[](http://www.google.com.mx/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&docid=G9CmGraXWW9TMM&tbnid=6FUsnBtl_yYhiM:&ved=0CAUQjRw&url=http://www.cedesoft.com.mx/Default.aspx&ei=QfOIU8ytAdWcqAb0-IGYDw&psig=AFQjCNElZaMdgm2JOB5N2iSnh8GKM9cKrQ&ust=1401570486255318)

**Universidad Tecnológica del Centro de Veracruz**

**Tecnologías de la información y la comunicación**

**Aplicación Desktop PCMAX**

**Presenta**

**Elizabeth Ramírez Medina.**

**Gustavo Orbezo Hernández.**

**Alejandro Martínez Valdés.**

**INDICE**

[**Capitulo I.** **Antecedentes** 2](#_Toc420551090)

[1.1 Antecedentes de la empresa 2](#_Toc420551091)

[**1.1.1 Nombre de la empresa** 2](#_Toc420551092)

[**1.1.2 Actividad de la empresa** 2](#_Toc420551093)

[**1.1.3 Antecedentes** 2](#_Toc420551094)

[1.2 Antecedentes del proyecto 3](#_Toc420551095)

[1.3 Planteamiento del problema 3](#_Toc420551096)

[1.5Justificación 4](#_Toc420551097)

[1.6 Objetivo General 4](#_Toc420551098)

[1.7 Objetivos Específicos 4](#_Toc420551099)

[1.8 Alcance 5](#_Toc420551100)

[1.9 Limitaciones 5](#_Toc420551101)

[**Capitulo II.** **Herramientas** 6](#_Toc420551102)

[2.1 Eclipse 6](#_Toc420551103)

[**Figura 2.1** 6](#_Toc420551104)

[2.2. Scenbuilder 6](#_Toc420551105)

[Figura 2.2 6](#_Toc420551106)

[2.3 Jafafx 6](#_Toc420551107)

[Figura 2.3 7](#_Toc420551108)

[2.4 Data modeler 7](#_Toc420551109)

[Figura 2.4 8](#_Toc420551110)

[2.5 Postgres 8](#_Toc420551111)

[Figura 2.5 8](#_Toc420551112)

[**Capitulo III.** **Marco teórico** 8](#_Toc420551113)

[3.1 Eclipse 8](#_Toc420551114)

[3.1.1Principales características 9](#_Toc420551115)

[3.2 JavaFX 10](#_Toc420551116)

[3.3 Data modeler 11](#_Toc420551117)

[3.4 Postgres 11](#_Toc420551118)

[3.5. Scenbuilder 12](#_Toc420551119)

[3.6 Metodología 13](#_Toc420551120)

[3.6.1 moprosoft 13](#_Toc420551121)

[3.6.2 Características 13](#_Toc420551122)

[3.6.2 Estructura de MoProSoft 14](#_Toc420551123)

[Figura 3.6.2 14](#_Toc420551124)

[3.6.3 Proceso 14](#_Toc420551125)

[3.6.4 Ciclo de desarrollo 16](#_Toc420551126)

[3.6.5FASE ESPECIFICACIÓN DE REQUERIMIENTOS. 16](#_Toc420551127)

[3.6.6Ventajas del modelo 18](#_Toc420551128)

[3.6.7 Desventajas del modelo 19](#_Toc420551129)

[3.6.7 Diagrama de clases de Microsoft 20](#_Toc420551130)

# **Antecedentes**

**1.1 Antecedentes de la empresa**

**1.1.1 Nombre de la empresa**

Servicio de reparación de equipos de cómputo (PCMAX)

**1.1.2 Actividad de la empresa**

Soporte y mantenimiento a equipos de cómputo así como venta al mayoreo y menudeo de refacciones para el mismo.

**1.1.3 Antecedentes**

La empresa PCMAX se creó en el año 2012 como un centro de soporte y mantenimiento a equipos de cómputo, la empresa ha crecido al grado de ser el referente entre los técnicos locales para el servicio y reparación de equipos de cómputo, así como la compra venta de refacciones y son los distribuidores de teclados para laptops así como pantallas para las mismas en la zona Córdoba Orizaba, anteriormente se ubicaban en av.11 Num.1120 Colonia Las estaciones, Pero ante la demanda de trabajo y el tratar de dar un mejor servicio actualmente se ubican en av. 1 calle 13 interior plaza palmas donde actualmente son el referente entre los técnicos de la ciudad para la compra venta de refacciones y accesorios de computo.

