Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

Licenciatura en Sistemas Computacionales

**Desarrollo de software para control de inventarios.**

**Caso de Estudio: Tiendas de celulares “Celulares Castillo”.**

**P R E S E N T A**

Christian Yamil Castillo Covarrubias.

**Asesores del trabajo:**

L. S. C. Edgar Calderón Toledo.

M. C. C. Eduardo Cornejo Velázquez.

Mineral de la Reforma, Hidalgo, abril de 2014

México

**Resumen**

El desarrollo de un sistema de información para cualquier pequeña empresa que está creciendo y distribuyéndose poco a poco es de suma importancia, pues tener control sobre las sucursales llega a convertirse en una tarea un tanto complicada a medida que pasa el tiempo. Se ha seleccionado la pequeña empresa “Celulares Castillo” como caso de estudio.

El presente trabajo contiene una propuesta para diseñar e implementar un par de sistemas de información para la pequeña empresa “Celulares Castillo” que tiene cuatro sucursales distribuidas en cuatro municipios del estado de Hidalgo. Este trabajo contiene de igual manera el diagrama entidad-relación y los diagramas relacionales de los sistemas que se diseñarán e implementarán.

El objetivo principal de los sistemas de información a desarrollar es ayudar a tener un mejor control sobre los equipos celulares en reparación, garantías y ventas, así como también brindar un catálogo de clientes en garantía, reparación y ventas.

**Índice**

[Introducción 1](#_Toc387583210)

[Antecedentes 2](#_Toc387583211)

[Planteamiento del problema. 2](#_Toc387583212)

[Propuesta de solución. 3](#_Toc387583213)

[Justificación 4](#_Toc387583214)

[Objetivos 5](#_Toc387583215)

[General 5](#_Toc387583216)

[Específicos 5](#_Toc387583217)

[Metas 5](#_Toc387583218)

[Alcances y limitaciones 6](#_Toc387583219)

[Alcances 6](#_Toc387583220)

[Limitaciones 6](#_Toc387583221)

[Capítulo 1: Marco teórico 7](#_Toc387583222)

[1.1 Sistema 7](#_Toc387583223)

[1.2 Dato 7](#_Toc387583224)

[1.3 Información 7](#_Toc387583225)

[1.4 Sistema de Información 8](#_Toc387583226)

[1.5 Base de datos 8](#_Toc387583227)

[1.6 Framework 8](#_Toc387583228)

[1.7 Modelo Vista Controlador 8](#_Toc387583229)

[1.8 Entorno de Desarrollo Integrado 8](#_Toc387583230)

[1.9 UML ® 9](#_Toc387583231)

[1.9.1 Estereotipo de UML. 9](#_Toc387583232)

[1.9.2 Tipos de diagramas en UML 9](#_Toc387583233)

[Capítulo 2: Metodología 10](#_Toc387583234)

[2.1 Características, ventajas, desventajas y propiedades de la metodología 11](#_Toc387583235)

[2.1.1 Características 11](#_Toc387583236)

[2.1.2 Ventajas 11](#_Toc387583237)

[2.1.3 Desventajas 11](#_Toc387583238)

[2.1.4 Propiedades 11](#_Toc387583239)

[2.2 Aplicación de la metodología 12](#_Toc387583240)

[2.2.1 Planificación de requerimientos 12](#_Toc387583244)

[2.2.2 Diseño 12](#_Toc387583245)

[2.2.3 Desarrollo 12](#_Toc387583246)

[2.2.4 Implementación 12](#_Toc387583247)

[Capítulo 3: Estado del arte 13](#_Toc387583248)

[3.1 Groovy & Grails 13](#_Toc387583249)

[“Desarrollo Ágil de Aplicaciones Web con Grails Framework. Caso de estudio: PROMEP­UAEH.” 13](#_Toc387583250)

[3.2 Microsoft Visual Basic y SQL Server 13](#_Toc387583251)

[“MOBILE STORE MANAGEMENT SYSTEM” 13](#_Toc387583252)

[“MOBILE SHOP MANAGEMENT SYSTEM SOFTWARE” 13](#_Toc387583253)

[3.3 MySQL 14](#_Toc387583254)

[“Development of a SaaS Inventory Management System” 14](#_Toc387583255)

[3.4 Java 15](#_Toc387583256)

[“Mobile Shop Management System” 15](#_Toc387583257)

[3.5 Conclusiones 15](#_Toc387583258)

[Capítulo 4: Herramientas computacionales 16](#_Toc387583259)

[4.1 Embarcadero Delphi XE ® 16](#_Toc387583260)

[Características 16](#_Toc387583261)

[Ventajas 16](#_Toc387583262)

[Desventajas 16](#_Toc387583263)

[4.2 Oracle MySQL ® 17](#_Toc387583264)

[Características 17](#_Toc387583265)

[Ventajas 17](#_Toc387583266)

[Desventajas 17](#_Toc387583267)

[4.3 Groovy 18](#_Toc387583268)

[Características 18](#_Toc387583269)

[Ventajas 18](#_Toc387583270)

[Desventajas 18](#_Toc387583271)

[4.4 Grails 19](#_Toc387583272)

[Características 19](#_Toc387583273)

[Ventajas 19](#_Toc387583274)

[Desventajas 19](#_Toc387583275)

[4.5 Twitter Bootstrap 19](#_Toc387583276)

[Características 19](#_Toc387583277)

[Ventajas 19](#_Toc387583278)

[Desventajas 19](#_Toc387583279)

[Capítulo 5: Diseño de la base de datos 20](#_Toc387583280)

[5.1 Diagrama entidad-relación 20](#_Toc387583281)

[5.1.1 Diagrama E-R para sistema de empleados 21](#_Toc387583282)

[5.1.2 Diagrama E-R para sistema de administradores 22](#_Toc387583283)

[5.1 Diagrama relacional 23](#_Toc387583284)

[5.2 Entidades 24](#_Toc387583285)

[Capítulo 6: Diseño del sistema 31](#_Toc387583286)

[Diagramas de casos de uso 31](#_Toc387583287)

[Sistema de empleados 31](#_Toc387583288)

[Venta de celular 31](#_Toc387583289)

[Recepción de equipos en reparación 32](#_Toc387583290)

[Recepción de equipos en garantía 32](#_Toc387583291)

[Venta de recargas 33](#_Toc387583292)

[Sistema de administradores 33](#_Toc387583293)

[Alta de equipos para inventarios 33](#_Toc387583294)

[Revisión de movimientos del usuario 34](#_Toc387583295)

[Consulta de equipos entregados 34](#_Toc387583296)

[Consulta de equipos recibidos 34](#_Toc387583297)

[Diagramas de flujo de datos 36](#_Toc387583298)

[Sistema para empleados 36](#_Toc387583299)

[Acceso al sistema de empleados 36](#_Toc387583300)

[Conclusiones parciales 40](#_Toc387583301)

[Anexo A: Cronograma de actividades 41](#_Toc387583302)

[Anexo B: Ubicación de las sucursales 45](#_Toc387583303)

[Anexo C: Manual de usuario “Sistema para empleados” 46](#_Toc387583304)

[Referencias 47](#_Toc387583305)

**Índice de Figuras**

**Capítulo 1**

[Figura 1. 1: Proceso de transformación de datos a información. 6](#_Toc381252624)

**Capítulo 2**

[Figura 2. 1: Ciclo de vida RAD. 9](#_Toc381486193)

**Capítulo 4**

[Figura 4. 1: Logotipo del IDE Embarcadero Delphi XE. 14](#_Toc384845743)

[Figura 4. 2: Logotipo de Oracle MySQL ®. 15](#_Toc384845744)

[Figura 4. 3: Logotipo del lenguaje de programación Groovy. 16](#_Toc384845745)

[Figura 4. 4: Logotipo del framework Grails. 16](#_Toc384845746)

**Capítulo 6**

[Figura 6. 1: Diagrama de casos de uso para la venta de un celular. 25](#_Toc384845730)

[Figura 6. 2: Diagrama de casos de uso para la recepción de equipos en reparación. 25](#_Toc384845731)

[Figura 6. 3: Diagrama de casos de uso para recepción de equipos en garantía. 26](#_Toc384845732)

[Figura 6. 4: Diagrama de casos de uso para venta de recargas. 26](#_Toc384845733)

[Figura 6. 5: Diagrama de casos de uso para alta de equipos en inventarios. 26](#_Toc384845734)

[Figura 6. 6: Diagrama de casos de uso para revisión de movimientos de usuarios. 27](#_Toc384845735)

[Figura 6. 7: Diagrama de casos de uso para consulta de equipos entregados. 27](#_Toc384845736)

[Figura 6. 8: Diagrama de casos de uso para consulta de equipos recibidos. 27](#_Toc384845737)

[Figura 6. 9: Diagrama de flujo para el acceso al sistema de empleados. 28](#_Toc384845738)

[Figura 6. 10: Diagrama de flujo de venta de un celular. 28](#_Toc384845739)

[Figura 6. 11: Diagrama de flujo de asignación de un código de recarga. 29](#_Toc384845740)

[Figura 6. 12: Diagrama de flujo de recepcion y entrega garantía. 29](#_Toc384845741)

[Figura 6. 13: Diagrama de flujo de recepción y entrega reparación. 30](#_Toc384845742)

**Índice de tablas**

**Capítulo 5**

[Tabla 5.2. 1: Entidad "cliente" 20](#_Toc384845715)

[Tabla 5.2. 2: Entidad "venta\_detalle" 20](#_Toc384845716)

[Tabla 5.2. 3: Entidad "empleado" 21](#_Toc384845717)

[Tabla 5.2. 4: Entidad "cliente\_garantía" 21](#_Toc384845718)

[Tabla 5.2. 5: Entidad "cliente\_reparacion" 21](#_Toc384845719)

[Tabla 5.2. 6: Entidad "codigo\_recarga" 22](#_Toc384845720)

[Tabla 5.2. 7: Entidad "equipo\_entregado" 22](#_Toc384845721)

[Tabla 5.2. 8: Entidad "equipo\_garantia" 22](#_Toc384845722)

[Tabla 5.2. 9: Entidad "equipo\_reparacion" 22](#_Toc384845723)

[Tabla 5.2. 10: Entidad "municipio" 23](#_Toc384845724)

[Tabla 5.2. 11: Entidad "sucursal" 23](#_Toc384845725)

[Tabla 5.2. 12: Entidad "nivel\_usuario" 23](#_Toc384845726)

[Tabla 5.2. 13: Entidad "equipo\_almacen" 23](#_Toc384845727)

[Tabla 5.2. 14: Entidad "empleado\_movimiento" 24](#_Toc384845728)

[Tabla 5.2. 15: Entidad "administrador\_movimiento" 24](#_Toc384845729)

# Introducción

El presente trabajo expone de una manera sencilla una propuesta para que la cadena de tienda de celulares “Celulares Castillo” logre llevar un mejor control sobre los inventarios existentes en cada una de sus sucursales. g

Con una lectura sencilla, se describe la propuesta de desarrollo del sistema para empleados y para administradores para la cadena de tiendas de celulares “Celulares Castillo”. Pretendiendo con esto último la optimización de los procesos de alta y consulta de equipos celulares en garantía, reparación y vendidos, clientes en garantía, en reparación y clientes de ventas.

En el capítulo 1 se dan a conocer las definiciones de algunos conceptos que son necesarios saber para poder comprender la lectura del presente documento. Desde la definición de que es un sistema hasta la definición de un *framework*; herramientas que se ocuparán a lo largo del desarrollo del sistema para administradores y las cuales se describirán en el capítulo 4.

En el capítulo 2 se define lo que es una metodología, así como también se ilustra y explica la metodología que se utilizará para llevar a cabo el desarrollo y mantenimiento del software para empleados y para administradores. En este capítulo se explica de igual forma la manera en que se aplicará la metodología.

Posteriormente, en el capítulo 3 se exhiben algunos trabajos de tesis similares al presente, así como también algunos productos ya comerciales a los cuales se les realizó un análisis y críticas constructivas.

El capítulo 4 define las herramientas computacionales que serán utilizadas a lo largo del desarrollo del sistema para empleados y del sistema para administradores. Dentro de este capítulo se especifican las ventajas, desventajas y principales características de cada una de las herramientas computacionales a utilizar, así como una breve descripción de cada una de estas.

Dentro del capítulo 5 se define el modelado de datos de los sistemas a desarrollar, así como también se definen conceptos básicos y necesarios para poder comprender lo que significa una base de datos. Se pueden encontrar conceptos como entidad, atributo, diagrama de entidad-relación y entidad relacional. Acto seguido, se encuentran y describen las entidades de la base de datos con la que interactuarán los sistemas a desarrollar.

