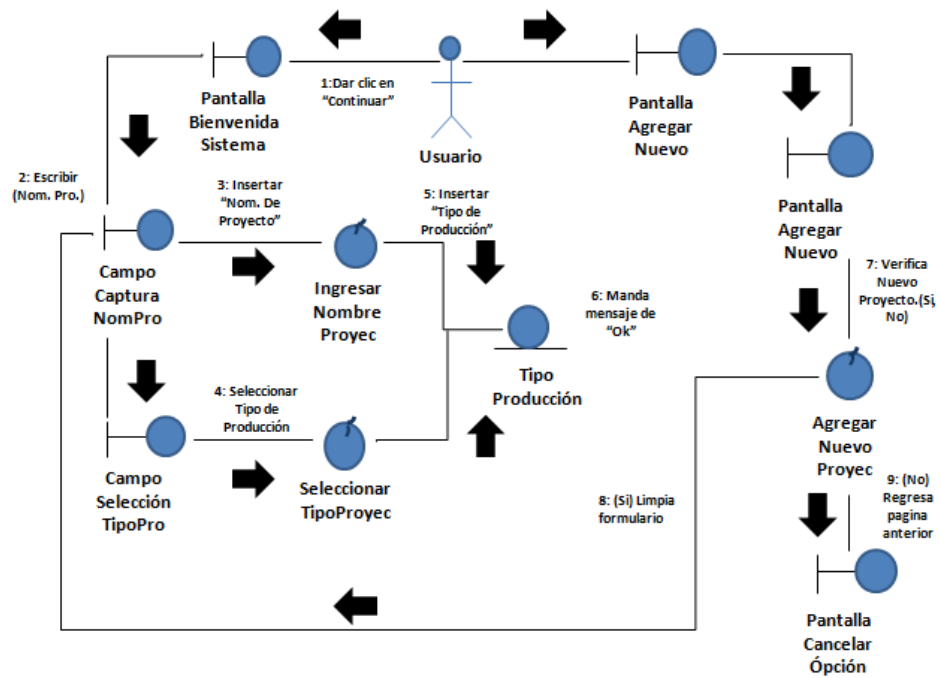
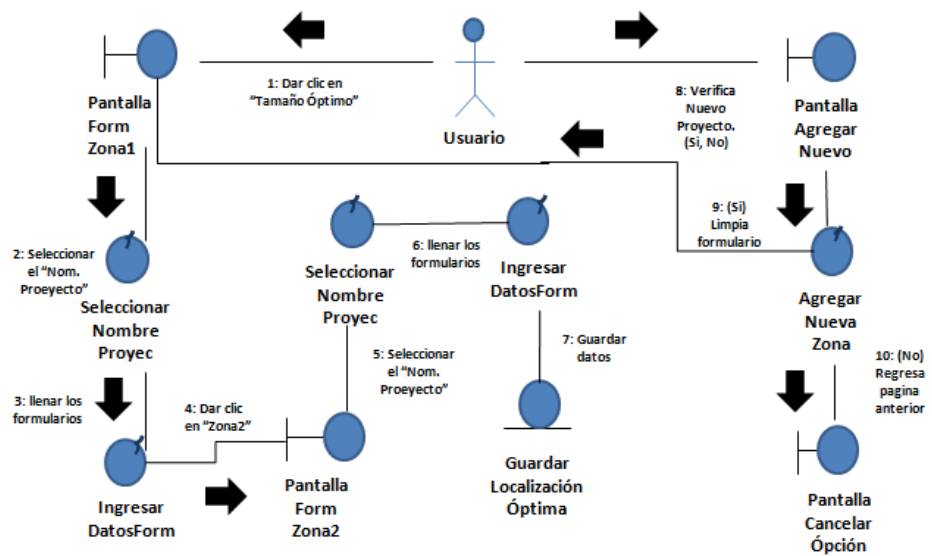


Fig 3.12 Diagrama de Colaboración tipo de Producción

Nombre: Localización Óptima



46  
Fig 3.13 Diagrama de Localización Óptima

Nombre: Tamaño Óptimo

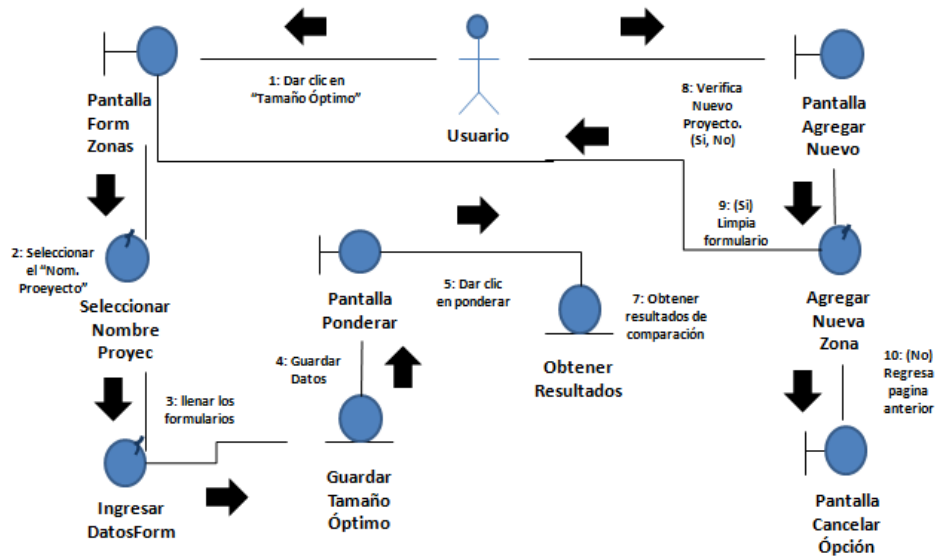


Fig 3.14 Diagrama de Tamaño Óptimo



## DISEÑO DE LA BASE DE DATOS

En este capítulo definiremos la estructura de los datos que debe tener la base de datos de este sistema en particular y las relaciones que existen entre ellos, para realizar esto necesitaremos de las teorías que permiten diseñar el modelo de la base de datos [4].