## 1.2 Antecedentes del proyecto

Actualmente la empresa no cuenta con una aplicación para el manejo de sus productos.

**1.3 Planteamiento del problema**

**1.3.1 Descripción de la problemática**

La empresa PCMAX no cuenta actualmente con una aplicación para el manejo de sus productos, ni para el control de las ventas diarias así como la base de datos de sus clientes, por lo tanto se realizara una aplicación acorde a las necesidades, con las limitantes establecidas.

**1.4 Descripción del producto y entregables**

Se desarrollara una aplicación de escritorio, la cual contara con cinco módulos.

**Módulo 1**

El cliente solicita un módulo donde pueda llevar el control de sus ventas diarias donde especifique la fecha de venta, el folio de venta, el nombre del cliente, dirección, teléfono, así como también una descripción del producto junto con el identificador de el mismo y la garantía que se ofrece, así mismo el precio del producto junto con el respectivo IVA si es que aplica y el total de la compra.

**Módulo 2**

El cliente solicita un módulo donde lleve el control de los servicios prestados, ya sea mantenimiento o garantía de los mismos, en el cual se deberán apreciar la fecha, el número de folio del servicio, las especificaciones del equipo, los datos del cliente, si se procede a mantenimiento o garantía, así como la falla detectada si procede y el técnico que recibe el equipo.

**Módulo 3**

El cliente solicita un módulo donde pueda llevar el control de sus clientes, en el cual pueda agregar su RFC, nombre, dirección, teléfono, email, así como un id del cliente para el manejo interno de los mismos.

**Módulo 4**

El cliente solicita un módulo para llevar el control de sus proveedores en el cual pueda mantener los datos de la empresa y el contacto de la misma, llevando un id de proveedor el nombre de la empresa, su dirección, la ciudad donde esta se ubica, teléfono así como su email, aunado a estos datos también los datos de los contactos de la misma con los datos de nombre de contacto, email y teléfono.

**Módulo 5**

El cliente solicita un módulo para el control de los productos que maneja dentro de la empresa, en el cual se agregara un id del producto, la fecha de compra, el proveedor, núm. de parte, núm. de piezas, el precio de compra, precio de venta y la descripción del producto.

**1.5Justificación**

El propósito del proyecto es desarrollar una aplicación de escritorio para el manejo de sus productos y así mantener un control sobre los mismos, con lo cual se proporcionara una herramienta para agilizar

El proyecto beneficiará a la empresa directamente siendo el desarrollo de la aplicación una manera de mejorar su control de productos.

**1.6 Objetivo General**

Desarrollar una aplicación Desktop para la empresa PCMAX en la cual pueda ingresar sus productos en existencia, con lo cual se espera llenar las expectativas del propietario de la misma, con las limitantes establecidas.

**1.7 Objetivos Específicos**

* Analizar los requerimientos para desarrollar de la aplicación.
* Detectar claramente la necesidad del cliente.
* Mostar el corporativo a sus clientes potenciales.
* Conocer la empresa y la historia de la misma.
* Responder a las inquietudes de los clientes potenciales.
* Dar a conocer la misión, la visión y valores de la empresa.

**1.8 Alcance**

El desarrollo de la aplicación Desktop para la empresa PCMAX se hará en 4 meses, delimitando que aplicación será básicamente, para el ingreso de productos y donde se podrá mantener un administrador y un usuario para el manejo de los mismos

El alcance que se pretende este proyecto es desarrollar una aplicación Desktop, donde permita almacenar información referente a los productos, así mismo la aplicación umplirá ciertas funciones durante su implementación por ejemplo:

* Proceso de registro de productos.
* Logueo de usuarios.

En un momento dado esta aplicación en un futuro podrá cumplir más expectativas o más funciones que se vallan requiriendo.

**1.9 Limitaciones**

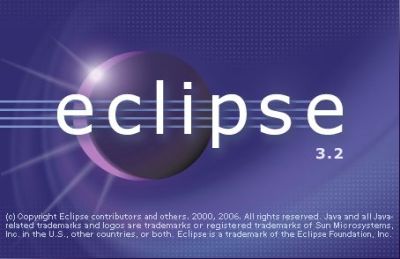
Por la limitante del tiempo, el proyecto no podrá ir más allá de los alcances establecidos, de igual manera no se podrán realizar cambios significativos al mismo, durante el periodo de desarrollo.