Finalmente, en el capítulo 6 se encuentra el modelado de los sistemas de información a desarrollar, así como también es posible encontrar una definición sobre los diagramas que se utilizaron para analizar ambos sistemas de información. Dentro de este capítulo se pueden encontrar los diagramas de casos de uso para el sistema de administradores y empleados, así como una breve descripción sobre el posible flujo de trabajo para cada uno de los diagramas de casos de uso mostrados.

Acto seguido, se muestran los diagramas de flujo, así como una breve descripción para cada uno de estos, explicando el flujo de datos para cada diagrama. Cabe mencionar que en esta sección se pueden encontrar los diagramas de flujo para ambos sistemas de información a implementar.

# Antecedentes

Los pequeños negocios que con el tiempo comienzan a crecer, eventualmente enfrentan un problema: El acceso a la información de otra sucursal comienza a ser complicado, así mismo, su control se torna un poco más complejo y a su vez requiere más tiempo por parte de los administradores. Hojas de cálculo, llamadas a otras sucursales correspondientes e innumerables mensajes de texto llegan a ser insumos que los dueños de las sucursales tienen que utilizar día a día para poder llevar un control y estar al tanto de lo que pasa diariamente en sus sucursales.

En la tienda de celulares “Celulares Castillo”, el proceso para poder ingresar equipos a garantía o reparación consiste en llenar unas pequeñas hojas de las cuales una copia se la queda el cliente, una el vendedor y la original es para el administrador. Tal como lo ilustra la figura 1, estas hojas son pequeñas, frágiles y fácilmente pueden llegar a extraviarse lo cual puede llegar a causar conflictos son los clientes o empleados.



Figura : Formatos para celulares en garantía o en reparación.

Para entregar equipos a garantía o reparación, el cliente debe de entregar la pequeña hoja que se le entregó anteriormente, en caso de no contar con ella, basta con que el cliente indique el modelo del equipo o bien la terminación (los últimos 4 dígitos) del IMEI del celular en reparación o garantía.

# Planteamiento del problema.

La dueña de las tiendas de celulares “Celulares ‘Castillo’” enfrenta un problema que las pequeñas empresas que van creciendo eventualmente tienen que enfrentar: El hacer seguimiento a sus productos, en este caso, a teléfonos celulares: Alta y baja de equipos celulares, control de almacén, seguimiento de equipos en reparación y garantía, son algunos problemas que se enfrentan.

Al enfrentar este problema, no solamente se tiene que invertir en llamadas a celulares sino también realizar hojas de cálculo para poder llevar a cabo la contabilidad y llevar un pequeño control de equipos en inventario, entre otras cosas.

Pese a que se lleva un control local, a veces se llegan a perder los comprobantes que se utilizan para reclamar equipos celulares ya sea en garantía o en reparación. Un problema menor, pero no menos significativo, es estar revisando periódicamente el saldo que se tiene disponible para recargas en cada una de las tiendas de celulares.

# Propuesta de solución.

Para poder brindar una solución al problema, se ha propuesto desarrollar un sistema informático para empleados para que los mismos puedan dar de alta equipos, venta de equipos, verificar el estado de un equipo y puedan determinar si un celular se encuentra en garantía o en reparación. De igual forma, para la dueña de la tienda de celulares “Celulares Castillo” se desarrollará un sistema informático basado en web para que desde cualquier dispositivo pueda acceder a este y poder ver los reportes, equipos en almacén, equipos en garantía o reparación que se tengan en cada una de las sucursales en cualquier momento.

Para que el acceso a la información de las distintas sucursales sea óptimo, se propone implementar una base de datos que esté replicada. Tal como lo ilustra la figura 2, en cada sucursal se instalará un Sistema Gestor de Bases de Datos (SGBD) que permita replicación tipo “maestro-esclavo”. Se contará con un servidor dedicado (servidor maestro) y 4 servidores esclavo, mismos que permitirán que exista una interacción amena con los sistemas de información a desarrollar.

Se buscará un SGBD que se adapte mejor a las necesidades y requerimientos de la persona que hará uso del sistema para administradores. Los datos almacenados día a día en las tiendas estarán disponibles los 365 días del año, durante cualquier momento del día, pues si en algún momento del día la administradora de las tiendas de celulares “Celulares Castillo” decide acceder al sistema web desde su dispositivo móvil, el sistema tendrá que estar disponible y operar amenamente.

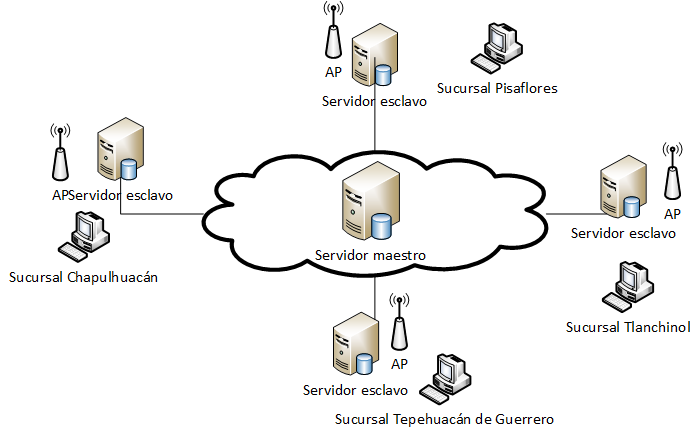


Figura : Diagrama de propuesta de solución.

# Justificación

La importancia de brindar una solución eficiente a este problema garantizará un ahorro importante de tiempo e insumos, tales como: saldo en celulares y llamadas que indudablemente pueden ser no necesarios a partir que el sistema comience a ser totalmente funcional. El sistema a implementar no solo solucionará éstos problemas, si no que brindará una solución a problemas externos dentro de las tiendas de celulares, tal como el haber perdido la nota de garantía o de equipo en reparación.

Para desarrollar el sistema que utilizarán los empleados, se utilizará el IDE Delphi XE en conjunto con el lenguaje de programación Object Pascal se logrará desarrollar de forma rápida el sistema de información que será ocupado por los empleados, pues una de las ventajas más notorias del lenguaje es la facilidad en su sintaxis, así como la notoria potencia del mismo. Así mismo, en caso de que se requieran modificaciones o actualizaciones de este sistema, será posible realizarlas en un tiempo corto.

Para desarrollar el sistema que será utilizado por los administradores, se utilizará el framework Grails en conjunto con el lenguaje de programación Groovy (que se basa en Java) se alcanzará una alta productividad y estabilidad.

Gracias al Modelo Vista Controlador (MVC) que implementa el framework Grails, será posible codificar el comportamiento del sistema web y las acciones posibles que el usuario realice, así como establecer una interfaz gráfica amigable y transparente para el usuario.

Para desarrollar la base de datos que será centralizada. Para ello, se utilizará el Sistema Gestor de Bases de Datos Relacionales (SGBDR) MySQL server, pues al tener licencia GNU GPL no tendrá costo para ser adquirido. A su vez, éste potente SGBDR garantizará el acceso al catálogo de las demás sucursales sin problemas, en cualquier momento.

Se propone una solución fiable, de tal forma que los administradores puedan acceder al sistema utilizando ya sea una computadora portátil o bien desde sus celulares o tablets. El conjunto de tecnologías implementado en éste Framework, agilizará el desarrollo del sistema para los administradores.

# Objetivos

## General

Desarrollar un sistema de control de inventarios robusto, escalable e intuitivo para que este sea utilizado por los empleados o bien por la administradora de la tienda de celulares y que pueda resolver y automatizar tareas que podrían estar propensas a fallar o completarse con poco éxito. Mediante la utilización y aplicación de herramientas computacionales propuestas en este trabajo de tesis, el sistema propuesto logrará llevarse a cabo y ser desarrollado en un tiempo rápido.

¿Qué vas a hacer?

¿Para que?

¿Mediante que?

## Específicos

* Implementar una interfaz amigable tanto para empleados como para administradores.
* Hacer que el sistema sea escalable.
* Desarrollar el sistema informático de empleados en aproximadamente 7 meses.
* Desarrollar el sistema para el administrador en menos de 6 meses.
* Implementar el sistema completo en un año.
* Reducir el gasto en llamadas a celulares.
* Utilizar el IDE Delphi XE y el framework Grails en conjunto para desarrollar un sistema robusto y estable.

# Metas

* Terminar ambos sistemas en menos de seis meses.
* Reducir el número de quejas cada año después de la implementación de ambos sistemas de información.
* Programar un “actualizador” de versiones para software de plataforma de escritorio.
* Desarrollar e implementar una sección de ayuda dentro del programa de escritorio.
* Desarrollar un sistema robusto e intuitivo.
* Implementar un apartado dentro del sistema para reporte de errores.

# Alcances y limitaciones

El sistema de información para empleados y para administradores tendrá algunos alcances y limitaciones. A continuación se mencionan los alcances y limitaciones que se pretende tenga este proyecto.

## Alcances

El sistema para empleados tendrá un control sobre:

* Garantías.
* Reparaciones.
* Equipos en almacén.

Se generarán reportes sobre:

* Garantías.
* Reparaciones.
* Equipos en almacén.
* El administrador podrá asignar un sueldo base a los empleados.
* El administrador podrá seguir un movimiento de usuarios por cada sucursal.
* Los empleados podrán ganar un pago extra cada ciertos equipos vendidos, o bien por cada equipo. Los administradores podrán elegir esto.
* El administrador podrá ver los equipos más vendidos y menos vendidos por sucursal.
* Los empleados pueden generar archivos en Microsoft Excel con el listado de ventas.
* Los empleados pueden generar archivos en Microsoft Excel con los celulares en inventario.
* El sistema para empleados tendrá un pequeño módulo para el administrador.

## Limitaciones

* El sistema para empleados correrá sobre Windows 7 o Windows 8, independientemente si su arquitectura es de 32 bits o 64 bits.
* El sistema para administradores estará montado sobre un servidor.

Capítulo 1: Marco teórico

**NOTA: AGREGAR REFERENCIAS POR TODOS LOS CONCEPTOS UTILIZADOS**

En este capítulo se presentan los conceptos que servirán como base para el desarrollo del sistema de control de inventarios. Además, se identifican las herramientas y técnicas que serán utilizadas a lo largo de este proyecto.

## Sistema

Un sistema es un conjunto de elementos que interactúan los unos con los otros para alcanzar un objetivo en común. Los sistemas están compuestos por entradas, procesamientos, mecanismos, salidas y retroalimentación. Los sistemas se clasifican en:

* **Sistema abierto:** Sistema que interactúa con su entorno.
* **Sistema cerrado:** No interactúa con su entorno.

**Sistema simple:** Consta de pocos elementos. La interacción con los mismos es sencilla.

* **Sistema complejo:** Contiene muchos elementos, mismos que están relacionados entre sí de forma estrecha.
* **Sistema estable:** Este tipo de sistemas no se ven tan afectados con el paso del tiempo.
* **Sistema adaptable:** Este tipo de sistemas son capaces de modificarse, adaptarse a cambios en el entorno.
* **Sistema no adaptable:** No se modifica en respuesta al entorno.

## Dato

Un dato es un símbolo o carácter que representa un hecho real.

## Información

Conjunto de datos procesados que presentan un valor mayor al de su estado original.

**De dato a información**

Los datos, por si solos no tienen casi ningún valor. Para que éstos sean útiles dentro de cualquier organización, deben de ser presentados y procesados de forma adecuada.

Para convertir datos en información se llevan a cabo una serie de tareas relacionadas entre sí, ejecutadas para producir un resultado definido. El proceso para definir relaciones entre datos requiere de conocimientos. Éste proceso es el que muestra la figura 1.1

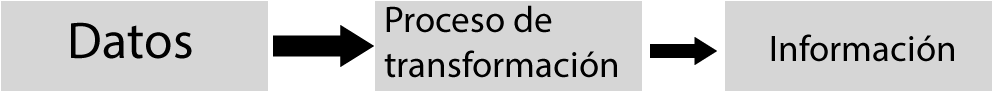


Figura 1. : Proceso de transformación de datos a información.

## Sistema de Información

Un sistema de información (SI) es un conjunto de elementos dedicados a recolectar, manipular, almacenar y distribuir datos e información.

Pese a que un sistema de información puede ser manual o automatizado, es común que en la actualidad estos sean automatizados.

Un sistema de información automatizado no garantizará un mejor desempeño, pues si el SI es defectuoso, se logrará aumentar el impacto de los errores.

