



**Universidad Tecnológica del Centro de Veracruz.**

**Programa educativo de:**

***TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN.***

**Nombre del proyecto:**

***“DISEÑO DE UNA APLICACIÓN DESKTOP”***

**Presentan:**

* **Espinoza Gómez Héctor Rene.**
* **Fernández Vásquez Luis Ángel.**
* **Machuca Rojas Francisco Javier.**

**Grado y grupo: 6A TICSI DZ.**

**Periodo cuatrimestral: Mayo-Agosto 2015.**



**Índice.**

[1 Resumen 4](#_Toc426580630)

[2 Summary 5](#_Toc426580631)

[3 Antecedentes de la empresa 6](#_Toc426580632)

[4 Descripcion de la problemática 7](#_Toc426580633)

[5 Descripción del producto 7](#_Toc426580634)

[5.7 Entregables 10](#_Toc426580635)

[5.7.1 Entregables para el equipo de desarrollo 10](#_Toc426580636)

[5.7.2 Entregables para el cliente 11](#_Toc426580637)

[6 Objetivos 12](#_Toc426580638)

[7 Justificación 12](#_Toc426580639)

[8 Alcance y limitaciones 13](#_Toc426580640)

[9 Herramientas 15](#_Toc426580641)

[10 Marco Teórico 16](#_Toc426580642)

[10.9.1 Introducción 32](#_Toc426580643)

[10.9.2 Ventajas 32](#_Toc426580644)

[10.9.3 Desventajas 32](#_Toc426580645)

[10.9.4 Fases 33](#_Toc426580646)

[10.9.4.1 Inicio 33](#_Toc426580647)

[10.9.4.2 Requerimientos 33](#_Toc426580648)

[10.9.4.3 Análisis y diseño 34](#_Toc426580649)

[10.9.4.4 Construcción 34](#_Toc426580650)

[10.9.4.5 Integración y pruebas 35](#_Toc426580651)

[10.9.4.6 Cierre 35](#_Toc426580652)

[10.9.5 Diagrama de la metodología MoProSoft 35](#_Toc426580653)

[10.11 Diagrama de clases 38](#_Toc426580654)

[10.12 Diagramas de secuencia 40](#_Toc426580655)

[10.13 Diagramas de relación de entidad 41](#_Toc426580656)

[11 Bibliografía 42](#_Toc426580657)

**Índice de figuras.**

[Figura 11.11‑1 Scene builder 15](file:///D:\Documentos\team%204\1_DEFINICION\DESCRIPCION_DEL_PROYECTO.docx#_Toc426580025)

[Figura 11.11‑2 Paqueteria javaFX visualizacion 17](file:///D:\Documentos\team%204\1_DEFINICION\DESCRIPCION_DEL_PROYECTO.docx#_Toc426580026)

[Figura 11.11‑3 Eclipse , entorno de desarrollo 19](file:///D:\Documentos\team%204\1_DEFINICION\DESCRIPCION_DEL_PROYECTO.docx#_Toc426580027)

[Figura 11.11‑4 Gestor de base de datos PostgreSQ 23](#_Toc426580028)

[Figura 11.11‑5 Servidor bit-bucket 24](file:///D:\Documentos\team%204\1_DEFINICION\DESCRIPCION_DEL_PROYECTO.docx#_Toc426580029)

[Figura 11.11‑6 Cliente- Source-tree 27](file:///D:\Documentos\team%204\1_DEFINICION\DESCRIPCION_DEL_PROYECTO.docx#_Toc426580030)

[Figura 11.11‑7 Logo del lenguaje de programacion Java 28](#_Toc426580031)

[Figura 11.11‑8 Toad modeler vista 30](file:///D:\Documentos\team%204\1_DEFINICION\DESCRIPCION_DEL_PROYECTO.docx#_Toc426580032)

[Figura 11.11‑9 Metodologia MoProSoft 34](file:///D:\Documentos\team%204\1_DEFINICION\DESCRIPCION_DEL_PROYECTO.docx#_Toc426580033)

Resumen

El presente proyecto consiste en elaborar una aplicación de escritorio para la empresa “ROSSMA”, esta se desarrollara en las herramientas de Eclipse con paquetería JavaFX, Escene Builder, utilizando el lenguaje Java para su desarrollo e implementación, así mismo la aplicación cuenta con los módulos que la empresa requiere y son implementados sobre una base de datos que almacenara los registros que en esta puedan ingresarse. Se anexan de igual forma información de la empresa, problemática, requisitos entre otras cosas que son de vital importancia para el buen desarrollo de la aplicación.

Se indica la información de las herramientas a utilizar, metodología empleada para el desarrollo de la aplicación, resultados que salgan durante el desarrollo, para el conocimiento del cliente a quien se le está ofreciendo el producto.