Herramientas

# **Herramientas**

## 2.1 Eclipse

Un entorno de desarrollo, integrado, de código abierto y multiplataforma en el cual se ha trabajado con la aplicación de escritorio ya que permite la codificación y la obtención de una interfaz gráfica ya que se le puede montar herramientas de desarrollo de cualquier lenguaje solo implementando los plugins adecuados .



**Figura 2.1**

## 

## 2.2. Scenbuilder

Un editor de interfaz en cual se utiliza en Eclipse para la creación del diseño en javaFX, permite editar archivos FXML, es una herramienta de tipo RAD (desarrollo rápido de aplicaciones), que permite la creación grafica de una interfaz, la cual genera un fichero XML (FXML), que es la vista utilizando el patrón MVC (Modelo-Vista-Controlador).



## Figura 2.2

## 2.3 Jafafx

Esta herramienta nos permite el desarrollo de la aplicación con el lenguaje de programación java, ya que permite la creación de aplicaciones de escritorio.

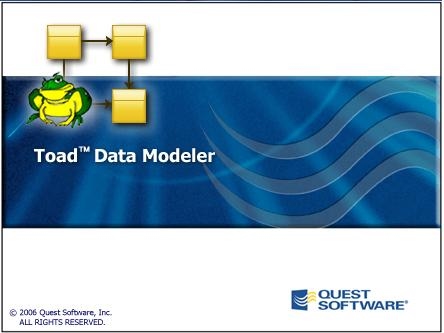


## 

## Figura 2.3

## 2.4 Data modeler

Esta herramienta fue utilizada para el modelado y diseño de bases de datos, en la que se creó el diagrama E/R para después generar el script de creación de objetos BD, una vez normalizada la base de datos.



## Figura 2.4

## 2.5 Postgres

Esta herramienta fue utilizada para realizar nuestra base de datos, una vez normalizadas y creado el diagrama E/R.se  utilizo un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema.



## 

## Figura 2.5

# **Marco teórico**

## 3.1 Eclipse

**Eclipse** es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través deplug-ins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. No tiene en mente un lenguaje específico, sino que es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje **Java** usando el plug-in **JDT** que viene incluido en la distribución estándar del IDE.

Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones.

## 3.1.1Principales características

**Perspectivas, editores y vistas**: en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una pre configuración de ventanas y editores, relacionada entre sí, y que nos permiten trabajar en un determinado entorno de trabajo de forma óptima.

**Gestión de proyectos**: el desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser el código fuente, documentación, ficheros configuración, árbol de directorios,… El IDE nos proporcionará asistentes y ayudas para la creación de proyectos. Por ejemplo, cuando creamos uno, se abre la perspectiva adecuada al tipo de proyecto que estemos creando, con la colección de vistas, editores y ventanas pre configurada por defecto.

**Depurador de código**: se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente nos ayuda a mejorar nuestro código. Para ello sólo debemos ejecutar el programa en modo depuración (con un simple botón). De nuevo, tenemos una perspectiva específica para la depuración de código, la **perspectiva depuración**, donde se muestra de forma ordenada toda la información necesaria para realizar dicha tarea.

**Extensa colección de plug-ins**: están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros. Al haber sido un estándar de facto durante tanto tiempo (no el único estándar, pero sí uno de ellos), la colección disponible es muy grande. Los hay gratuitos, de pago, bajo distintas licencias, pero casi para cualquier cosa que nos imaginemos tenemos el plug-inadecuado.

**Eclipse** es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través deplug-ins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. No tiene en mente un lenguaje específico, sino que es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje **Java** usando el plug-in **JDT** que viene incluido en la distribución estándar del IDE.

Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones.

**Perspectivas, editores y vistas**: en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una pre configuración de ventanas y editores, relacionada entre sí, y que nos permiten trabajar en un determinado entorno de trabajo de forma óptima.