### **4.1 Modelo Entidad Relación**

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de un conjunto de objetos básicos llamados entidades y de relaciones entre estos objetos. En este capítulo se procederá a realizar el análisis y diseño del sistema tomando a consideración los objetivos antes mencionados y a aplicar la metodología del lenguaje de modelado unificado con lo que empezaremos describiendo los requisitos del sistema [4].

#### **4.1.1 Entidades**

Una entidad es una “cosa” u “objeto” en el mundo real que podemos distinguir de otros objetos y del que nos interesan sus propiedades [4].

El método entidad-relación parte del hecho de que uno de los resultados del análisis del sistema es la comprensión clara de cuáles son las entidades incluidas en él [10].

Considerando lo anterior y tomando en cuenta los requerimientos del sistema encontramos que la clases obtenidas anteriormente las cuales representan persistente del sistema, tomaremos dichas clases como entidades y anexaríamos una nueva llamada “Ponderar” en el presente

modelo para poder obtener el resultado de comparación de las entidades que a continuación en listaremos:

- Usuario
- Proyecto
- Localización Óptima
- Tamaño Óptimo
- Ponderar

#### 4.1.2 Atributos

Los atributos son información acerca de una entidad que necesita ser conocida o descrita. Los atributos representan un tipo de descripción o detalles para cada instancia [10].

Por lo anterior tenemos que los atributos que tendrían nuestras entidades serian las siguientes:

- Usuario (idUsuario, Nombre, ApellidoP, userName, passWord).
- Proyecto (idPro, nombreProyecto, tipoProyecto, nomUsuario).
- Localización Óptima (idLocalización, nomPrima, distanciaKm, distanciaD, servPu1, servPu2, servPu3, zonaIndustrial, desToxico).
- Tamaño Óptimo (idTamaño, cantidadDemandada, cantidadProduccion, cantidad\*, produccionAño,

tipoPersonas, costoTerreno, costoConstruccion, Log, Sup, Vol).

- Ponderar (vid, vdistanciaKm, vdistanciaD, vservPu1, vservPu2, vservPu3, vzonaIndustrial, vdesToxico)

#### 4.1.3 Identificadores Únicos

Una entidad debe ser distinguible del resto de los objetos del mundo real. Esto hace que para toda Entidad sea posible encontrar un conjunto de atributos que permitan identificarla [4].

Ahora procedemos a puntualizar los identificadores únicos de las entidades antes mencionadas.

- Usuario (idUsuario)
- Localización Óptima (idLocalización)
- Tamaño(idTamaño)
- Ponderar(vid)
- 

Así mismo, detallaremos los atributos de cada entidad como sigue:

Usuario

Atributo	Descripción	Tipo	Longitud	Rango
idUsuario	Identificador Único	Entero		0...*
Nombre	Nombre del usuario	Carácter	20	a-z, A-z
ApellidoP	Apellido Paterno	Carácter	20	a-z, A-z
userName	Nombre de Usuario	Carácter	20	a-z, A-z
passWord	Password del	Carácter	20	a-z, A-z,

	Usuario			0...9
--	---------	--	--	-------

### Proyecto

Atributo	Descripción	Tipo	Longitud	Rango
idPro	Identificador Único	Entero		0...*
nombreProyecto	Nombre del Proyecto	Carácter	20	a-z, A-z
tipoProyecto	Tipo del Proyecto	Carácter	20	a-z, A-z
nomUsuario	Nombre de Usuario	Carácter	20	a-z, A-z

### Localización Óptima

Atributo	Descripción	Tipo	Longitud	Rango
idLocalizacion	Identificador Único	Entero		0...*
nomPrima	Nombre de Materia Prima	Carácter	40	a-z, A-z
distanciaKm	Distancia en Kilómetros	Entero	10	0...9
distanciad	Distancia en Días	Entero	10	0...9

servPu1	Servicios Público 1	Carácter	20	a-z, A-z
servPu2	Servicio Público 2	Carácter	20	a-z, A-z
servPu3	Servicio Público 3	Carácter	20	a-z, A-z
zonaIndustrial	Zona Industrial	Carácter	2	Si, No
desToxico	Desecho Tóxico	Carácter	2	Si, No

### Tamaño Óptimo

Atributo	Descripción	Tipo	Longitud	Rango
idTamaño	Identificador Único	Entero		0...*
cantidadDemanda	Cantidad de Demanda	Carácter	20	0...9
cantidadProduccion	Cantidad a Producir	Carácter	20	0...9
cantidad*	Cantidad	Entero	10	0...9
produccionAño	Años a Producir	Entero	10	0...9

tipoPersonas	Tipo de Mercado Dirigido	Carácter	30	a-z, A-z
costoTerreno	Servicio Público 3	Entero	15	0...9
costoConstruccion	Zona Industrial	Entero	15	0...9
Long	Longitud de la Empresa	Entero	10	0...9
Sup	Superficie de la Empres	Entero	10	0...9
Vol	Volumen de la Empresa	Entero	10	0...9