## Base de datos

Una base de datos consiste en un conjunto de datos organizados e interrelacionados entre sí para un posible uso posterior, conteniendo información importante para cualquier empresa. Los datos almacenados en la base de datos son utilizados dentro de un sistema de información, donde éste las procesa y la convierte en información útil e interesante para el usuario final.

## Framework

“En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.” (Wikipedia, 2013)

## Modelo Vista Controlador

El modelo vista controlador (MVC) es un patrón de diseño de software que separa los datos y lógica de una aplicación de la interfaz gráfica del usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos, que son: El modelo, la vista y el controlador.

Esto quiere decir que se definen componentes para la interfaz grafica, y por otro lado se definen componentes para la interacción con el usuario. (Wikipedia, 2013)

## Entorno de Desarrollo Integrado

Software que contiene un conjunto de herramientas de programación que puede dedicarse a un solo lenguaje de programación o bien, puede utilizarse para varios.

Los IDE brindan un marco de trabajo ameno para la mayoría de los lenguajes de programación como C++, PHP, Java, Python, Delphi, etc. Un IDE, en algunos casos, puede funcionar también como un sistema en tiempo de ejecución, en donde se permite usar el lenguaje de programación de forma interactiva, como es el caso de Smalltalk u Objective-c.

## UML ®

*Unified Modeling Language*, por sus siglas en inglés, UML es un lenguaje de modelado de sistemas que es usado para el desarrollo de software orientado a objetos.

Para organizar el código del programa, los programadores suelen crear “objetos” que son un conjunto de datos dentro de programas. UML, el cual ha sido estandarizado por la *Object Managmente Group* (OMG), fue diseñado para éste propósito. El lenguaje ha ganado suficiente soporte, que ha convertido en un lenguaje estándar para la visualización y construcción de software.

Con UML se puede modelar cualquier tipo de aplicación, corriendo en cualquier tipo y cualquier combinación de software, sistema operativo, lenguaje de programación, y red. La flexibilidad de UML permite modelar aplicaciones distribuidas que usan cualquier tipo de middleware en el mercado. Construido bajo conceptos fundamentales de orientación a objetos incluyendo clases y operaciones.

Es un perfecto complemento para lenguajes de programación a objetos y ambientes tales como C++, Java, y el reciente C#. También se puede usar UML para modelar aplicaciones que no sean orientadas a objetos, como, por ejemplo, Fortran, VB, o incluso COBOL.

### Estereotipo de UML.

Master Magazine define los estereotipos son el mecanismo de extensibilidad incorporado más utilizado dentro de UML. Un estereotipo representa una distinción de uso. Puede ser aplicado a cualquier elemento de modelado, incluyendo clases, paquetes, relaciones de herencia, etc. Por ejemplo, una clase con estereotipo \’actor\’ es una clase usada como un agente externo en el modelado de negocio. Una clase patrón es modelada como una clase con estereotipo parametrizado, lo que significa que puede contener parámetros.

### 1.9.2 Tipos de diagramas en UML

A continuación se clasifican los tipos de diagramas UML, así como también mencionan algunos ejemplos de cada tipo de diagrama UML clasificado.

**Estructura**

* Diagramas de clases.
* Diagrama de objetos.
* Diagrama de componentes.
* Diagrama de estructura compuesta.
* Diagrama de paquetes.
* Diagrama de despliegue.

**Comportamiento**

* Diagrama de casos de uso
* Diagrama de actividades
* Diagrama de estado

**Interacción**

* Diagrama de secuencia
* Diagrama de colaboración
* Diagrama de tiempo
* Diagrama de interacción

Capítulo 2: Metodología

La metodología que se utiliza para el desarrollo del proyecto, es la metodología RAD (Rapid Application Development, por sus siglas en inglés) ya que será de utilidad para ahorrar tiempo de desarrollo del sistema completo a desarrollar. Ante un cambio de diseño en el sistema por parte del usuario final, ésta metodología nos permite realizar cambios de forma rápida, además de una interacción directa con el cliente.

Se seguirán las siguientes etapas de la metodología:

* Planificación de requerimientos.
* Diseño.
* Desarrollo.
* Implementación de sistema.

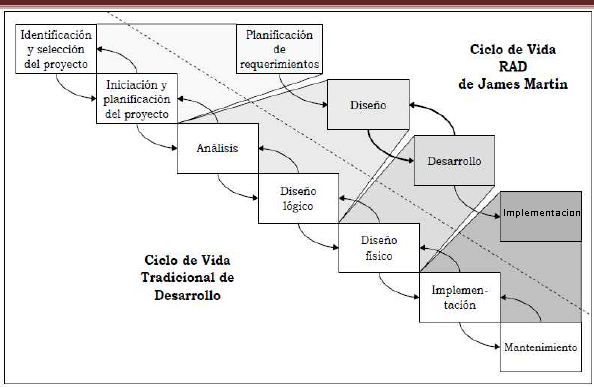


Figura 2. : Ciclo de vida RAD.

## Características, ventajas, desventajas y propiedades de la metodología

Anteriormente se habló acerca de la metodología que se ocupará para poder llevar a cabo el desarrollo y mantenimiento del software que se desarrollará. Sin embargo, es necesario conocer algunas características, ventajas y desventajas de la misma. A continuación se mencionan las características, ventajas, desventajas y propiedades de la metodología.

### Características

* Fue creada por el gurú de la computación James Martin, en 1999.
* Consta de 4 etapas: Planificación de requisitos, implementación, desarrollo e implantación.
* Diseñado para disminuir tiempo de implementación e implantación de Sistemas de Información.
* Involucra construcción de prototipos.
* Involucra reutilización de código.
* Uso de frameworks para aplicaciones web.

### Ventajas

* Ahorro importante en el desarrollo del sistema.
* Uso de generadores de código.
* Se puede ahorrar esfuerzo humano y dinero.
* Permite cambiar el diseño del sistema si el usuario lo requiere.
* Trabaja muy bien cuando la velocidad del desarrollo es crítica.
* Puede utilizar un amplio rango de herramientas y técnicas.

### Desventajas

* Pueden producirse inconsistencias entre diseños internos y sistemas.
* Puede haber posibles violaciones de estándares de programación.
* Algunas herramientas RAD pueden ser caras.

### Propiedades

* Debe de ser entregado en un periodo de dos a seis meses.
* Cada incremento es implementado separadamente con frecuente entrega de partes funcionales del sistema.
* Documentación apropiada actualizada.
* Hace posible una buena comunicación entre el usuario y el desarrollador.
* Reduce el tiempo de debug, pruebas y desarrollo al tener el usuario siguiendo el proyecto.

Como puede observarse, la selección de ésta metodología se basa en la rapidez de la misma, así como en la utilidad de los generadores de código que cuenta la misma. Cabe mencionarse que se ha seleccionado por su capacidad de adaptarse a los cambios que se hagan en cualquier punto del desarrollo del sistema de información.

## Aplicación de la metodología

Para poder aplicar la metodología, es necesario identificar primeramente las etapas de la metodología RAD (ver figura 2.1).



### Planificación de requerimientos

Para poder realizar la planificación de los requerimientos para los sistemas que se implementarán, previamente se llegará a un acuerdo para citas con el responsable de las tiendas de celulares, mismas que están incluidas en el cronograma de actividades que involucra todo el desarrollo del sistema de información.

### Diseño

Después de haber reunido los requisitos del administrador del sistema, se comenzará a crear *mockups* o maquetas que serán el diseño que tendrá la aplicación para administradores y la aplicación para empleados para cada sucursal.

Después de haber terminado de realizar las maquetas visuales, el desarrollador y el administrador de las tiendas de celulares se reunirán una vez más para realizar una retroalimentación de los diseños creados. Si el administrador de las tiendas de celulares no concuerda con el diseño, se procederá a realizar un “ajuste” a los mismos hasta que la persona que administre dichas tiendas de celulares esté de acuerdo con el diseño final.

Al haber terminado lo anterior, será necesario que la persona encargada de las tiendas de celulares deba firmar las hojas en las que se presentaron los

### Desarrollo

Una vez que las dos etapas anteriores concluyeron con éxito, en éste etapa se procederá a desarrollar el sistema para empleados utilizando el IDE RAD Studio XE. En ésta etapa también se procederá a mostrar los prototipos al administrador de las tiendas de celulares, pues si el administrador quisiera hacer modificaciones sencillas, será posible.

Es importante mencionar que durante ésta etapa se realizarán las pruebas alfa para poder detectar errores importantes y mínimos que el software pueda tener. Dichas pruebas serán realizadas primeramente por el programador. De ser posible, el administrador de las tiendas de celulares tendrá la oportunidad de ejecutar el programa por sí mismo y detectar si existe alguna falla que no se haya detectado antes.

Para realizar el sistema de información basado en web, se procederá a realizar el mismo procedimiento especificado para el desarrollo de la aplicación para empleados. En éste caso, será necesario considerar que la aplicación web se ejecute correctamente en distintos dispositivos y que esté disponible en todo momento.

### Implementación

En ésta etapa se procederá a implementar el sistema de empleados en cada sucursal que se tenga para verificar que el sistema de información cumpla con las expectativas del administrador de las tiendas de celulares. Así mismo, se buscará un servidor que soporte Apache Tomcat para poder ejecutar la aplicación web que se haya creado anteriormente.

En ése servidor también se instalará la base de datos maestro que contendrá una réplica de la base de datos que estará en cada sucursal. De ésta forma, si el administrador decide acceder al sistema web desde su computadora, tablet o celular, será capaz de acceder a los registros de cada una de las sucursales.

Capítulo 3: Estado del arte

Los sistemas de información, aunque no son necesarios para las pequeñas y medianas empresas, sí son importantes para incrementar su productividad y eficiencia para sus ventas. Es importante mencionar que el control sobre los pagos de los empleados de una pequeña y mediana empresa (PyMe) también puede ser controlado, ayudando al administrador de la pequeña empresa a establecer un sueldo base, pudiendo así establecer pagos extras (comisiones) por productos que se vendan.

Éste capítulo contiene algunas tesis y monografías desarrolladas por ex alumnos de la Universidad Autónoma del Estado de Hidalgo, así como trabajos de tesis de otras universidades que han utilizado las herramientas computacionales que se utilizarán en éste trabajo o bien un sistema que tenga una función parecida.

Los presentes trabajos están organizados por categorías, de acuerdo a las herramientas computacionales que se utilizaron.

## Groovy & Grails

### “Desarrollo Ágil de Aplicaciones Web con Grails Framework. Caso de estudio: PROMEP­UAEH.”

En ésta monografía, el autor nos habla sobre el framework Groovy and grails y el desarrollo ágil de aplicaciones, el patrón de diseño Modelo Vista Controlador, así como sobre la configuración y arranque del framework Grails. En la monografía se puede conocer el proceso que se llevó a cabo para poder desarrollar el sistema de Control y Seguimiento de Indicadores PROMEP-UAEH (CSI).

La aplicación creada en ésta tesis, es básica y tiene implementada únicamente la hoja de estilos que genera el framework Grails, así mismo, el diseño adaptativo de ésta es básico y no es del todo aceptable. Sin embargo, es importante mencionar que la hoja de estilos generada por el framework Grails puede modificarse, y las *vistas* generadas por Grails también pueden modificarse para crear un sistema web intuitivo y atractivo.

## Microsoft Visual Basic y SQL Server

### “MOBILE STORE MANAGEMENT SYSTEM”

En éste trabajo de tesis, el autor realiza un sistema para la administración de inventarios y de facturación que puede convertirse en la columna vertebral para las pequeñas y medianas organizaciones. Este software proporciona un sistema sencillo para ejecutar en tiendas de celulares. El desarrollo de éste software fue hecho con Visual Basic y SQL Server.

El autor indica que éste software fue inspirado por sistemas de gestión de hoja y lápiz, proporciona una interfaz sencilla para que el usuario pueda gestionar y utilizar el software de forma sencilla. Éste sistema provee capacidad para administrar los celulares almacenados en las sucursales, información de compra y venta, facturación y clientes.

Sin embargo, éste sistema no contempla registro de movimientos de usuarios, ni controla equipos en garantía ni en reparación.

### “MOBILE SHOP MANAGEMENT SYSTEM SOFTWARE”

Este software fue realizado por un grupo de desarrolladores palestinos llamados “hotlinesoftdev”. Este sistema fue desarrollado utilizando Visual Basic .NET. El software permite registrar compras y ventas realizadas de equipos celulares existentes en almacén, así como también permite registrar lo mismo para los accesorios existentes.