**Gestión de proyectos**: el desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser el código fuente, documentación, ficheros configuración, árbol de directorios,… El IDE nos proporcionará asistentes y ayudas para la creación de proyectos. Por ejemplo, cuando creamos uno, se abre la perspectiva adecuada al tipo de proyecto que estemos creando, con la colección de vistas, editores y ventanas pre configurada por defecto.

**Depurador de código**: se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente nos ayuda a mejorar nuestro código. Para ello sólo debemos ejecutar el programa en modo depuración (con un simple botón). De nuevo, tenemos una perspectiva específica para la depuración de código, la **perspectiva depuración**, donde se muestra de forma ordenada toda la información necesaria para realizar dicha tarea.

**Extensa colección de plug-ins**: están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros. Al haber sido un estándar de facto durante tanto tiempo (no el único estándar, pero sí uno de ellos), la colección disponible es muy grande. Los hay gratuitos, de pago, bajo distintas licencias, pero casi para cualquier cosa que nos imaginemos tenemos el plug-inadecuado.

## 3.2 JavaFX

**JavaFX** es una familia de productos y tecnologías de [Sun Microsystems](http://es.wikipedia.org/wiki/Sun_Microsystems), adquirida por [Oracle Corporation](http://es.wikipedia.org/wiki/Oracle_Corporation), para la creación de [Rich Internet Applications (RIAs)](http://es.wikipedia.org/wiki/Rich_Internet_Application), esto es, aplicaciones web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas. Las tecnologías incluidas bajo la denominación JavaFX son [JavaFX Script](http://es.wikipedia.org/wiki/JavaFX_Script) y [JavaFX Mobile](http://es.wikipedia.org/wiki/JavaFX_Mobile), aunque hay más productos JavaFX planeados.

Las aplicaciones JavaFX pueden ser ejecutadas en una amplia variedad de dispositivos. En su versión (JavaFX 1.3, abril 2010) permite crear aplicaciones de escritorio, para celulares, la Web, TV, consolas de videojuegos, reproductores Blu-ray, entre otras plataformas planeadas. En [octubre de 2011](http://es.wikipedia.org/wiki/Octubre_de_2011) fue lanzada la versión 2.0. Para el desarrollo de aplicaciones JavaFX un lenguaje declarativo, tipado llamado JavaFX Script, además puede integrarse código Java en programas JavaFX. JavaFX es compilado a código Java, por lo que las aplicaciones JavaFX pueden ser ejecutadas en computadores con la máquina virtual de Java instalada ([JRE](http://es.wikipedia.org/wiki/JRE)), o celulares corriendo

JavaFX fue anunciado en la conferencia de desarrolladores   y liberado en diciembre de 2008.

La intención de Sun Microsystems respecto de JavaFX es competir en el espacio que ya ocupan  de, y de  En palabras de "La mayoría de los lenguajes de script están orientados a las páginas web; éste está orientado a las interfaces que son altamente animadas"

## 3.3 Data modeler

La herramienta  Oracle Data Modeler (ODM) permite entre otras cosas, definir un diagrama E/R para después generar el script de creación de objetos en la BD, y también hacer "ingeniería inversa"; es decir, conectarse a un esquema de una BD y seleccionar una serie de tablas, a partir de las cuales genera el diagrama E/R (que se puede guardar como png, svg o pdf). Los diseños se guardan en un fichero XML y un directorio con el mismo nombre que el citado fichero xml (p.e. se podrían guardar en un repositorio SVN). Al contrario que Oracle Designer, no guarda los datos en un repositorio de BD.

## 3.4 Postgres

**PostgreSQL** es un [Sistema de gestión de bases de datos](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos) [relacional](http://es.wikipedia.org/wiki/Modelo_relacional) [orientado a objetos](http://es.wikipedia.org/wiki/Base_de_datos_orientada_a_objetos) y [libre](http://es.wikipedia.org/wiki/Software_libre), publicado bajo la licencia [BSD](http://es.wikipedia.org/wiki/Licencia_BSD).

Como muchos otros proyectos de [código abierto](http://es.wikipedia.org/wiki/Software_de_c%C3%B3digo_abierto), el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma, desinteresada, altruista, libre y/o apoyados por [organizaciones comerciales](http://es.wikipedia.org/wiki/Empresas). Dicha comunidad es denominada el [PGDG](http://es.wikipedia.org/w/index.php?title=PostgreSQL_Global_Development_Group&action=edit&redlink=1)(*PostgreSQL Global Development Group*).