### Ponderar

Atributo	Descripción	Tipo	Longitud	Rango
Vid	Identificador Único	Entero		0...*
vdistanciaKm	Valor de la Distancia en Km	Entero	2	0, 1
vdistanciaD	Valor de la Distancia en Días	Entero	2	0, 1

vservPu1	Valor del Servicios Público 1	Entero	2	0, 1
vservPu2	Valor del Servicio Público 2	Entero	2	0, 1
vservPu3	Valor del Servicio Público 3	Entero	2	0, 1
vzonaIndustrial	Valor de la Zona Industrial	Entero	2	0, 1
vdesToxico	Valor del Desecho Tóxico	Entero	2	0, 1

## 4.2 Diagrama Entidad Relación

A continuación mostraremos el diagrama general Entidad-Relación del problema que se abarca en la presente tesis y daremos pie a mencionar el tipo de relación que tiene las entidades mencionadas.

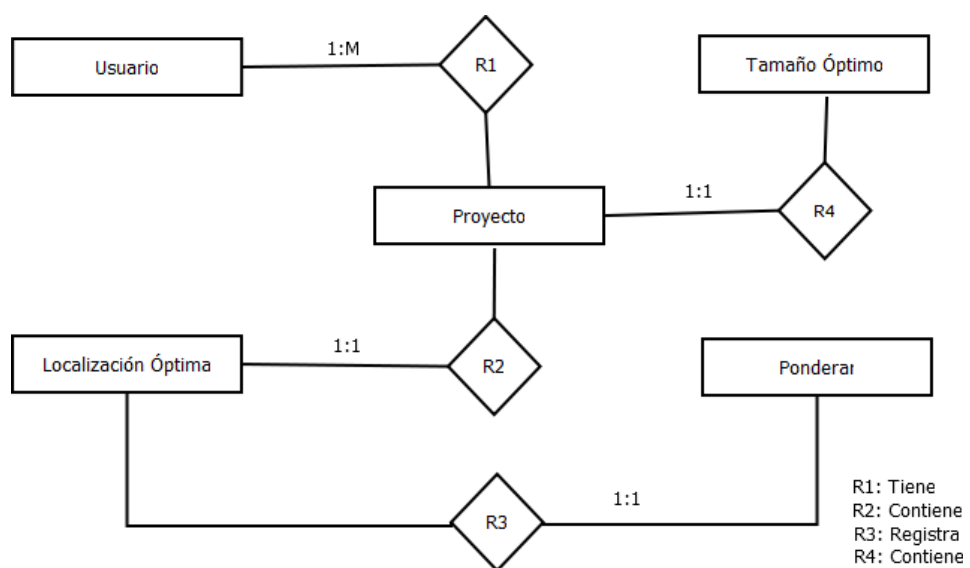


Fig 4 1 Diagrama de Entidad Relación

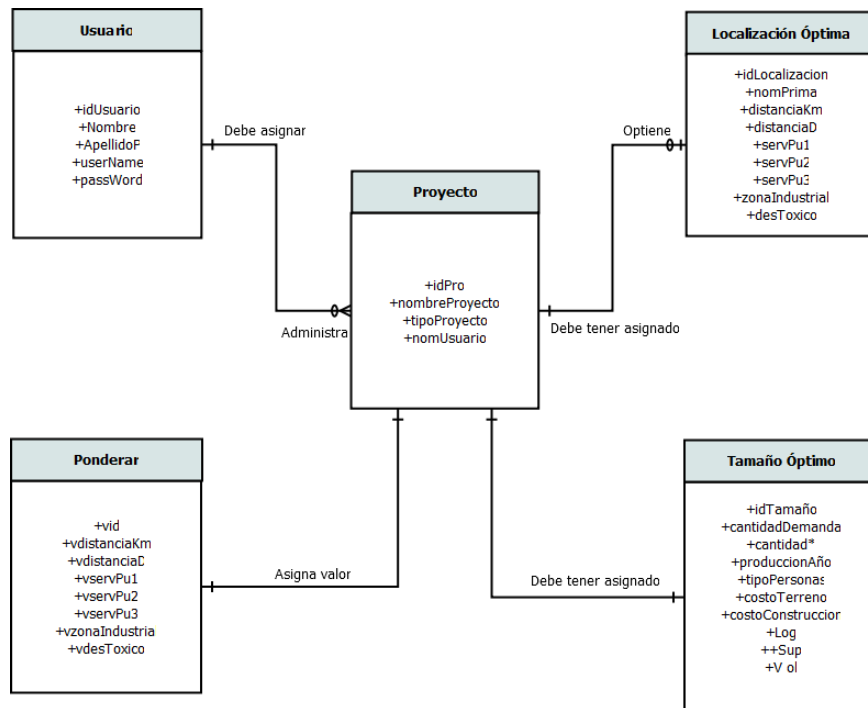


Fig 4.2 Diagrama de Entidad Relación con Día

#### 4.2.1 Relaciones

Las relaciones representan las reglas y la información que el negocio necesita. [4]

De las Entidades que tenemos detectadas de nuestro análisis de requerimientos, estableceremos las relaciones que existen entre estas y qué tipo de relación tienen establecida que a continuación describiremos:  
 Usuario con Proyecto

- Usuario en relación con Proyecto(idusuario, nomUsuario)

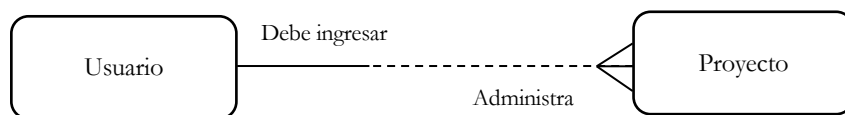


Fig. 4.2 Diagrama Relación Usuario-Proyecto



La entidad Usuario su relación es uno a muchos con la entidad Proyecto ya que un usuario debe ingresar uno o varios proyectos pero un Proyecto solo puede estar administrado por un solo usuario.