Palabras clave: Aplicación de escritorio, Java.

Summary

This project is to develop a desktop application For " ROSSMA " this version be developed in Eclipse Tools package JavaFX , Scene Builder, using the Java language para do Development and Implementation , likewise the application has the modules the base version requires an Envelope and are implemented will store data records can be entered in this . Likewise appended Company Information, Issues, requirements among other things that the son of fundamental importance for the good Application Development.

Information indicated Tools use one, used Methodology for Application Development, Results departing During Development for Knowledge of customer who is offering the product.

Antecedentes de la empresa

ROSSMA la empresa fabricante y distribuidora de uniformes, que cuenta con una gran variedad en uniformes escolares, dicha empresa se encuentra ubicada en la Calle.15, Av.1 y 2 #2 Altos. La empresa nace el 23 de Enero del año 2011, comenzando con la elaboración en uniformes escolares. Posteriormente la empresa fabricante y distribuidora de uniformes ROSSMA, fue creciendo y hoy en día ofrece a sus clientes diversos productos de calidad los cuales van de ser uniformes escolares, de trabajo, ropa deportiva, camisas y trajes de vestir.

Actualmente cuenta con distintos proveedores que suministran los materiales esenciales para la elaboración de sus uniformes, la situación económica de dicha empresa es muy buena ya que la ventaja de esta empresa es que se encuentra ubicada en un lugar muy céntrico de la ciudad de Córdoba, sus ingresos varían un poco debido a las temporadas escolares.

Descripcion de la problemática

La empresa ROSSMA cuenta con una aplicación que muestra ciertos módulos que le interesa a la empresa para una mejora administración, además cuenta con un panel para iniciar sesión, se encuentra de igual manera que no cuenta con una base de datos normalizada, y los módulos que cuenta realiza registros simples.

Para ello nosotros tenemos la propuesta de una restructuración a la aplicación, además de una mejora hacia la base de datos para la mejor organización de la información que maneja, con esto igual pueden generarse reportes a lo que la empresa requiere para saber los movimientos que se realizan dentro de la aplicación, al mejorar la forma de los datos para llenar en los módulos permitirá que esta lleve una buena administración de sus clientes, proveedores, presentándolos en una forma clara al momento de registrarlos, ingresando solo los datos que a la empresa le interesa y mostrándolos en una ventana la lista de los clientes y proveedores que se llevan registrados, en productos un registro similar mostrando los precios que se manejan en la empresa y mostrándolos en un stock que lleve el total de las piezas existentes ,mínimas y máximas, las ventas que genere mostraran el total de la venta e imprimirán un ticket para el cliente de la empresa, y al mismo tiempo generara un descuento a las existencias de los productos mostrados en stock, así como un modelo de usuarios que contara con administrador y usuario normal , para el uso de una persona capaz de administrar bien la aplicación y el otro para los empleados que puedan generar una venta y/o visualizar productos.

Descripción del producto

Se realizara mantenimiento y adecuaciones a la aplicación de escritorio, con el fin de que cuente con una ventana para poder iniciar sesión de 2 formas diferentes, la primera de administrador que podrá visualizar todos los módulos y podrá hacer inserción, modificación y/o eliminaciones, tal como lo prefiera el administrador, y la segunda con un usuario normal, que solo podrá visualizar el módulo de ventas, generar una venta, el módulo de clientes por si llegara a realizarse un registro de un cliente nuevo.

Modulo Clientes

Se adecuara un módulo de clientes que tendrá 3 paneles donde se registraran clientes con razón social ya sea moral o físico, en caso de tratarse de una persona moral y los campos a llenar son nombre de la empresa, RFC, país, estado, municipio, calle, exterior, interior, colonia, localidad, CP, correo, teléfono, celular y un campo extra por si tiene radio, además de contar con un contacto para los datos de la persona con la que se están haciendo los enlaces con la empresa; En caso de que sea física serán los campos nombre, apellido paterno, apellido materno RFC, país, estado, municipio, calle, exterior, interior, colonia, localidad, CP, correo, teléfono, celular y un campo extra por si tiene radio, tendrá una lista de los clientes que tiene la empresa, con un buscador para encontrar un registro los más rápido posible. El módulo de clientes se encuentra implementado para tener un registro bien organizado de los clientes que tiene la empresa y catalogarlos para su mayor funcionamiento en la búsqueda de los registros.