El sistema permite que la persona que utilice el software pueda también crear objetos (dar de alta celulares o accesorios, por ejemplo), manejar un control sobre los clientes al mostrar reportes de quiénes de ellos son clientes frecuentes, entre otras cosas. La creación de usuarios es algo que también permite realizar directamente.

Una de las desventajas observadas en este sistema es que no considera algo importante para muchos administradores: los movimientos realizados por los empleados durante su jornada de trabajo. Asimismo, se puede observar que el sistema está hecho para ser utilizado por administradores y no por empleados; esto último puede ser deducido al haber un botón para poder cambiar la contraseña del usuario quien ha iniciado sesión y al haber otro botón para poder crear usuarios.

## MySQL

### “Development of a SaaS Inventory Management System”

El autor indica que el objetivo de ésta tesis es proporcionar una herramienta básica para el seguimiento y monitoreo de las ventas y el inventario a los individuos y las pequeñas empresas que no pueden permitirse la inversión de un Sistema de Gestión de Inventario dedicado completa.

Éste sistema web fue realizado con MYSQL, Apache HTTP, PHP, HTML, AJAX, CSS y JavaScript.

En este trabajo de tesis se explica cómo funciona cada módulo del sistema que desarrolla el autor con detallados diagramas de actividades. Así mismo, se encuentra aclarado el rol de administrador y usuarios, así como las funciones de cada uno.

El sistema anteriormente descrito podría funcionar bien para una tienda de celulares y ser implementado vía web, pues dadas las tecnologías que utiliza, es un sistema robusto. Sin embargo, este sistema no contempla ningún registro de movimientos, por lo tanto, sería necesario tener que adaptar este sistema a las necesidades de alguna tienda de celulares.

## Java

### “Mobile Shop Management System”

Este software es comercial, fué desarrollado por Susantha Kumara y un grupo de programadores de Sri Lanka llamado “TsDevelopers” es un sistema completo para la gestión de equipos celulares que está escrito en Java. Este sistema permite administrar los usuarios que utilizan el sistema con una interfaz simple, aunque repetitiva.

El sistema desarrollado permite registrar ventas, formas de pago y la creación de reportes para los equipos que existen en almacén. Dicho sistema permite administrar no solamente celulares, sino también accesorios, baterías, etcétera.

Este sistema permite también registrar equipos en garantía o en reparación, aunque el usuario tiene que especificar estrictamente para qué fin. Otra desventaja, es que el sistema está hecho para que lo administre un administrador o dueño de alguna tienda de celulares, mas no un empleado; esto último puede ser deducido al notar que el programa permite administrar los usuarios de forma directa, sin necesidad de pasar por algún tipo de filtro.

Así mismo, el software desarrollado por este grupo de desarrolladores permite conocer el estado de los equipos en garantía, así como también permite realizar un filtrado por el estatus de estos.

Es importante hacer mención que el sistema anteriormente descrito no considera registrar un log con movimientos de los usuarios que empleen este software, ya que al no hacerlo, quien sea el administrador de las tiendas de celulares en donde se implemente este software no tendrá conocimiento de las acciones realizadas durante el día.

## Conclusiones

Observando los resultados que han obtenido hasta el momento, los autores que se mencionaron antes, se puede llegar a la conclusión de que MySQL, el framework Grails y Delphi son excelentes herramientas para el desarrollo de sistemas de información en donde se necesite tener un estricto control de la información que se procese día a día.

Así mismo, se puede notar cómo las tres herramientas antes mencionadas, son de gran ayuda si lo que se pretende es desarrollar aplicaciones robustas en un tiempo corto.

Capítulo 4: Herramientas computacionales

En éste capítulo se pretende dar a conocer las características principales, así como ventajas y desventajas de cada una de las herramientas computacionales que se ocuparán a lo largo del desarrollo del sistema de empleados y del sistema para administradores. Esto con objeto de que el lector conozca de forma rápida las herramientas y para que pueda evaluar la potencia de las mismas.

A continuación se muestran y mencionan las herramientas que se utilizarán en el desarrollo de éstos sistemas de información.

## Embarcadero Delphi XE ®

Embarcadero RAD Studio es un entorno de desarrollo de software que está diseñado para programación de propósito general, con énfasis a la programación visual. En Delphi XE, el lenguaje de programación utilizado es una versión moderna del lenguaje de programación Pascal: Object Pascal.

Entre las aplicaciones más populares programadas con Delphi XE, destaca Skype, programa de telefonía por IP.

### Características

* Utiliza Object Pascal como lenguaje de programación.
* Produce archivos ejecutables para GNU/Linux, Windows, y plataforma .NET.
* Permite conectividad a diferentes motores de bases de datos.
* Extensa gama de herramientas disponibles.
* Comparador de archivos integrado.

### Ventajas

* Permite desarrollar aplicaciones en modo diseño.
* Permite desarrollar aplicaciones para Mac, Linux, Windows, y plataforma .NET.
* Se puede agregar componentes de terceras personas y compilarlos.
* Contiene generadores de código.
* Permite el uso de eventos en la aplicación a desarrollar.

### Desventajas

* Precio alto.



Figura 4. : Logotipo del IDE Embarcadero Delphi XE.

## Oracle MySQL ®

MySQL es un Sistema Gestor de Bases de Datos Relacional (SGBDR), multihilo y multiusuario. Se ofrece bajo la licencia GNU GPL. Sin embargo, aquellas empresas que quieran incorporar MySQL en sectores privativos, deben de comprar la licencia específica que les permita éste uso. MySQL es usado por muchos sitios web populares tales como Wikipedia, Google, Facebook, Twitter, Flickr, Youtube, entre otros.

### Características

* Multiplataforma.
* Está escrito en C y C++.
* Registros de longitud fija y variable.
* El objetivo principal de MySQL son la robustez y velocidad.
* Se desarrolló para manejar grandes bases de datos.
* Soporte de control para transacciones en tablas transaccionales.
* Posee un buen control de acceso de usuarios y seguridad en datos.
* Soporte para procedimientos almacenados, sub-consultas y triggers.
* Permite replicación de bases de datos.
* Trabaja con el entorno MySQL-Cluster.

### Ventajas

* Facilidad de configuración e instalación.
* Soporta gran variedad de sistemas operativos.
* Multiusuario.
* Permite concurrencia.
* Puede ser ejecutado en una máquina con escasos recursos.
* Se integra muy bien con PHP.

### Desventajas

* El soporte para triggers es básico.
* Los privilegios para una tabla no se eliminan cuando se borra una tabla.



Figura 4. : Logotipo de Oracle MySQL ®.

## Groovy

Lenguaje de programación orientado a objetos implementado sobre Java. Tiene características similares a Python, Ruby, Perl y Smalltalk. El bytecode generado durante la compilación es compatible con el generado por la Java Virtual Machine, por ende, las aplicaciones en Groovy son multiplataforma.

### Características

* Comparte el mismo modelo de objetos, hilos, y de seguridad.
* Los programadores de Java se familiarizan con Groovy fácilmente.
* Los puntos y comas son opcionales.

### Ventajas

* Puede ser usado como lenguaje de scripting.
* Sintaxis parecida a Java.
* Multiplataforma.
* Se pueden acceder a todas las APIs de Java.
* Sintaxis minimalista.

### Desventajas



Figura 4. : Logotipo del lenguaje de programación Groovy.

## Grails

Framework para desarrollar aplicaciones web desarrollado sobre el lenguaje de programación Groovy (mismo que se basa en Java Platform). Grails es un framework altamente productivo al seguir paradigmas como convención sobre configuración ó "no te repitas" (DRY) que proporciona un entorno de desarrollo estandarizado y que encapsula una gran parte de los detalles de configuración al programador.

### Características

* Ofrece un framework web de alta productividad para la plataforma Java.
* Proporciona un entorno de desarrollo orientado a pruebas.
* Soporte AJAX.

### Ventajas

* Soporta uso de Scaffolding para operaciones CRUD (Create, Read, Update, Delete).
* Es fácil integrarlo con librerías Java.
* Ha sido diseñado de acuerdo al patrón de diseño MVC.
* Multiplataforma.

### Desventajas

* Existen problemas si se trabajan aplicaciones multi-hilos.
* Como es un lenguaje interpretado, incrementa el peso y eso directamente afecta el tiempo de ejecución.



Figura 4. : Logotipo del framework Grails.

## Twitter Bootstrap

Es una colección libre de herramientas para crear sitios web y aplicaciones web. Contiene plantillas basadas en CSS y HTML para tipografías, formularios, botones, nagevación y otros compontentes de la interfaz, así como también extensiones adicionales de JavaScript.

### Características

* Tiene compatibilidad con la mayoría de los navegadores.
* Soporta diseño adaptativo; permitiendo a la aplicación web o página web adaptarse a cualquier dispositivo con el que se acceda.
* Proyecto open Source y disponible en GitHub.

### Ventajas

* Muy personalizable.
* Brinda consistencia.
* Incluye algunos plugins jQuery.
* Existe una gran variedad de plantillas disponibles.
* Soporta diseño adaptativo. (responsive design)
* Utiliza LESSCSS.

### Desventajas

* Aunque es muy personalizable, muchos sitios acaban pareciéndose entre sí.
* Al comenzar a usarlo, Bootstrap tiene pocos plugins jQuery.

Capítulo 5: Diseño de la base de datos

Antes de desarrollar cualquier sistema de información, es necesario primeramente tener un buen diseño de la base de datos, pues con un buen diseño, se evitarán innecesarias reingenierías de la misma.

En éste capítulo se expone el modelado de datos que se utilizará para desarrollar los sistemas de información que serán utilizados por los empleados de las tiendas de celulares y por el administrador o administradores de dichas tiendas.

## Diagrama entidad-relación

Según Henry F. Korth, Abraham Silberschatz y S. Sudarshan, el modelo entidad-relación (E-R) es una percepción del mundo real consistente en objetos básicos llamados “entidades” y de “relaciones” entre dichos objetos.

Este diagrama se creó para facilitar el diseño de las bases de datos, mostrando la especificación de un “esquema” de la empresa, misma que representa la estructura lógica de la base de datos.

Una **entidad** es algo de la vida real, es decir, una cosa u objeto. Un ejemplo de esto sería decir que cada persona en un desarrollo es una entidad. Esta entidad contiene propiedades que pueden identificar a la entidad de forma única e inequívoca.

Los **atributos** describen las propiedades que posee cada miembro dentro de un conjunto de entidades.

A continuación se presentan y explican los diagramas entidad-relación de los sistemas para empleados y para administradores.

### Diagrama E-R para sistema de empleados

El empleado, generalmente registra ventas que se hagan a diario, considerando aspecto como la fecha de venta, precio del celular, número de celular que se vendió, así como los datos del cliente (tales como dirección, nombre y teléfono de contacto). El empleado de igual forma puede registrar uno o ningún equipo para garantía o para reparación; considerando aspectos tales como: IMEI del equipo celular, ICCID del equipo celular, observaciones o bien accesorios recibidos (en caso de proceder como garantía), así como la fecha en la que se recibió el equipo celular.

La figura 5.1 ilustra el diagrama E-R de lo descrito anteriormente, así como también los atributos que tendrá cada entidad mostrada en el diagrama.

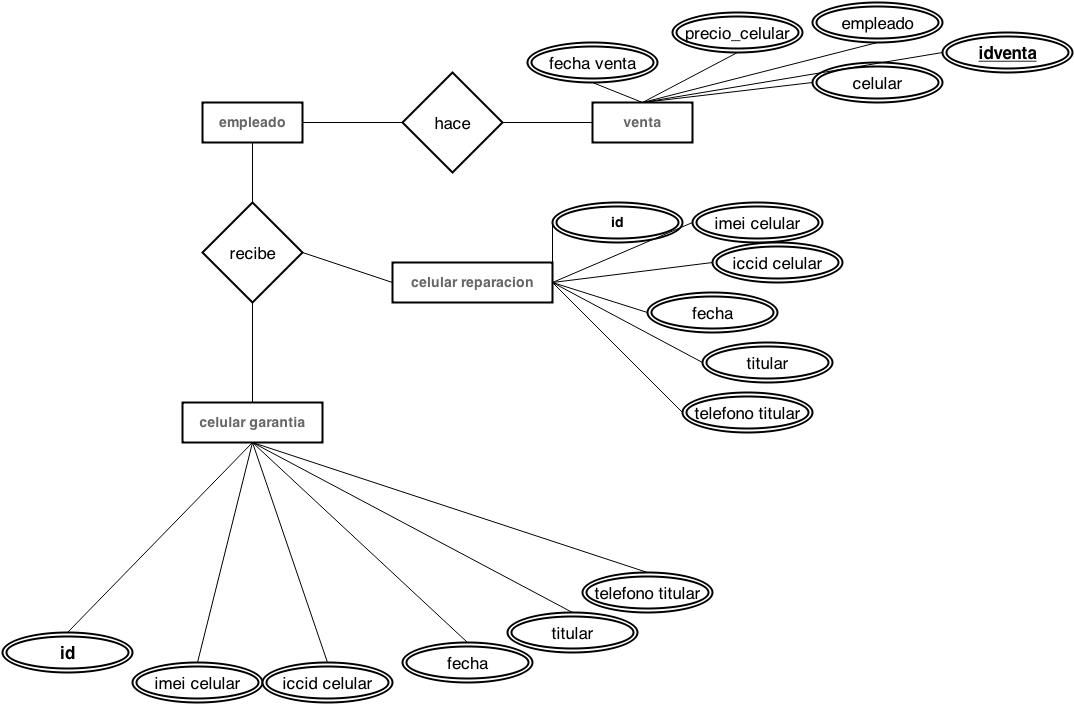


Figura 5.1.1. : Diagrama E-R para sistemas de empleados.