- Proyecto en relación con Localización Óptima  
 $\text{Óptima}(\text{idPro}, \text{idLocalizacion})$

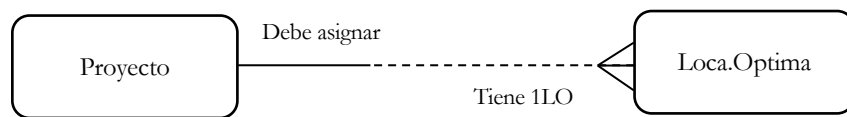


Fig. 4.3 Diagrama Relación Usuario-Localización Óptima

Lo anterior podemos apreciar que entidad Proyecto su relación es uno a uno con la entidad Localización Óptima ya que un Proyecto ingresado solo puede tener una respuesta de Localización Óptima.

- Proyecto con Tamaño Óptimo( $\text{idPro}, \text{idTamaño}$ )

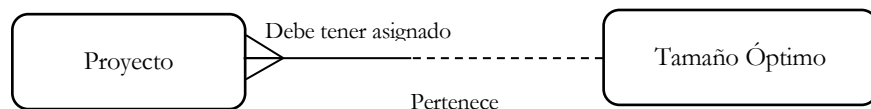


Fig. 4.4 Diagrama Relación Proyecto-Tamaño Óptimo

En este ejemplo también podemos apreciar que nuevamente proyecto está relacionado uno a uno con Tamaño Óptimo ya que un proyecto solo se le puede asignar un solo valor con respecto al Tamaño Óptimo.

- Proyecto en relación con Ponderar( $\text{idPro}, \text{vid}$ )

Cabe mencionar que la relación de la entidad Proyecto con Ponderar se omitió ya que es la misma que tiene con Localización Óptima toda vez que resulta que Ponderar es una entidad que guarda solo los resultados ingresados a la entidad Localización Óptima.

### 4.3 Normalización de la Base de Datos

Ahora que contamos con las entidades, sus atributos y los identificadores únicos definidos, procederemos a normalizar cada una de las entidades.

Se iniciará mostrando la dependencia funcional de los atributos de las entidades.

Usuarios

idUsuario → Nombre, apellidoP, userName, password.

Proyecto

idPro → nombreProyecto, tipoProyecto,  
nomUsuario.

Localización Óptima

idLocalización → nomPrima, distanciaKm, distanciaD,  
servPu1, servPu2, servPu3, zonaIndustrial,  
desToxico.

Tamaño Óptimo

idTamaño → cantidadDemanda, cantidadProduccion,  
cantidad\*, produccionAño, tipoPersonas,  
costoTerreno, costoConstruccion, Log, Sup,  
Vol.

Ponderar

vid  $\longrightarrow$  vdistanciaKm, vdistanciaD, vservPu1,  
vservPu2, vservPu3, vzonaIndustrial,  
vdesToxico.

Con lo anterior mostramos que todos los atributos de las entidades dependen funcionalmente de sus claves primarias.

Ahora analizaremos las entidades y realizaremos los procesos necesarios para llevarlas a la 3FN forma normal, de ser necesario.

#### **4.3.1 Primera Forma Normal (1FN):**

Una relación está en la primera forma normal (o 1FN) si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos [10].

Para verificar que las entidades cumplen con estar en primera forma normal, en el apartado diseño lógico se muestran dichas entidades como tablas con datos que cumplen con el requisito marcado en la primera forma normal.

#### **4.3.2 Segunda Forma Normal (2FN)**

Una relación pertenece a la segunda forma normal si es 1FN y cada atributo no-clave de la relación es total y funcionalmente dependiente de su clave principal [6].

Como observamos en la descripción de dependencia funcional, se nota como cada atributo no clave de cada entidad, es total y

funcionalmente dependiente de la clave principal, por lo que podemos asegurar que está en 2FN

### 4.3.3 Tercera Forma Normal (3FN)

Una relación es 3FN, si es 2FN y ningún atributo no-clave en la relación es funcionalmente dependiente de algún otro atributo no-clave [10].

Nuevamente al analizar la descripción de dependencia funcional podemos apreciar que no existen atributos no clave, que sean dependientes de otros no clave en las relaciones, permitiendo afirmar que se encuentran en 3FN.

## 4.4 Implementación de la Base de Datos

Para la parte de la implementación de la BD de nuestro sistema, hicimos uso del Sistema de Gestión de Base de Datos MySQL, el cual facilitó la creación de la BD, sus tablas y atributos, y proporcionó los mecanismos para la manipulación de los datos.

Es mediante la elaboración de un script de instrucciones MySQL, que nos proporcionan la base de datos con todas las características diseñadas en capítulos anteriores.