El origen de PostgreSQL se sitúa en el gestor de bases de datos POSTGRES desarrollado en la Universidad de Berkeley y que se abandonó en favor de PostgreSQL a partir de 1994. Ya entonces, contaba con prestaciones que lo hacían único en el mercado y que otros gestores de bases de datos comerciales han ido añadiendo durante este tiempo. PostgreSQL se distribuye bajo licencia BSD, lo que permite su uso, redistribución, modificación con la única restricción de mantener el copyright del software a sus autores, en concreto el PostgreSQL Global Development Group y la Universidad de California. PostgreSQL puede funcionar en múltiples plataformas (en general, en todas las modernas basadas en Unix) y, a partir de la próxima versión 8.0 (actualmente en su segunda beta), también en Windows de forma nativa. Para las versiones anteriores existen versiones binarias para este sistema operativo, pero no tienen respaldo oficial. Para el seguimiento de los ejemplos y la realización de las actividades, es imprescindible disponer de los datos de acceso del usuario administrador del gestor de bases de datos. Aunque en algunos de ellos los privilegios necesarios serán menores, para los capítulos que tratan la administración del SGBDOO será imprescindible disponer de las credenciales de administrador. Las sentencias o comandos escritos por el usuario estarán en fuente monoespaciada, y las palabras que tienen un significado especial en PostgreSQL estarán en negrita. Es importante hacer notar que estas últimas no siempre son palabras reservadas, sino comandos o sentencias de psql (el cliente interactivo de PostgreSQL).

## 3.5. Scenbuilder

##### JavaFX es una familia de productos y tecnologías de Sun Microsystems, adquirida por Oracle Corporation, para la creación de Rich Internet Applications (RIAs), esto es, aplicaciones web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas. Las tecnologías incluidas bajo la denominación JavaFX son JavaFX Script y JavaFX Mobile, aunque hay más productos JavaFX planeados.

la herramienta “Scene Builder” para la creación de forma gráfica de la interfaces, la cual generara como resultado un fichero XML (FXML) que será la clase “Vista” dentro del patrón MVC (Modelo-Vista-Controlador). Las interfaces en JavaFX tambien se pueden realizar a través de código como se hacía en Swing y AWT pero esta forma que se explica en el video tutorial es una forma más fácil de hacer (que para nada significa que sea mejor que las demás).

## 3.6 Metodología

## 3.6.1 moprosoft

METODOLOGA DE PROCESOS PARA LA IDUSTRIA DEL SOFTWARE

Es un Modelo de Procesos para la Industria del Software. Este Modelo sirve para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. MoProSoft es un modelo de calidad que permitirá a la pequeña y mediana empresa de desarrollo de software, el acceso a las prácticas de Ingeniería de Software de clase mundial. Está fundamentado en el modelo SW-CMM, el estándar ISO 9000 y el reporte técnico ISO/IEC TR 15504. La adopción de MoProSoft habilita la obtención de un certificado ISO 9000. Debido a su estructura y diseño, resulta de fácil comprensión y aplicación. La adopción de este modelo permitirá elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad. Moprosoft contiene tres categorías de procesos que corresponden a las capas de Alta Dirección, Gestión y Operación. Lo que este modelo pretende es ayudar a las organizaciones para la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua.

## 3.6.2 Características

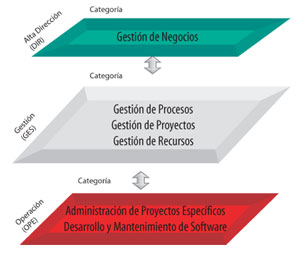
* Es específico para el desarrollo y mantenimiento de software.
* Es sencillo de entender y adoptar.
* Facilita el cumplimiento de los requisitos de otros modelos como ISO 9000:2000, CMM y CMMI.
* Se enfoca a procesos.
* Se le considera práctico en su aplicación, principalmente en organizaciones pequeñas, con bajos niveles de madurez.
* Comprende un documento de menos de 200 páginas que, al compararlo con otros modelos y estándares, lo hace bastante práctico.
* Resulta acorde con la estructura de las organizaciones mexicanas de la industria de software.
* Está orientado a mejorar los procesos, para contribuir a los objetivos de negocio, y no simplemente ser un marco de referencia o certificación.
* Tiene un bajo costo, tanto para su adopción como para su evaluación.