Modulo Proveedores

El módulo de proveedores contara con una ventana que registrara nombre, RFC, país, estado, municipio, calle, exterior, interior, colonia, localidad, CP, correo, teléfono, celular y un campo extra por si tiene radio, y una visualización de los registros que se tienen dentro de una tabla con un buscador para la búsqueda más efectiva y rápida de un registro. Este módulo esta implementado para la obtener mejores resultados y la perdida nula de la información de los proveedores que maneja la empresa.

Modulo Productos

El módulo de productos cuenta con los datos necesarios para la empresa y su buena organización, tiene una unidad de medida, descripción del producto, existencia actual, existencia mínima, existencia máxima y una categoría de los productos para tenerlos organizados conforme la empresa labora a diferentes ramos de prendas, el módulo de productos esta implementado para la correcta organización y visualización real de lo que tiene la empresa en existencia o no, de igual forma cuenta con una tabla de los productos que se tienen registrados con un buscador automatizando las búsquedas de los registros dentro de todos los productos que maneja la empresa.

Modulo Stock

El modulo stock es la visualización de los productos que se tienen registrados dentro de la empresa y que se dan de alta en el módulo de productos, está solo tiene una lista con un buscador por si se desea conocer las existencias que se tienen de aquel producto que se está manejando.

Modulo Ventas

El módulo de ventas cuenta con una ventana que permite traer el nombre del cliente a quien se le está vendiendo los productos, datos de los productos que se están vendiendo y el usuario y/o empleado que está realizando la ventana en su momento, generan una cantidad de productos vendidos, las unidades en la que se maneja aquel producto, la fecha y hora que se realizó la venta , el importe de los productos y el IVA unitario de aquel producto vendido, generara un ticket que permita visualizar al cliente de la empresa los datos anteriores, al igual que el guardar la venta en una tabla que podrá visualizar el administrador de la aplicación.

Modulo usuarios

El modulo usuarios permite un registro de los usuarios que tendrán una interacción con la aplicación , se registra un nombre de usuario, una contraseña, la categoría de administrador o usuario , un estatus, así como datos esenciales para poder encontrar al usuario en caso de no poder ingresar con su usuario.

## Entregables

### Entregables para el equipo de desarrollo

Entre los productos entregables que se darán al equipo de desarrollo, para el aprendizaje de los mismos, y las formas que se estuvieron trabajando, entre los productos desarrollados se encuentran los siguientes:

* **Documento de equipo de trabajo:** En este se mencionan los roles que desempeñaran cada integrante del equipo, y los significados de cada rol que se maneja para el desarrollo del proyecto.
* **Glosario:** Documento donde se mencionan las palabras propias del negocio, para el conocimiento de los desarrolladores.
* **Especificación de requisitos:** Son los requerimientos que indican las necesidades del cliente y sirven para el desarrollo de la aplicación.
* **Diagrama de casos de uso:** Son las interpretaciones graficas de los requerimientos y le sirven al analista para poder discernir la funcionalidad de la aplicación.
* **Diagrama de clases:** Estas sirven para la interpretación de las clases, los atributos que tendrán y los métodos que podrán realizar cada una de las clases.
* **Diagrama de secuencia:** Sirve para que el analista pueda analizar mejor la funcionalidad y interacción que tendrá con las demás clases.
* **Diagrama de entidad – relación:** Sirve para guiarse sobre la base de datos que se trabajara, y las cardinalidad mismas que tendrá.
* **Manual de repositorio**: Manual donde se indica el servidor y cliente con el que se trabaja para el desarrollo de documentación , productos y aplicación.
* **Lecciones aprendidas:** Documento que guarda todas las experiencias que se vivieron dentro del equipo de trabajo, en cuanto la generación de problemas y el uso adecuado para dar solución a estos mismos.
* **Descripción del proyecto:** Documento donde se encuentran partes de la empresa y el conocimiento del cliente herramientas, metodología con la que se trabaja entre otras partes.

**Los productos se deberán entregar el día 08 de agosto a las 14:00 horas en la empresa Rossma, ubicada en la avenida 2 y 4 calle 15.**

### Entregables para el cliente

Los productos entregables que se entregan al cliente son para el conocimiento del producto que requiere.

* **Archivo ejecutable de la aplicación:** Es la aplicación que se requiere, contiene los requisitos que la empresa “Rossma” ha decidido adquirir.
* **Manual de usuario:** Es el manual donde se integran paso por paso el manejo total que la aplicación llega a tener.

Los documentos que se mencionan anteriormente se entregaran el día 10 de agosto a las 13:00 hrs.

Objetivos

General

Restructurar una aplicación de escritorio utilizando herramientas tales como ECLIPSE, SCENEBUILDER Y LIBRERÍA JAVAFX, para la administración y agilizacio de los procesos que mantiene la empresa Rossma para el control de sus productos y ventas , el cual maneja clientes, proveedores , ventas , cumpliendo las reglas de seguridad para al acceso de datos que esta maneja, además una interfaz amigable para el usuario y fácil de manejar.