El script de implementación es el siguiente:

```
CREATE DATABASE EstudyTecni;
USE EstudyTecni;
DROP TABLE IF EXISTS Usuario;
CREATE TABLE Usuario (
idUsuario INT NOT NULL AUTO_INCREMENT PRIMARY
KEY,
Nombre VARCHAR(20),
apellidoP VARCHAR(20),
userName VARCHAR (20),
password VARCHAR (20));
```

```

DROP TABLE IF EXISTS Proyecto;
CREATE TABLE Proyecto (
idPro INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
nombreProyecto VARCHAR(20) NOT NULL,
tipoProyecto VARCHAR(20) NOT NULL,
nomUsuario INT (10) NOT NULL,
FOREIGN KEY(nomUsuario) REFERENCES
usuario(idUsuario));
DROP TABLE IF EXISTS LocalizacionOptima;
CREATE TABLE LocalizacionOptima (
idLocalizacion INT (10) NOT NULL PRIMARY KEY,
nomPrima VARCHAR(20) NOT NULL,
distanciaKm INT (10) NOT NULL,
distanciaD INT (10) NOT NULL,
servPu1 VARCHAR(20) NOT NULL,
servPu2 VARCHAR(20) NOT NULL,
servPu3 VARCHAR(20) NOT NULL,
zonaIndustrial VARCHAR(2) NOT NULL,
desToxico VARCHAR(2) NOT NULL,
FOREIGN KEY(idLozalicacion) REFERENCES proyecto(idpro));
DROP TABLE IF EXISTS TamañoOptimo;
CREATE TABLE TamañoOptimo (
idTamaño INT NOT NULL PRIMARY KEY,
cantidadDemanda VARCHAR(20) NOT NULL,
cantidadProduccion VARCHAR(20) NOT NULL,
cantidad INT(10) NOT NULL,
produccionAño INT(10) NOT NULL,
tipoPersona VARCHAR(30) NOT NULL
costoTerreno INT(15) NOT NULL,
costoConstruccion INT(15) NOT NULL,
Longd INT(10) NOT NULL,
Sup INT(10) NOT NULL,
Vol INT(10) NOT NULL,
FOREIGN KEY(idTamaño) REFERENCES proyecto(idpro));
DROP TABLE IF EXISTS Ponderar;
CREATE TABLE Ponderar (
vid INT NOT NULL PRIMARY KEY,
vdistanciaKm INT (2) NOT NULL,
vdistanciaD INT (2) NOT NULL,
vservPu1 INT(2) NOT NULL,
vservPu2 INT(2) NOT NULL,
vservPu3 INT(2) NOT NULL
vzonaIndustrial INT(2) NOT NULL,
vdesToxico INT(2) NOT NULL),
FOREIGN KEY(vid) REFERENCES proyecto(idpro));

```

## Capítulo 5

### IMPLEMENTACIÓN

Finalmente en este capítulo verificaremos los resultados obtenidos en el desarrollo de este proyecto, veremos la creación de la interfaz y los diferentes módulos creados para la comparación de zonas, para la obtención de un tamaño óptimo con respecto a su producción y de la verificación de resultados todo esto de la mano con el proceso de análisis que se precisó en los capítulos anteriores.

#### 5.1 Interfaz del Sistema

##### 5.1.1 Página de Inicio

A continuación veremos en las siguientes figuras la página de inicio de nuestro sistema y el sistema instalado en un emulador de IOS.



Fig 5.1 Pantalla de Inicio en IOS

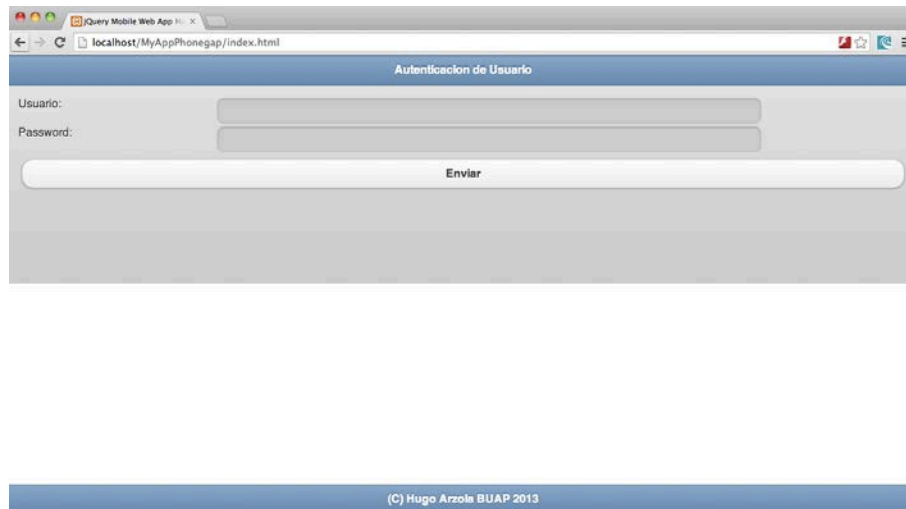


Fig 5.2 Pantalla de Inicio en Web

Como podemos apreciar nuestra interfaz fue desarrollada para adaptarse tanto en dispositivos móviles como en pantallas de un computador común, esta página está compuesta por un formulario simple con los campos nombre, password y un botón de acceso.

### 5.1.1 Página de Usuario

Si el usuario se encuentra registrado dentro de la Base de Datos e ingresa su usuario y contraseña correctamente podrá acceder a las funciones que ofrece el sistema.