### Diagrama E-R para sistema de administradores

El administrador puede dar de alta nuevas sucursales (únicamente puede dar de alta sucursales dentro del estado de Hidalgo), considerando aspectos como el nombre de la sucursal y el municipio en donde esta se encuentra localizada. Otra posibilidad es que el administrador puede dar de alta códigos de recargas especificando en qué sucursal serán asignados, así mismo, el administrador puede dar de alta nuevos usuarios, asignándoles un nivel o jerarquía.

El administrador podrá leer los movimientos del usuario que tenga registrados la base de datos.

La figura 5.1.2.1 ilustra la interacción del administrador con el sistema para administradores o bien para el módulo de administradores.

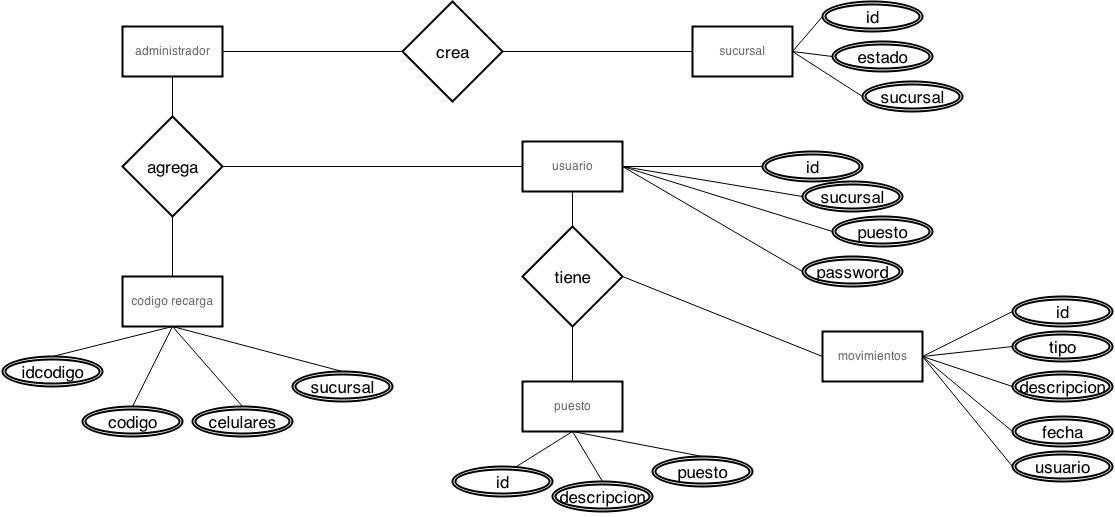


Figura 5.1.2. : Diagrama E-R para el sistema de administradores.

## Diagrama relacional

El modelo relacional es el modelo más utilizado en la actualidad, y su función es modelar problemas reales y administrar datos de forma dinámica. Estos diagramas se basan en el uso de “relaciones”, las cuales pueden considerarse como conjuntos de datos llamados “tuplas”. Sin embargo, frecuentemente se conceptualiza pensando que cada relación fuese una tabla que tiene registros dentro de ella (donde cada file de la tabla sería una tupla), y columnas (campos).

La figura 5.1.1 ilustra el diagrama relacional que se utilizará para el desarrollo de la base de datos.

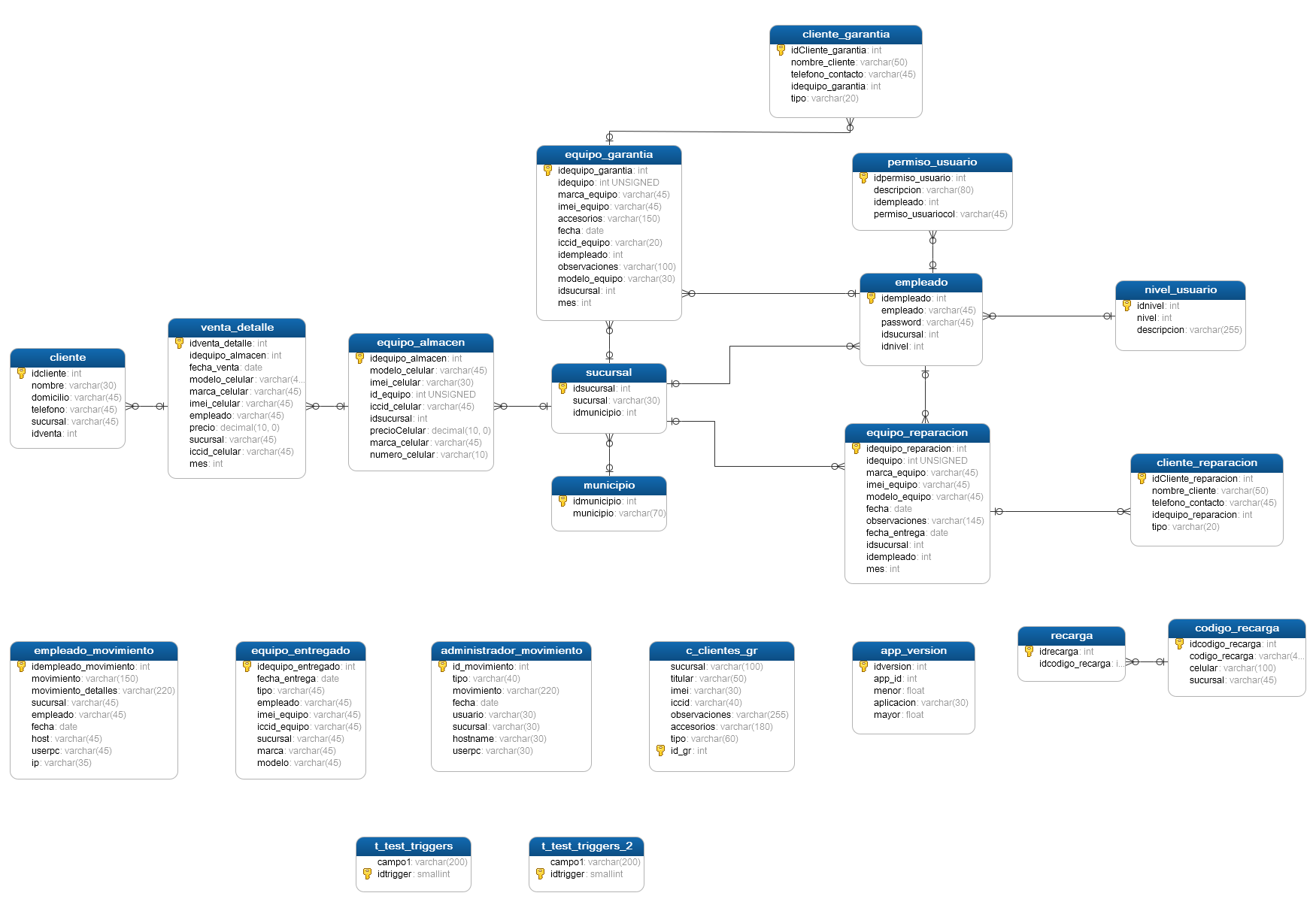


Figura 5.1. : Diagrama Relacional de la base de datos.

## Entidades

En la presente sección se pueden observar las entidades, en conjunto con los atributos de estas. Asimismo, es posible ubicar el atributo con la que una entidad tiene relación con otra entidad. A continuación se explican los atributos y el uso principal de dicha tabla.

En la entidad “cliente” se almacenarán datos importantes del cliente respecto a las ventas de los celulares. Esta entidad tendrá atributos como el id del cliente, nombre, domicilio, teléfono de contacto, la sucursal en donde se encuentra alojada la información de dicho cliente, así como la id de la venta realizada. En la tabla 5.2.1 se pueden observar los atributos que anteriormente se mencionaron, así como su llave primaria y foránea.

Tabla 5.2. : Entidad "cliente"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| Idcliente (PK) | Integer | ID para el cliente de ventas. | - |
| Nombre | Varchar | Nombre del cliente. | - |
| Domicilio | Varchar | Domicilio del cliente. | - |
| Teléfono | Varchar | Teléfono del cliente. | - |
| Sucursal | Varchar | Sucursal en donde se registró el cliente. | - |
| Idventa (FK) | Integer | ID para los detalles de la venta. | Venta\_detalle |

En la entidad “venta\_detalle” se almacenarán los datos respectivos a una venta. Entre estos se encuentran la id de la venta, la fecha de venta, modelo, marca, imei, precio del celular vendido, así como también el empleado que realizó la venta y la sucursal en donde fue vendido. También se encuentra disponible un atributo “mes” de tipo entero, con el cual se podrán realizar búsquedas y reportes para el mes seleccionado.

Tabla 5.2. : Entidad "venta\_detalle"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idventa\_detalle (PK) | Int | ID generada para los detalles de las ventas. | - |
| fecha\_venta | Date | Domicilio del cliente. | - |
| modelo\_celular | Varchar | Teléfono del cliente. | - |
| marca\_celular | Varchar | Sucursal en donde se registró el cliente. | - |
| imei\_celular | Varchar | IMEI del celular vendido. | - |
| empleado | Varchar | Empleado que registró la venta. | - |
| precio | Decimal | ID para los detalles de la venta. | Venta\_detalle |
| sucursal | Varchar | Sucursal en la que se registró la venta. | - |
| iccid\_celular | Varchar | ICCID de celular vendido. | - |
| mes | Int | Número de mes de venta registrada. | - |

En la entidad “empleado”, existen atributos para los empleados, tales como el id del empleado, el empleado en sí, que contendrá el nombre del empleado o el usuario que ingresará al sistema. El atributo password contendrá la contraseña que el administrador del sistema podrá definir. Existe un atributo foráneo que es el ID de la sucursal a la que pertenece dicho empleado y también existe otra llave foránea, la cual indica qué tipo de nivel (o jerarquía) tiene el usuario. Con esto último, el sistema podrá definir si el usuario tiene permisos para acceder a ciertas partes de la aplicación (como al panel de administradores, por ejemplo). En la tabla 5.2.3 se muestran los atributos que anteriormente fueron descritos, así como también las llaves foráneas y las relaciones con otras entidades.

Tabla 5.2. : Entidad "empleado"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idempleado (PK) | Integer | ID para el empleado. | - |
| Empleado | Varchar | Nombre del empleado. | - |
| Password | Varchar | Password del empleado. | - |
| Idsucursal | Varchar | ID de la sucursal a la que pertenece el empleado. | Sucursal |
| idnivel | Varchar | ID del nivel del usuario. | Nivel\_usuario |

La entidad “cliente\_garantia” contiene datos de los clientes que tengan algún celular en garantía. Dichos datos son el nombre del cliente, algún teléfono de contacto que el cliente brinde, la id para tal cliente, y una ID del celular en garantía que le corresponde al cliente. Con el ID de equipo de garantía, se podrán listar los datos del celular (IMEI, accesorios recibidos, ICCID, modelo del celular, marca y observaciones) que el empleado dio de alta. La tabla 5.3.4 contiene los atributos de esta entidad, el tipo, descripción de cada atributo así como la relación con la entidad “equipo\_garantia”.