MoProSoft sirve para Mejora la calidad del software producido por la empresa que adopta el modelo. Ofrece servicios con calidad y niveles internacionales de competitividad elevando la capacidad de organización para ofrecer servicios de calidad, implementa un programa de mejora continua nos facilita la selección de proveedores como también permite obtener acceso a las prácticas de ingeniería de software de clase mundial.

## 3.6.2 Estructura de MoProSoft

El modelo pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua. Sintetiza las mejores prácticas en un conjunto pequeño de procesos que abarcan las responsabilidades asociadas a la estructura de una organización que son: la Alta Dirección, Gestión y Operación.

MoProSoft es un modelo integrado donde las salidas de un proceso están claramente dirigidas como entradas a otros; las prácticas de planeación, seguimiento y evaluación se incluyeron en todos los procesos de gestión y administración; por su parte los objetivos, los indicadores, las mediciones y las metas cuantitativas fueron incorporados de manera congruente y práctica en todos los procesos; las verificaciones, validaciones y pruebas están incluidas de manera explícita dentro de las actividades de los procesos; y existe una base de conocimientos que resguarda todos los documentos y productos generados. Veamos a continuación el propósito de los procesos de MoProSoft:



## Figura 3.6.2

## 3.6.3 Proceso

|  |  |  |
| --- | --- | --- |
| Categoría | proceso | Propósito |
| Alta dirección | Gestión de Negocio | Establecer la razón de ser de la organización, sus objetivos y las condiciones para lograrlos, para lo cual es necesario considerar las necesidades de los clientes, así como evaluar los resultados para poder proponer cambios que permitan la mejora continua. Adicionalmente habilita a la organización para responder a un ambiente de cambio y a sus miembros para trabajar en función de los objetivos establecidos. |
| Gestión | Gestión de Procesos | Establecer los procesos de la organización, en función de los procesos requeridos identificados en el Plan Estratégico. Así como definir, planificar e implantar las actividades de mejora en los mismos. |
| Gestión | Gestión de Proyectos | Asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización. |
| Gestión | Gestión de Recursos | Conseguir y dotar a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores, así como crear y mantener la Base de Conocimiento de la organización. La finalidad es apoyar el cumplimiento de los objetivos del Plan Estratégico de la organización. Las actividades de este proceso se apoyan en tres subprocesos: - Recursos humanos y ambiente de trabajo.  - Bienes, servicios e infraestructura.  - Conocimiento de la organización. |
| Operación | Administración de Proyectos Específicos | Establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados. |
| Operación | Desarrollo y Mantenimiento de Software | Realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requerimientos especificados. |

## 3.6.4 Ciclo de desarrollo

El proceso de Desarrollo y Mantenimiento de Software se compone de uno o más ciclos de desarrollo. Cada ciclo está compuesto de las siguientes fases:

**Inicio:** Revisión del Plan de Desarrollo por los miembros del Equipo de Trabajo para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización.

**Requerimientos:** Conjunto de actividades cuya finalidad es obtener la documentación de la Especificación de Requerimientos y Plan de Pruebas de Sistema, para conseguir un entendimiento común entre el cliente y el proyecto.

**Análisis y Diseño:** Conjunto de actividades en las cuales se analizan los requerimientos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado se obtiene la documentación del Análisis y Diseño y Plan de Pruebas de Integración.

**Construcción:** Conjunto de actividades para producir componente(s) de software que correspondan al Análisis y Diseño, así como la realización de pruebas unitarias. Como resultado se obtienen el (los) Componente(s) de software probados.

**Integración y Pruebas:** Conjunto de actividades para integrar y probar los componentes de software, basados en los Planes de Pruebas de Integración y de Sistema, con la finalidad de obtener el Software que satisfaga los requerimientos especificados. Se genera la versión final del Manual de Usuario, Manual de Operación y Manual de Mantenimiento. Como resultado se obtiene el producto de Software probado y documentado.

**Cierre:** Integración final de la Configuración de Software generada en las fases para su entrega. Identificación y documentación de las lecciones aprendidas. Generación del Reporte de Mediciones y sugerencias de mejora.