Las siguientes imágenes muestran las diferentes funcionalidades del sistema.



Pantalla de instalación.- como pudimos apreciar en la imagen anterior a pesar de no haber usado Objective-C como lenguaje base para el desarrollo de esta

aplicación, de igual forma con los lenguajes usados podemos hacer una App con instalación nativa.

Fig 5.3 Pantalla de Instalación IOS



Fig 5.4 Pantalla de Bienvenida

Pantalla de bienvenida del usuario.- una vez ingresado correctamente a la aplicación, la misma envía una pantalla de bienvenida.





Pantalla de datos.- En esta pantalla es la pantalla principal ya que desde aquí el usuario puede acceder a la mayoría de las funcionalidades.



Fig 5.6 Pantalla de Localización

Pantalla de localización.- En esta pantalla el usuario ingresará los datos de la “zona 1” y la “zona 2” para posteriormente hacer el cálculo de comparación.

Pantalla de tamaño.-  
podrá ingresar los  
para después  
el tamaño



Fig 5.7 Pantalla de Tamaño

En esta sección el usuario  
datos en un formulario  
ponderar los datos y sacar  
óptimo.



Fig 5.8 Pantalla de Ponderar

Pantalla de ponderar.- En esta última sección veremos los cálculos y resultados que surgieron entre las comparaciones “zona 1” y “zona 1”.

## 5.2 Pruebas del Sistema

Ahora verificaremos las funcionalidades del sistema para ello aplicaremos una serie de pruebas que permitan verificar el buen funcionamiento del sistema y los módulos del mismo.

Las pruebas que se realizaron fueron:

De la caja blanca: Que son aquellas pruebas enfocadas a probar módulos del sistema a nivel de código, mediante la verificación de datos a través de los posibles caminos que puede tomar la información.

De la caja negra: Estas están orientadas a probar el sistema desde el punto de vista de la interfaz con los usuarios, integrando datos y verificando que el sistema realizando la funciones solicitadas.

### 5.2.1 Caja Blanca

Para realizar esta prueba elegimos un módulo específico del sistema, elaboramos nuestro caso de prueba y realizamos la verificación del mismo.

#### Caso de Prueba 1

*Módulo:* Verificar usuario

*Descripción:* Se ingresa un usuario o un password incorrecto, pero de igual forma puede no haber ingresado ningún dato.

*Resultado Esperado:* El sistema envía un letrero de “error”

*Código:*

#### HMTL 5

```
<script src="jquery-mobile/jquery.js"></script>
<!--Libreria de Phonegap-->
<!--Libreria JQueryMobile-->
<script src="jquery-mobile/jquery-1.5.min.js"></script>
<!--Libreria de Phonegap-->
<script src="/phonegap.js"></script>
<script src="lijs/login.js"></script>
<!--Libreria JQueryMobile-->
<script src="jquery-mobile/jquery.mobile-1.0a3.min.js"></script>
</head>
<body>
<div data-role="page"
```

```

id="inicio" data-theme="b" data-position="fixed">
  <div data-role="header">
    <h1>Autenticacion de Usuario</h1>
  </div>
<div data-role="content">
<form id="formulario">
  <label for="usuario">Usuario:</label>
    <input type="text" id="usuario" name="usuario">
    <label for="password">Password:</label>
    <input type="password" id="password" name="password">
    <input type="submit" value="Enviar" id="enviar">
  </form>
</div>
<div data-role="footer" data-theme="b" data-position="fixed">
<h4>(C) Hugo Arzola BUAP 2013</h4>
</div>
</div>

```

## JavaScript

```

$(document).ready(function() {
  $('#formulario').submit(function() {

    var datosUsuario = $("#usuario").val()
    var datosPassword = $("#password").val()

    archivoValidacion
    ="http://localhost/MyAppPhonegap/liphp/validacion.php?jsoncallback=?"

    $.getJSON(
      archivoValidacion, {usuario:datosUsuario
      ,password:datosPassword})
    .done(function(respuestaServer) {

      if(respuestaServer.validacion == "ok"){

        window.location.href = "pagsig.html";

      }else{

        alert("Error usuario o contraseña incorrecta");
      }
    })
    return false;
  })
})

```

## PHP

```
$usuario = mysql_real_escape_string($_GET['usuario']);
$password = mysql_real_escape_string($_GET['password']);
$consulta = "SELECT nombre,password FROM usuario WHERE
nombre='$usuario' AND password='$password'";
$resultado = mysql_query($consulta);
if (mysql_num_rows($resultado) > 0) {
$resultados["validacion"] = "ok";
} else {
$resultados["validacion"] = "error";
}
mysql_close($conexion);
$resultadosJson = json_encode($resultados);
echo $_GET['jsoncallback'] . '(' . $resultadosJson . ');';
```

Respuesta esperada: El sistema manda un mensaje de “usuario o password incorrecto”

## Caso de Prueba 2

*Módulo:* Insertar Proyecto

*Descripción:* Una vez ingresado al sistema el usuario podrá ingresar el nombre y tipo de producción.