Tabla 5.2. : Entidad "cliente\_garantía"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idCliente\_garantia (PK) | Integer | ID para el cliente en garantía. | - |
| nombre\_cliente | Varchar | Nombre del cliente en garantía. | - |
| telefono\_contacto | Varchar | Teléfono del cliente en garantía | - |
| idequipo\_garantia (FK) | Integer | ID del equipo en garantía del cliente. | Equipo\_garantia |

La entidad llamada “cliente\_reparacion” contiene atributos que describen al cliente; estos son el nombre del cliente, teléfono del cliente así como el ID de reparación que tiene. Asimismo, se utiliza la llave foránea “idequipo\_reparacion” para poder conocer detalles del equipo (datos como el IMEI, ICCID, observaciones, modelo del celular y marca) que los clientes tengan en reparación. La tabla 5.3.5 contiene la descripción de los atributos mencionados anteriormente, así como la relación existente con la entidad “equipo\_reparacion”, también indica el tipo de dato para cada atributo.

Tabla 5.2. : Entidad "cliente\_reparacion"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idCliente\_reparacion (PK) | Integer | ID para el cliente en reparación. | - |
| nombre\_cliente | Varchar | Nombre del cliente en reparación. | - |
| telefono\_contacto | Varchar | Teléfono del cliente en reparación | - |
| idequipo\_reparacion (FK) | Integer | ID del equipo en garantía de reparación. | Equipo\_reparacion |

Mediante el uso de códigos de barra impresos, se considera la entidad “código\_recarga”, pues en ella se definen atributos tales como la ID del código para recarga, el código de recarga disponible y el celular o celulares que éste tiene asignados, así como la sucursal en la que se encuentran alojados dichos códigos. La tabla 5.3.6 muestra la descripción de dichos atributos, así como el tipo de datos los cuales serán asignados a estos.

Tabla 5.2. : Entidad "codigo\_recarga"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idcodigo\_recarga (PK) | Integer | ID para el código de recarga. | - |
| codigo\_recarga | Varchar | Código de barras para recarga. | - |
| Celular | Varchar | Celular o celulares asignados para un código. | - |
| sucursal | Varchar | Sucursal a la que pertenecen los códigos. | - |

Gracias a la entidad llamada “equipo\_entregado”, el administrador de las tiendas de celulares podrá obtener una lista de equipos entregados a clientes, obteniendo datos como el IMEI del celular entregado, ICCID del celular entregado, el nombre del empleado quien entregó un equipo celular, la fecha de entrega y el tipo de entrega (garantía o reparación). La tabla 5.3.7 ilustra los atributos anteriormente descritos, así como el tipo de dato que estos utilizarán.

Tabla 5.2. : Entidad "equipo\_entregado"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idequipo\_entregado (PK) | Integer | ID para el código de recarga. | - |
| fecha\_entrega | Date | Fecha de entrega de un celular. | - |
| empleado | Varchar | Empleado que entregó un celular. | - |
| imei\_equipo | Varchar | IMEI del celular entregado. | - |
| iccid\_equipo | Varchar | ICCID del celular entregado. | - |
| sucursal | Varchar | Sucursal en donde se entregó un celular. | - |
| tipo | Varchar | Tipo de celular entregado (garantía o reparación) | - |

El software para empleados controlará también los equipos que se reciban en garantía, considerando datos del equipo celular (como el IMEI, ICCID, marca y modelo del celular, así como también la fecha en la que el equipo celular fue dado de alta). En la tabla 5.3.8 se muestran los atributos que anteriormente se describieron, así como también la descripción y el tipo de dato que estos utilizarán. Se considera un IDEquipo de tipo entero para ordenamientos dentro del programa principal.

Tabla 5.2. : Entidad "equipo\_garantia"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idequipo\_garantia (PK) | Integer | ID para el celular en garantía. | - |
| idequipo | Integer | ID del celular para ordenamiento. | - |
| marca\_equipo | Varchar | Marca del celular en garantía. | - |
| imei\_equipo | Varchar | IMEI del celular en garantía. | - |
| Accesorios | Varchar | Accesorios del celular en garantía. | - |
| Fecha | Date | Fecha de recepción del celular en garantía. | - |
| iccid\_equipo | Varchar | ICCID del celular en garantía. | - |
| Idempleado (FK) | Integer | ID del empleado que recibe celular en garantía. | Empleado |
| Observaciones | Varchar | Observaciones del celular en garantía. | - |
| modelo\_equipo | Varchar | Modelo del celular en garantía. | - |
| Idsucursal (FK) | Integer | Sucursal que recibe el celular en garantía. | Sucursal |

El software para empleados controlará también los equipos que se reciban en reparación, considerando datos del equipo celular (como el IMEI, ICCID, marca y modelo del celular, así como también la fecha en la que el equipo celular fue dado de alta en reparación).

En la tabla 5.3.9 se muestran los atributos que anteriormente se describieron, así como también la descripción y el tipo de dato que estos utilizarán. Se considera un IDEquipo de tipo entero para ordenamientos dentro del programa principal.

Tabla 5.2. : Entidad "equipo\_reparacion"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idequipo\_reparacion (PK) | Integer | ID para el celular en reparación. | - |
| idequipo | Integer | ID del celular para ordenamiento. | - |
| marca\_equipo | Varchar | Marca del celular en reparación. | - |
| imei\_equipo | Varchar | IMEI del celular en reparación. | - |
| Accesorios | Varchar | Accesorios del celular en reparación. | - |
| Fecha | Date | Fecha de recepción del celular en reparación. | - |
| iccid\_equipo | Varchar | ICCID del celular en reparación. | - |
| idempleado | Integer | ID del empleado que recibe celular en reparación. | Empleado |
| Observaciones | Varchar | Observaciones del celular en reparación. | - |
| modelo\_equipo | Varchar | Modelo del celular en reparación. | - |
| idsucursal | Integer | Sucursal que recibe el celular en reparación. | Sucursal |

En caso de que se abra otra sucursal, los administradores de las tiendas de celulares podrán dar de alta nuevas sucursales, considerando datos como el nombre de la sucursal y el municipio dentro del estado de Hidalgo. Las tablas 5.3.10 y 5.3.11 muestran los atributos que tienen estas tablas, brindando también una breve descripción de los atributos listados en dichas entidades.

Tabla 5.2. : Entidad "municipio"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| Idmunicipio (PK) | Integer | ID del municipio. | - |
| municipio | varchar | Nombre del municipio en donde se cree otra sucursal. | - |

Tabla 5.2. : Entidad "sucursal"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idsucursal (PK) | Integer | ID de la sucursal. | - |
| sucursal | varchar | Sucursal. | - |
| Idmunicipio (FK) | integer | ID del municipio a la que pertenece la sucursal. | Municipio |

La entidad “nivel\_usuario” define la jerarquía o “nivel” que el usuario tiene a nivel de la aplicación de escritorio. Cuando el administrador desee dar de alta un nuevo empleado, podrá definir si es administrador o empleado. En cambio, si el desarrollador del sistema decide agregar otro desarrollador, podrá elegir entre darlo de alta como desarrollador, administrador o bien como empleado. Esta entidad es un complemento de la entidad “empleado” (ver tabla 5.2.3).

La tabla 5.3.13 define los atributos, tipo de dato y descripción de la entidad anteriormente descrita.

Tabla 5.2. : Entidad "nivel\_usuario"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idnivel (PK) | Integer | ID para el nivel de usuario. | - |
| Nivel | Integer | Nivel de usuario. | - |
| Descripción | Varchar | Descripción del nivel de usuario. | - |

La entidad titulada “equipo\_almacen” contiene datos de los equipos celulares, así como la sucursal en donde estos se encuentran alojados. Dichos atributos son la id del equipo en almacén, modelo y marca del equipo celular, IMEI e ICCID del equipo celular, precio del celular y la ID de la sucursal asignada para dichos equipos celulares. Cabe mencionar que también se encuentra disponible el número asignado para cada equipo celular.

La tabla 5.3.13 muestra los atributos, el tipo de datos y una descripción de los mismos, así como también se indica si existe una posible relación con otras entidades.

Tabla 5.2. : Entidad "equipo\_almacen"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idequipo\_almacen (PK) | Integer | ID del celular en almacén. | - |
| modelo\_celular | Integer | Modelo del celular en almacén. | - |
| imei\_celular | Varchar | IMEI del celular en almacén. | - |
| id\_equipo | Integer | ID del celular en almacén para ordenamientos. | - |
| iccid\_celular | Date | ICCID del celular en almacén. | - |
| Idsucursal (FK) | Integer | ID de la sucursal a la que pertenece el celular. | Sucursal |
| precioCelular | Integer | Precio del celular en almacén. | - |
| marca\_celular | Varchar | Marca del celular en almacén. | - |
| numero\_celular | Varchar | Número asignado del celular en almacén. | - |

El sistema para empleados podrá registrar los movimientos que tenga el empleado (inicio de sesión, cierre de sesión, venta de equipo, entrega de equipo en garantía y en reparación, recepción de equipos celulares en garantía y en reparación). La tabla 5.3.14 contiene los atributos y una descripción de estos que fueron necesarios considerar para poder crear los registros de movimiento.

Tabla 5.2. : Entidad "empleado\_movimiento"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| idempleado\_movimiento (PK) | Integer | ID del movimiento registrado para empleados. | - |
| Movimiento | Integer | Tipo de movimiento registrado. | - |
| movimiento\_detalles | Varchar | Detalles del movimiento registrado. | - |
| Sucursal | Integer | Sucursal en la que se registra el movimiento. | - |
| Empleado | Integer | Empleado que registra movimiento. | - |
| fecha | Integer | Fecha en la que se registra el movimiento. | - |

El sistema para empleados tiene un módulo para administradores. Si el administrador desea dar de alta nuevas sucursales, empleados, equipos celulares y códigos para recarga, será necesario que ingrese a este módulo. El sistema también registra los movimientos que tiene el administrador dentro del programa, sin embargo, estos movimientos no serán vistos por él, sino más bien podrán ser vistos por el desarrollador del sistema.

Tabla 5.2. : Entidad "administrador\_movimiento"

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descripción | Relación |
| id\_movimiento (PK) | Integer | ID del movimiento registrado para empleados. | - |
| Tipo | Integer | Tipo de movimiento registrado. | - |
| Movimiento | Integer | Detalles del movimiento registrado. | - |
| Fecha |  | Fecha en la que se registra el movimiento. |  |
| Usuario | Integer | Administrador que registra movimiento. | - |
| sucursal | Integer | Sucursal en la que se registra el movimiento. | - |

# Capítulo 6: Diseño del sistema

En el presente capítulo se ilustra el uso del sistema para empleados y del sistema para el administrador usando diagramas de casos de uso, así como también se ilustra el proceso del flujo de trabajo gracias a los diagramas de flujo en éste capítulo.

## Diagramas de casos de uso

Los diagramas de casos de uso son una herramienta visual que define una serie de pasos o actividades que puedan realizarse para llevar a cabo un cierto proceso. Estos diagraman muestran de forma amigable la forma en que el usuario y el sistema de información podrían interactuar.

Asimismo, estos diagramas se emplean para poder ilustrar los requerimientos del sistema al poder ilustrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

En la presente sección se ilustran los diagramas de casos de uso para cada “situación” o “caso de uso” que el sistema de empleados y el sistema para administradores puedan tener, así como una breve descripción del uso de cada diagrama de casos de uso mostrado.

### Sistema de empleados

A continuación se presentan el conjunto de diagramas de casos de uso que se considerarán para poder desarrollar el sistema de información que será utilizado por los empleados.

#### Venta de celular

La figura 6.1 pretende explicar de una forma gráfica el flujo de trabajo que es llevado a cabo al momento de realizar una venta de un equipo celular para ayudar a comprender las acciones realizadas por el cliente y el empleado en una venta de un equipo celular.

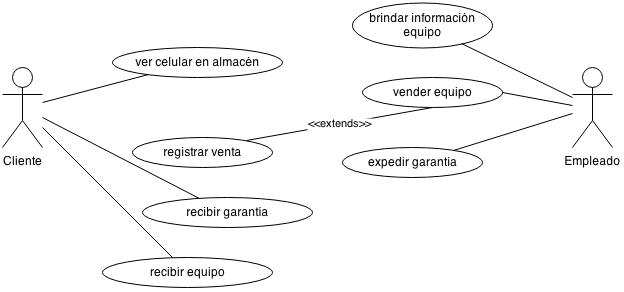


Figura 6. : Diagrama de casos de uso para la venta de un celular.

#### Recepción de equipos en reparación

Las figuras 6.2 y 6.3 ilustran el flujo de trabajo y acciones necesarias por parte del cliente, empleado y administrador o dueño de la tienda de celulares para poder procesar un equipo celular recibido para reparación o garantía respectivamente.