Para generar los productos de cada una de estas fases se realizan las siguientes actividades:

* Distribución de tareas, se asignan las responsabilidades de cada miembro del Equipo de Trabajo de acuerdo al Plan de Desarrollo.
* Producción, verificación, validación o prueba de los productos, así como su corrección correspondiente.
* Generación del Reporte de Actividades.

## 3.6.5FASE ESPECIFICACIÓN DE REQUERIMIENTOS.

**Descripción:** Se compone de una introducción y una descripción de requerimientos. **Introducción:** Descripción general del software y su uso en el ámbito de negocio del cliente.

**Descripción de requerimientos:**

**\* Funcionales**: Necesidades establecidas que debe satisfacer el software cuando es usado en condiciones específicas. Las funcionalidades deben ser adecuadas, exactas y Seguras.

**\* Interfaz con usuario:** Definición de aquellas características de la interfaz de usuario que permiten que el software sea fácil de entender, aprender, que genere satisfacción y con el cual el usuario pueda desempeñar su tarea eficientemente. Incluyendo la descripción del prototipo de la interfaz.

**\* Interfaces externas:** Definición de las interfaces con otro software o con hardware.

**\* Confiabilidad:** Especificación del nivel de desempeño del software con respecto a la madurez, tolerancia a fallas y recuperación.

\* Eficiencia: Especificación del nivel de desempeño del software conrespecto al tiempo y a la utilización de recursos.

\* Mantenimiento: Descripción de los elementos que facilitarán lacomprensión y la realización de las modificaciones futuras del software.

\* Portabilidad: Descripción de las características del software que permitan su transferencia de un ambiente a otro.

\* Restricciones de diseño y construcción: Necesidades impuestas por el cliente.

\* Legales y reglamentarios: Necesidades impuestas por leyes, reglamentos, entre otros.

**FASE DE ANÁLISIS Y DISEÑO:**

Descripción: Este fase contiene la descripción textual y grafica de la estructura de los componentes de software. El cual consta de las siguientes partes:

Arquitectónica: Contiene la estructura interna del sistema, es decir la descomposición del sistema en subsistemas. Así como la identificación de los componentes que integran los subsistemas y las relaciones de interacción entre ellos.

Detallada: Contiene el detalle de los componentes que permita de manera evidente su construcción y prueba en el ambiente de programación.

**FASE COMPONENTE:** Conjunto de unidades de código relacionadas.

**Software:** Sistema de software, destinado a un cliente o usuario, constituido por componentes agrupados en subsistemas, posiblemente anidados. Configuración de **Software:** Conjunto consistente de productos de software, que incluye:

• Especificación de Requerimientos.

• Análisis y Diseño.

• Software.

• Registro de Rastreo.

• Plan de Pruebas de Sistema.

• Reporte de Pruebas de Sistema.

• Plan de Pruebas de Integración.

• Reporte de Pruebas de Integración.

• Manual de Usuario.

• Manual de Operación.

• Manual de Mantenimiento.

**Manual de Usuario:** Documento electrónico o impreso que describe la forma de uso del software con base a la interfaz del usuario. Éste deberá ser redactado en términos comprensibles a los usuarios.

**Manual de Operación:** Documento electrónico o impreso que contenga la información indispensable para la instalación y administración del software, así como el ambiente de operación (sistema operativo, base de datos, servidores, etc.). Éste deberá ser redactado en términos comprensibles al personal responsable de la operación.

**Manual de Mantenimiento:** Documento electrónico o impreso que describe la Configuración de Software y el ambiente usado para el desarrollo y pruebas (compiladores, herramientas de análisis y diseño, construcción y pruebas). Este deberá ser redactado en términos comprensibles al personal de mantenimiento.

**Reporte de Actividades:** Registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como:

• Tiempo de producción, de corrección, de verificación y de validación, Defectos encontrados en verificación, validación o prueba,

• Tamaño de productos.

**Lecciones Aprendidas:** Registro de mejores prácticas, problemas recurrentes y experiencias exitosas en la solución de problemas, encontrados en un ciclo de desarrollo y mantenimiento. Reporte de Mediciones y Sugerencias de Mejora: Registro que contiene:

\* Mediciones de los indicadores del proceso de Desarrollo y Mantenimiento de Software

. \* Sugerencias de mejora al proceso de Desarrollo y Mantenimiento de Software (métodos, herramientas, formatos, estándares, etc.).