*Resultado Esperado:* El sistema envía un letrero de “se guardo correctamente”

*Código:*

## HTML 5

```
<div data-role="content">
<form id="insertar">
<label>Nombre del proyecto:</label>
<input type="text" id="nombrepro" name="nombrepro">
<br>
<br>
<label>Escoja solo una opción</label>
<br>
<br>
<label for="manufacturaform">Manufactura</label>
<select id="manufacturaform" class="manufacturaform"
name="manufactura" data-native-menu="false" multiple>
<option data-placeholder="true">Selecciona tipo de
producción</option>
```

```

<option value="porProyecto">Por proyecto</option>
<option value="porOrdenesdeProduccion">Por órdenes de
producción</option>
<option value="porLotes">Por lotes</option>
<option value="enLinea">En línea</option>
<option value="continuos">Continuos</option>
</select>
<label for="servicios">Servicios</label>
<select id="servicios" class="servicios" name="servicios" data-native-
menu="false" multiple>
<option data-placeholder="true">Selecciona tipo de
producción</option>
<option value="servProfesionales">Servicios profesionales</option>
<option value="servMasivos">Servicios masivos</option>
<option value="servTalleres">Talleres de servicios</option>
</select>
<br>
<br>
<input type="submit" id="guardar" value="Guardar">
</form>

```

## JavaScript

```

$(document).ready(function() {
$('#insertar').submit(function() {
var datosProyecto = $("#nombrepro").val()
var datosManufactura = $(".manufacturaform option:selected").val()
var datosServicios = $(".servicios option:selected").val()
if (datosProyecto == "" || datosManufactura == "" || datosServicios
== "") {
alert("Faltan datos en el formulario");
} else {
archivoValidacion
="http://localhost/MyAppPhonegap/liphp/insertarpro.php?jsoncallback=?"
$.getJSON( archivoValidacion, { nombrepro:datosProyecto
,manufacturaenv:datosManufactura, servicios:datosServicios})
.done(function(respuesta) {
if(respuesta.validacion == "ok") {
alert("Se guardo correctamente");
} else {
alert("Error al guardar");
}
})
return false;
}
})
})

```

```
)
```

## PHP

```
$nombrepro = mysql_real_escape_string($_GET['nombrepro']);  
$manufacturaphp = mysql_real_escape_string($_GET['manufacturaenv']);  
$servicios = mysql_real_escape_string($_GET['servicios']);  
if (!$manufacturaphp) {  
$consulta = "INSERT INTO produccion (proyecto, tipoproduc)  
VALUES ('.$nombrepro.', '$servicios.')";  
}  
else {  
$consulta = "INSERT INTO produccion (proyecto, tipoproduc)  
VALUES ('.$nombrepro.', '$manufacturaphp.')";  
}  
$resultado = mysql_query($consulta);  
if ($resultado == true) {  
$resultados["validacion"] = "ok";  
}  
else {  
$resultados["validacion"] = "error";  
}  
mysql_close($conexion);  
$resultadosJson = json_encode($resultados);  
echo $_GET['jsoncallback'] . '(' . $resultadosJson . ')';
```

Respuesta esperada: El sistema manda un mensaje de “se guardo correctamente”.

### 5.2.1 Caja Negra

Para realizar esta prueba seleccionamos la funcionalidad a probar, elaboramos el caso de prueba correspondiente y realizamos la verificación de los resultados obtenidos.

#### Caso de Prueba 1

*Módulo:* Verificar usuario

*Descripción:* Se ingresa un usuario o un password incorrecto, pero de igual forma puede no haber ingresado ningún dato.

*Resultado Esperado:* El sistema envía un letrero de “error”

*Pantallas:*



**Caso de Prueba 2**

*Módulo:*

Fig 5.9 Pantalla de Error con campos vacíos y campos llenos

**Insertar Proyecto**

*Descripción:* Una vez ingresado al sistema el usuario podrá ingresar el nombre y tipo de producción.

*Resultado Esperado:* El sistema envía un letrero de “se guardo correctamente”

*Pantallas:*





Fig 5.10 Pantalla de datos con selección de manufactura y servicios

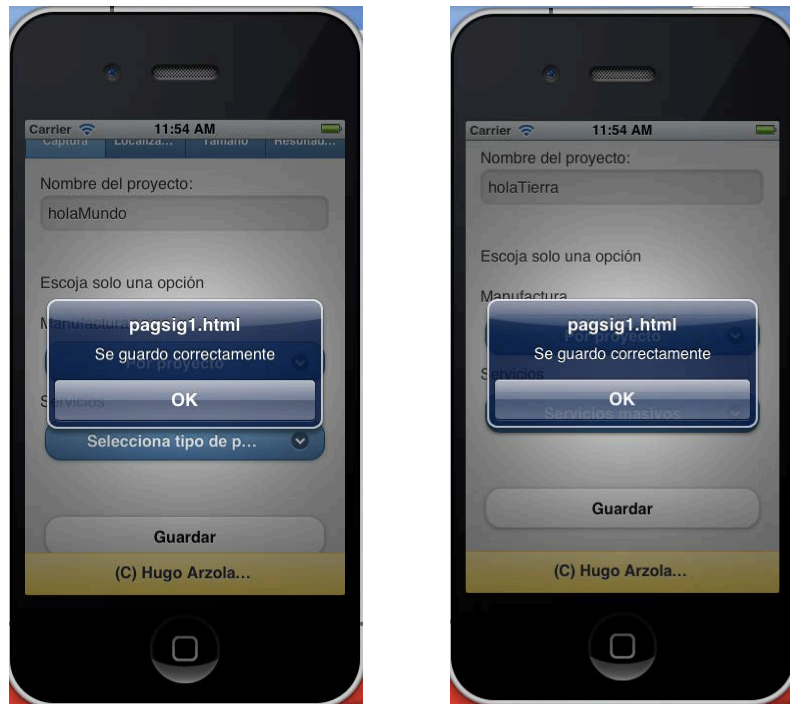


Fig 5.11 Pantalla de validación en datos con manufactura y servicios

## CONCLUSIONES

En el proyecto presentado nos proporciono un panorama del uso de las nuevas tecnologías asociadas o con tendencia hacia dispositivos móviles y la facilidad con que se puede reutilizar el código para distintas plataformas permitiéndonos ampliar el área de impacto en el sistema.