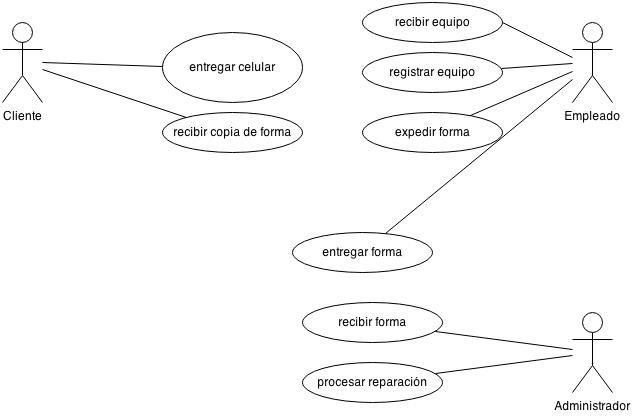


Figura 6. : Diagrama de casos de uso para la recepción de equipos en reparación.

#### Recepción de equipos en garantía

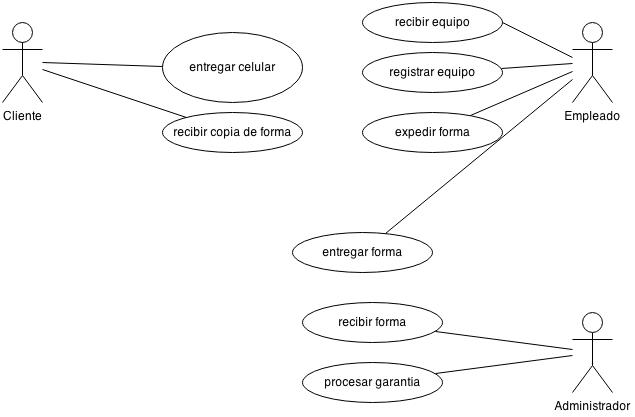


Figura 6. : Diagrama de casos de uso para recepción de equipos en garantía.

#### Venta de recargas

Para poder realizar la venta de recargas, el cliente primeramente tiene que solicitar una recarga y mostrar el código para recarga que previamente se le fue asignado. El empleado entonces podrá hacer una búsqueda de dicho código y revisar qué celular o celulares fueron asignados, para proceder a realizar y registrar la recarga. La figura 6.4 ilustra los procesos que anteriormente se mencionaron.

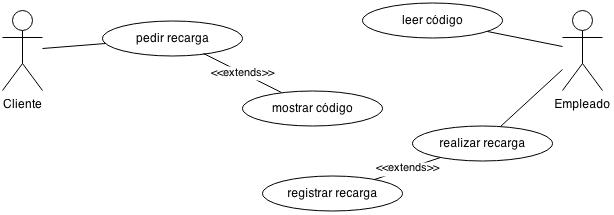


Figura 6. : Diagrama de casos de uso para venta de recargas.

### Sistema de administradores

A continuación se presentan el conjunto de diagramas de casos de uso que se considerarán para poder desarrollar el sistema de información que será utilizado por los administradores o dueños de las tiendas de celulares.

#### Alta de equipos para inventarios

El administrador tiene derecho para poder dar de alta equipos celulares a cualquiera de las sucursales, o bien dentro de cualquier sucursal, en caso de estar en alguna de estas. Será necesario que seleccione la sucursal, registre los datos del celular a asignar y finalmente les asigne un precio. Una vez que esto último haya sido completado exitosamente, se podrá dar de alta en la sucursal y el sistema lo registrará en la base de datos. La figura 6.5 ilustra la interacción que el administrador tendrá con el sistema dentro de este ámbito.

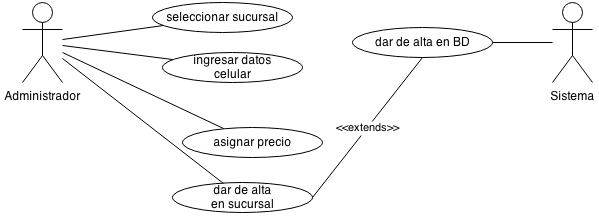


Figura 6. : Diagrama de casos de uso para alta de equipos en inventarios.

#### Revisión de movimientos del usuario

Para poder revisar el log de movimientos del usuario, el administrador primeramente tiene que seleccionar si desea conocer los movimientos de un empleado en específico o bien de todos los empleados. Una vez hecho esto, el sistema buscará y cargará los movimientos registrados para el usuario o usuarios especificados anteriormente, de esta forma, el administrador podrá ver los movimientos registrados, así como poder guardarlos en formato .TXT o .PDF.

La figura 6.6 ilustra lo anteriormente descrito y muestra la interacción que tiene el administrador con el sistema de información.

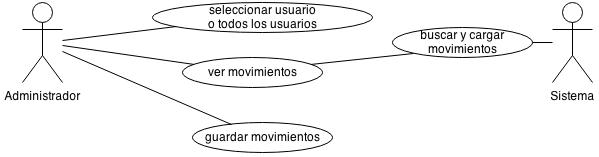


Figura 6. : Diagrama de casos de uso para revisión de movimientos de usuarios.

#### Consulta de equipos entregados

El sistema para administradores tendrá la posibilidad de consultar los equipos que el empleado o los empleados han entregado, considerando la posibilidad de poder filtrarlos por tipo (reparación o garantía). Cabe mencionar que el administrador podrá generar un reporte de dicho catálogo de equipos entregados. La figura 6.7 ilustra la interacción del administrador que tendrá con el sistema.

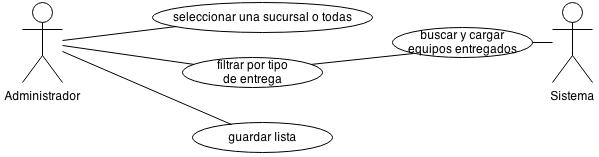


Figura 6. : Diagrama de casos de uso para consulta de equipos entregados.

#### Consulta de equipos recibidos

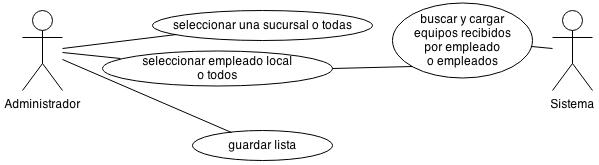


Figura 6. : Diagrama de casos de uso para consulta de equipos recibidos.

## Diagramas de flujo de datos

Un diagrama de flujo de datos (DFD) es una representación gráfica del flujo de datos de un sistema de información. Generalmente, estos diagramas son un paso preliminar para crear un panorama del sistema de información que posteriormente pueda ser elaborado. Los DFD también pueden ser usados para la visualización del proceso de los datos.

Estos diagramas muestran el tipo de información que será de entrada y salida del sistema; de donde vendrán y a donde irán, y donde en serán almacenados los datos. Este tipo de diagramas no muestran información sobre los tiempos de procesamientos o información sobre si el proceso operará en secuencia o en paralelo.

A continuación se muestran los diagramas de flujo de datos que se utilizaron para obtener un panorama del flujo de datos para los sistemas de información a desarrollar.

### Sistema para empleados

#### Acceso al sistema de empleados

{agregar texto}

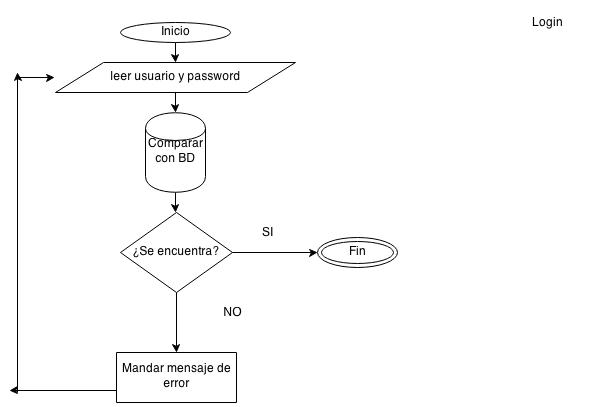


Figura 6. : Diagrama de flujo para el acceso al sistema de empleados.

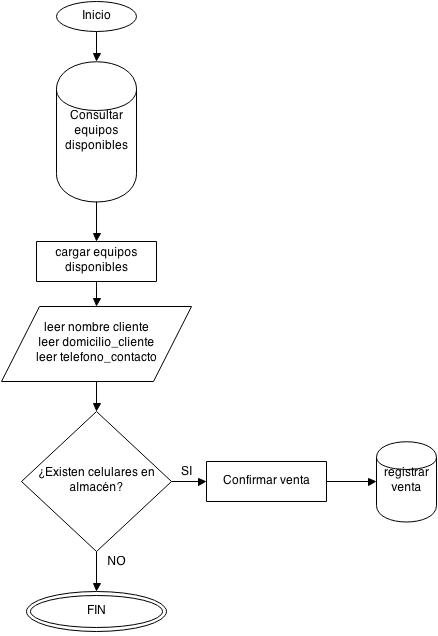


Figura 6. : Diagrama de flujo de venta de un celular.

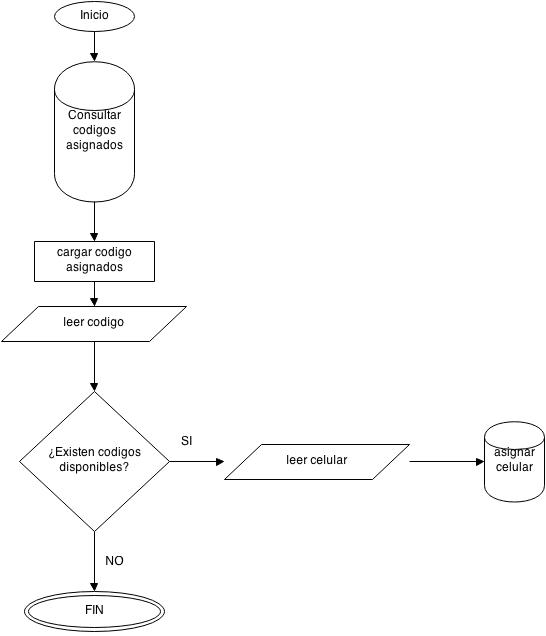


Figura 6. : Diagrama de flujo de asignación de un código de recarga.

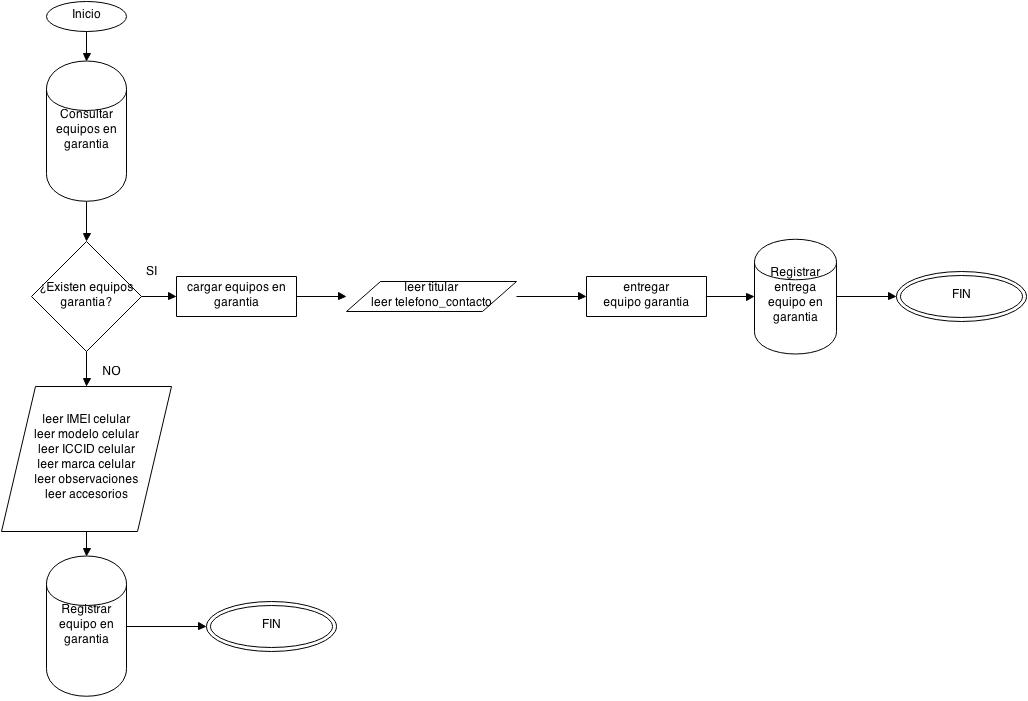


Figura 6. : Diagrama de flujo de recepcion y entrega garantía.