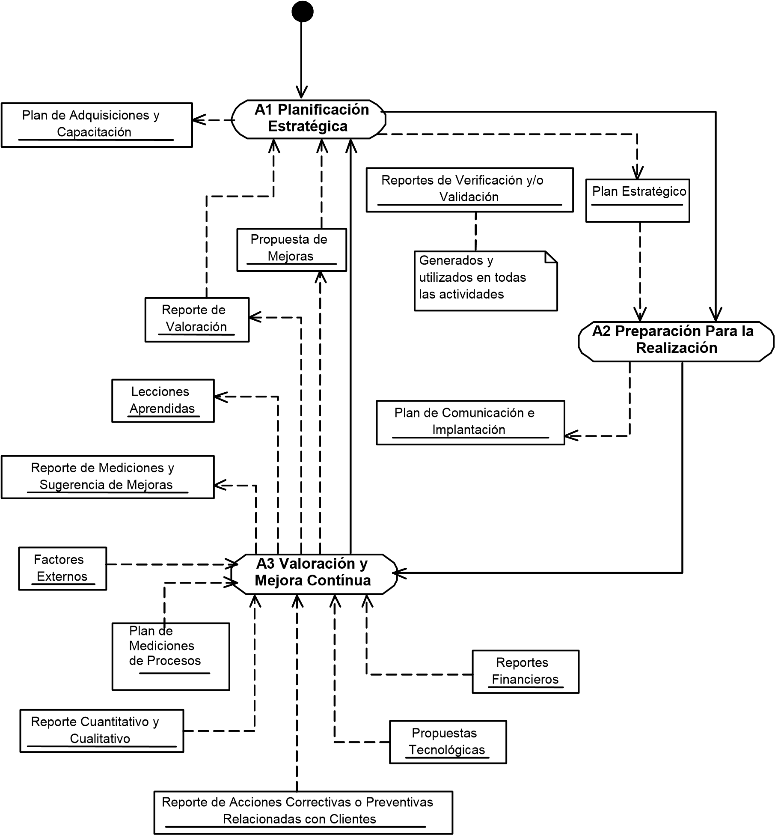
## 3.6.6Ventajas del modelo

* Al tener prácticas integradas, que abarcan desde la gestión de negocio hasta el desarrollo y mantenimiento de software, las empresas tendrían mayor control sobre su desempeño en el mercado.
* El costo de la incorporación del nuevo personal podría disminuir si se enfocan la educación y la capacitación a un modelo único.
* Las empresas pequeñas, al seguir procesos similares, podrían asociarse con mayor facilidad para afrontar proyectos de mayor envergadura.
* La exportación de servicios de software de las empresas mexicanas.

## 3.6.7 Desventajas del modelo

* Define actividades de manera muy general.
* Para asegurar la calidad de un producto y un proceso se requiere CMMI.
* El 33% de las prácticas no cubiertas de definir e implementar como lo son Administración de Configuración (CM) Y Medición y Análisis (MA).
* Evaluaciones formales constantes.
* No comprensible para los modelos ISO 9000: 2000.
* Mejora los procesos Orientado al Objeto del negocio.
* Proyectos para largos plazos.

## 3.6.7 Diagrama de clases de Microsoft



# Bibliografía

(s.f.).

cimes, r. (s.f.). fundamentos de base de datos . En remes, *fundamentos de base de datos.* pirson.

Elmasri, r. (s.f.). fundamentos de base de datos.

fundamentos de base de datos . (s.f.). En R. Elmasri, *fundamentos de base de datos* (5 ed.).

LAWRENCE PFLEEGER, S. (s.f.). Ingeniería de software . En S. LAWRENCE PFLEEGER, *Ingeniería de software .* Pearson Prentice Hall, año 2002.

PRESSMAN, R. S. (s.f.). Ingeniería de software, un enfoque práctico. En R. S. PRESSMAN, *Ingeniería de software, un enfoque práctico* (sexta ed.). Ingeniería de software, un enfoque práctico.

SOMMERVILLE, I. (s.f.). Ingeniería de software. En I. SOMMERVILLE, *Ingeniería de software* (sexta ed.). Pearson Addison Wesley, año 2005 (Libro auxiliar).