La problemática presenta en este proyecto se abarco desde la metodología UML que permitió la precisa organización de las tareas para el bueno funcionamiento y desarrollo del sistema, obteniendo un software de más calidad y facilidad para su mantenimiento.

El uso de la aplicación de la teoría y diseño de la BD, proporciono procesos adecuado para la elaboración adecuada de las base de datos evitando redundancia y cuidando la integridad de la información.

Las plataformas usadas más conocidas como MySQL, PHP y XAMP nos proporciono una serie de herramientas de uso común que facilitan el mantenimiento y aprendizaje de las mismas, cabe señalar que es vital importancia el tiempo y el esfuerzo que se empleara en el desarrollo de un sistema, por ello el uso de plataformas conocidas y adecuadas simplifican un buen desarrollo del mismo.

De igual forma el uso de herramientas como: JQueryMobile, HTML5, CSS3 y Phonegap nos facilitó la implementación de sistema en dispositivos móviles y además proporcionándonos herramientas simples para abarcar problemas complejos como el visto en este trabajo.

Para finalizar el emplear tecnologías de bajo costo en su desarrollo, permitirá implementar sistemas con un margen menor de costo de tiempo e inversión, reutilizando el código y con pequeños ajuste implementándolo en diferentes plataformas.

## PERSPECTIVAS

El presente proyecto no abarco toda la problemática que representa un estudio técnico, como trabajo a futuro se puede implementar los módulos de administración de personal, administración de maquinaria así como contemplar los procesos legales que deben ser contemplados en la elaboración e instalación de cualquier empresa.

Muchas características que ofrece un dispositivo móvil se omitieron ya que no fueron necesarias para dar solución a la problemática de este trabajo, es por ello que el sistema podrá crecer en funcionalidades para dar una mejor experiencia al usuario.

La interfaz de usuario es susceptible a ser mejorada agregando aspectos multimedia que desarrollen un aspecto más atractivo para el usuario.

Así mismo el sistema también puede ser tomado como base para la implementación de otros tipos de estudios técnicos de diferentes fines, con la adaptación de las funcionalidades respectivas.

## BIBLIOGRAFÍA

- [1] Microsoft. <http://www.microsoft.com/es-xl/dynamics/erp.aspx>  
Microsoft, 2013.
- [2] kpiERP. <http://www.kpierp.com.com/aboutus/>
- [3] CSS 3 [http://es.wikipedia.org/wiki/Hojas\\_de\\_estilo\\_en\\_cascada](http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada)
- [4] Yolanda Moyao, Notas del Diplomado Modulo II, Analisis y Diseño de Bases de Datos, Facultad de Ciencias de la Computación, BUAP, 2011.
- [5] Roberto Rodriguez Echeverría, Álvaro Prieto Ramos y Encarna Sosa Sánchez, Programación Orientada a Objetos, Creative Commons, 2004.
- [6] Miguel Rodríguez, Notas del Diplomado Modulo I, Analisis y Diseño Orientado a Objetos, Facultad de Ciencias de la Computación, BUAP, 2011.
- [7] Wikipedia.  
[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado), 2013.
- [8] Joseph Schmuller, UML en 24 Horas, Prentice Hall, 2001.
- [9] Wikipedia. [http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos), 2013.
- [10] Abraham Silberschatz, Fundamentos de Bases de Datos, McGRAW-HILL/INTERAMERICANA DE ESPAÑA, 4ta Edición, 2002.
- [11] Wikipedia.  
[http://es.wikipedia.org/wiki/Normalización\\_de\\_bases\\_de\\_datos](http://es.wikipedia.org/wiki/Normalización_de_bases_de_datos)
- [12] Wikipedia. <http://es.wikipedia.org/wiki/XAMPP>
- [13] Wikipedia. [http://es.wikipedia.org/wiki/OS\\_X](http://es.wikipedia.org/wiki/OS_X)

- [14] Wikipedia. [http://es.wikipedia.org/wiki/Mac\\_OS\\_X\\_v10.6](http://es.wikipedia.org/wiki/Mac_OS_X_v10.6)
- [15] Apple. <https://developer.apple.com/technologies/tools/whats-new.html>
- [16] Google. <http://developer.android.com/sdk/index.html>
- [17] Phonegap. <http://phonegap.com/>
- [18] Adobe. <http://kb2.adobe.com/es/products/dreamweaver.html>
- [19] MySQL. <http://dev.mysql.com/doc/refman/5.0/es/features.html>
- [20] PHP. <https://wiki.php.net/>
- [21] HTML 5. <http://www.w3.org/html/logo/>
- [22] JavaScript. <http://www.w3.org/html/logo/>
- [23] Miguel Vega, Presentación Casos de Uso UML, LSI-UGR, 2010.
- [24] JQueryMobile <http://jquerymobile.com/>