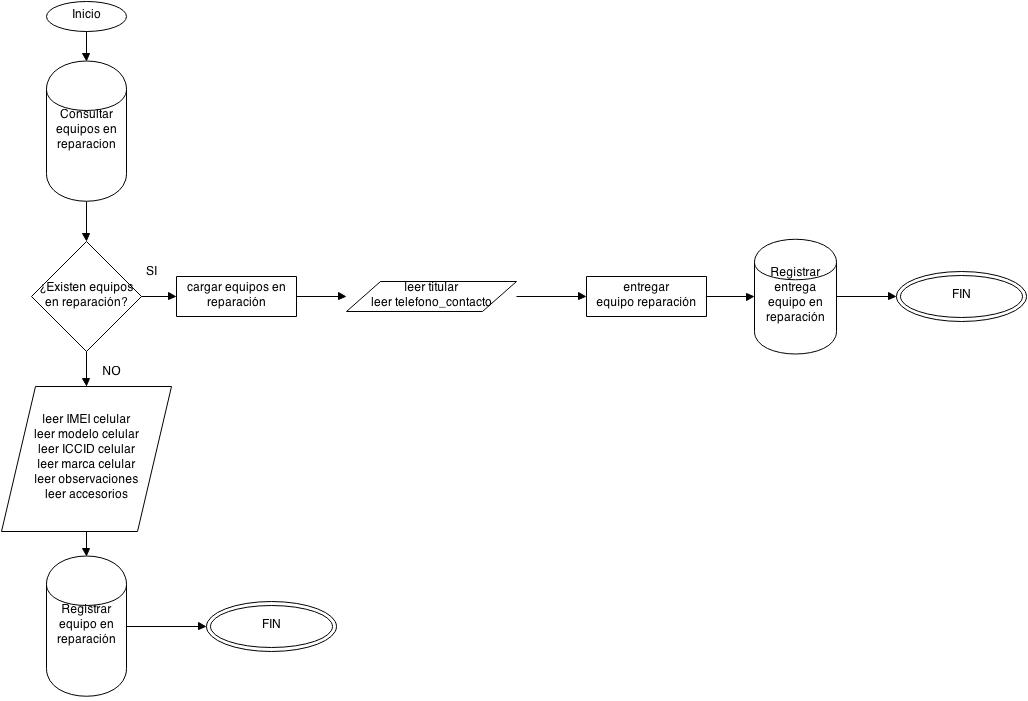


Figura 6. : Diagrama de flujo de recepción y entrega reparación.

# Conclusiones parciales

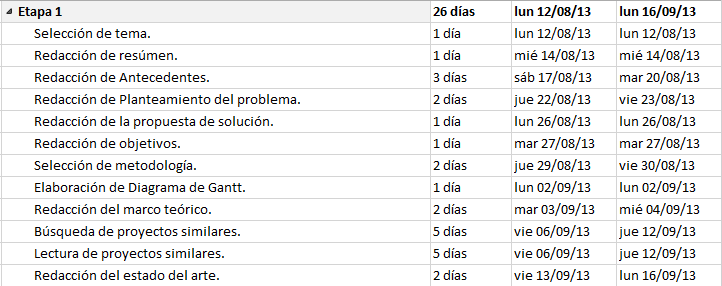
Se puede concluir que el desarrollo de este sistema va a ser, sin duda, un reto para la persona quien presenta este trabajo. Así mismo, es importante decir que gracias a las asesorías recibidas y revisiones con los asesores de éste trabajo, el proyecto poco a poco está saliendo adelante.

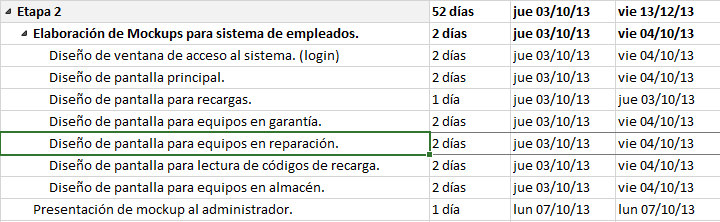
Se han realizado algunas pruebas locales, revisiones con quien hoy en día es el administrador o responsable de llevar la contabilidad de todas las sucursales de las tiendas de celulares “Celulares Castillo” y se han realizado algunas observaciones en cuanto al sistema de empleados, así como también se han recibido críticas constructivas en cuanto al desarrollo del mismo.

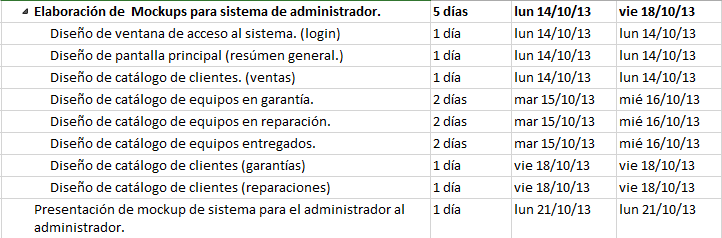
Hasta el día de entrega de este proyecto, los avances, aunque son pocos, deja en claro que con tiempo y una organización moderada se pueden sacar adelante proyectos medianos.

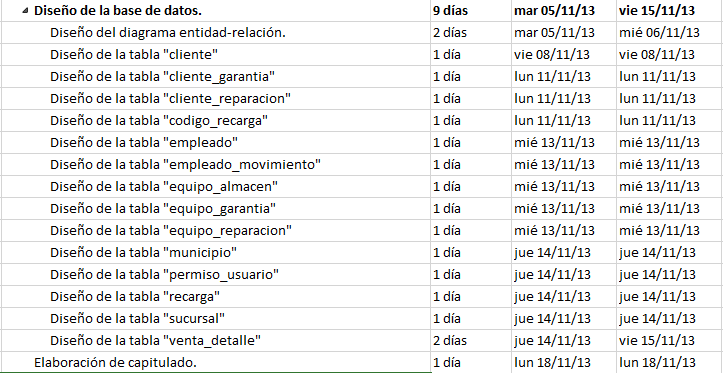
El desarrollo de este sistema está llevándose a una velocidad destacable y, sin lugar a dudas, una de las metas personales establecidas en este presente documento será cumplida: El desarrollo de sistema, funcional y estable en menos de 1 año.

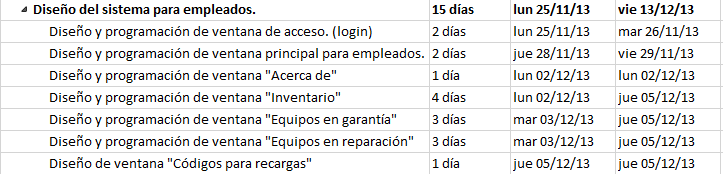
# Anexo A: Cronograma de actividades

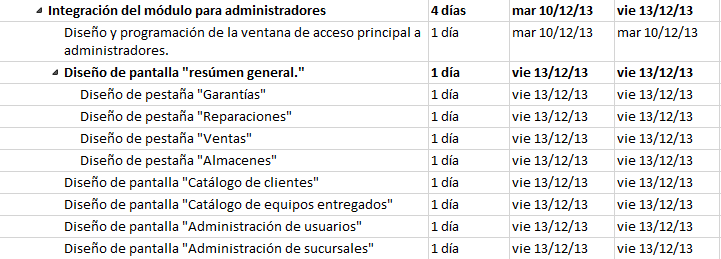


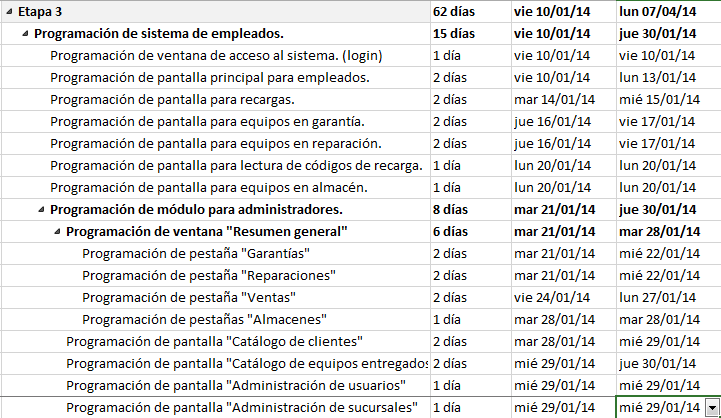


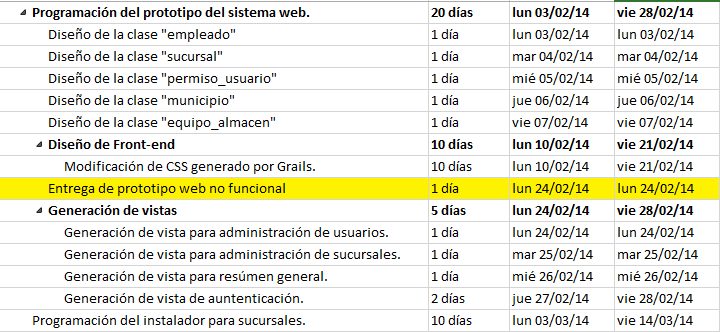


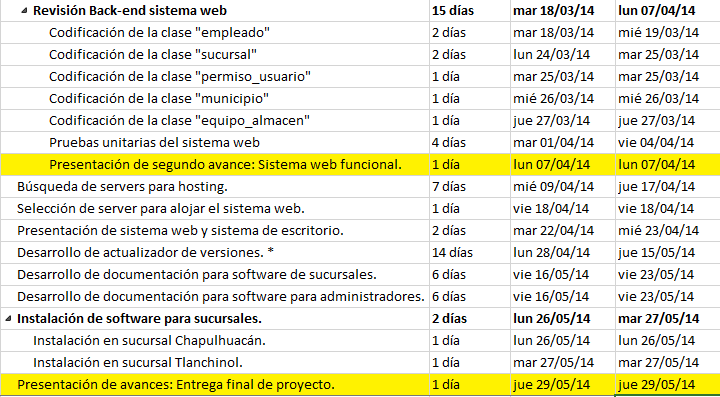












# Anexo B: Ubicación de las sucursales



Los municipios marcados con rojo son las ubicaciones en las que se encuentran las sucursales de las tiendas de celulares.

Donde:

**018** Chapulhuacán

**049** Pisaflores

**062** Tepehuacán de Guerrero

**073** Tlanchinol

# Anexo C: Manual de usuario “Sistema para empleados”

# Referencias

Alejandra Bautista Perea, L. A. (Noviembre de 2006). Desarrollo de un sistema de información para el control de artículos de las unidades médicas rurales en el programa IMSS-OPORTUNIDADES. Pachuca de Soto, Hidalgo, Mexico.

Clauda Ávila Arteaga, M. R. (Diciembre de 2005). *Biblioteca Digital UAEH.*

*Codehaus*. (27 de Octubre de 2013). Obtenido de http://groovy.codehaus.org/Differences+from+Java

*Grails*. (27 de 10 de 2013). Obtenido de http://grails.org/doc/latest/guide/introduction.html

Granados, L. A. (Enero de 2012). Desarrollo Ágil de Aplicaciones Web con Grails Framework. Caso de estudio: PROMEP-UAEH. Mineral de la Reforma, Hidalgo, México.

Henry F. Korth, A. S. (2002). *Fundamentos de bases de datos.* Madrid: Mc Graw Hill.

*Introducing Groovy*. (27 de 10 de 2013). Obtenido de http://www.oracle.com/technetwork/articles/java/groovy-1695411.html

Loizides, A. (10 de Febrero de 2014). Development of a SaaS Inventory Management System. Finlandia, Suecia.

*Master Magazine - Significado y definición de UML*. (21 de 10 de 2013). Obtenido de http://www.mastermagazine.info/termino/7006.php

*OMG UML: Introduction To OMG's Unified Modeling Language™ (UML®)*. (12 de 02 de 2013). Obtenido de http://www.omg.org/gettingstarted/what\_is\_uml.htm

Patel, P. B. (2008). “MOBILE STORE MANAGEMENT SYSTEM”. San Diego, California, Estados Unidos.

*TechTerms.com*. (21 de 10 de 2013). Obtenido de http://www.techterms.com/definition/uml

*What's New in Delphi and C++Builder XE*. (27 de Octubre de 2013). Obtenido de http://docwiki.embarcadero.com/RADStudio/XE/en/What's\_New\_in\_Delphi\_and\_C++Builder\_XE

Wikipedia, l. e. (19 de 09 de 2013). Obtenido de http://es.wikipedia.org/wiki/Framework

*Wikipedia, la enciclopedia libre*. (3 de Octubre de 2013). Obtenido de http://es.wikipedia.org/wiki/Modelo\_Vista\_Controlador

*Wikipedia: La enciclopedia libre*. (2 de 10 de 2013). Obtenido de http://es.wikipedia.org/wiki/UML