Específicos

* Tener un buen registro de clientes, productos y proveedores, validando los datos y un ingreso de datos esenciales para la empresa.
* Tener un buen control y ejecución de ventas, imprimiendo en un ticket los productos que se están vendiendo.
* Realizar entrada y salida de productos que tiene la empresa, refiriéndose a la salida de mercancía, visualizando los cambios en el módulo de stock.

Justificación

Para la empresa ROSSMA es de alta importancia la modificación de la versión anterior de la aplicación desktop para poder realizar un buen registro de clientes, con administración clara y un buen registro de productos y proveedores así como ventas que se registren.

La adecuación de la aplicación para la empresa es un tema de suma importancia, considerando que ya no tendrían un descontrol en sus ventas,como en la anterior versión en la cual el modulo que se requeria no realizaba dichas acciones, sabrían a qué hora de entrada y salida tendrían los productos, así como la mercancía, cuanta y que tipo de mercancía seria, es una herramienta fundamental en los registros de ventas así como de compra de mercancía, los beneficios serían los siguientes:

* Mayor control de los clientes y proveedores, agilizando la búsqueda dentro de los registros, la validación de los campos ayudarían a no perder la información que s esta guardando.
* Los productos dentro de la empresa estarán organizados dentro de la aplicación, a lo cual el cliente podrá saber qué cantidad de productos están saliendo y se encuentran en existencia, con descuentos por las ventas que se generen.
* La aplicación podrá generar ventas, con esto se tendrá un conteo de las ventas realizadas llevadas por un folio.
* Contará con diferentes usuarios, esto le servirá al dueño de la empresa catalogar a sus empleados y a él o una persona encargada de llevar la aplicación, limitando a los usuarios normales la visualización de lo que está manejando la empresa.

Los beneficios que presenta una aplicación como la que se está modificando busca generar una mayor agilidad a la empresa y pocas perdidas en lo que se respecta a los productos que maneja.

Alcance y limitaciones

Alcances

Este proyecto pretende realizar registros exitosos con validaciones mas concretas a la hora de registrar los datos en la base de los clientes, proveedores y productos que se manejan dentro de la empresa, que en la anterior versión no podía realizar, se podrán generar ventas y hacer descuentos en cuanto a las piezas que se encuentran en stock, al momento de realizar un venta, y se podrán registrar los detalles de venta de acuerdo a las ventas que se generan , generándolos con un identificar hora y fecha del momento que se esta misma se realizo.

La aplicación contara con 2 usuarios distintos, un administrador que podrá tener acceso a todos los módulos desarrollos para la aplicación, y el usuario normal o empleado podrá visualizar productos, agregar clientes y/o generar una venta, mas allá de esto no podrá hacer ediciones en precios de los productos y/o visualizaciones de los usuarios que están registrados dentro del producto realizado.

El usuario podrá cambiar de puerto para la conexión a la base de datos, esto aparecerá en el acceso que mostrara la pantalla de inicio, y solo podra tener acceso el administrador para realizar dichos cambios.

Tendrá validaciones de campo, para la evitar la pérdida de datos a ingresar, mejorando con respecto a la anterior versión la correcta inserción de ellos.

Limitaciones

* Esta aplicación será solo para escritorio, no se podrá usar en celulares, tabletas y/o algún otro dispositivo portátil ya que la aplicación va dedicada hacia el gestión amiento de la empresa y punto de venta.
* Solo podrá ser ejecutable para un equipo de cómputo.
* La aplicación será desarrollada a base de las características del equipo de cómputo de la empresa.
* No realizara las compras hacia los proveedores.

Herramientas

La aplicación es desarrollada con un lenguaje Java y con ayuda de una paquetería JavaFX se crearan la funcionalidad de los modulos, Un programa llamado Eclipse Luna para el entorno a desarrollar el código y Scene Builder para el entorno grafico de la aplicación , todo con el fin de agilizar el desarrollo de la aplicación.

Tendra una base de datos en Postgres SQL que servirá para almacenar la información de la empresa , esta se mostrara dentro de la aplicación a desarrollar y el cliente no podrá tener acceso directo a la base de datos , para evitar perdida de información.

Características del equipo de cómputo.

El equipo de cómputo donde la aplicación se ejecutara es:

* Intel® Core™ i3-4160 Processor (Dual Core, 3MB, 3.60GHz w/HD4400 Graphics)
* Windows® 7 Professional, 64-bit, Español (Incluye licencia y medio de Windows 8.1 Pro)
* 4 GB1 SDRAM DDR3 a 1600 MHz
* Disco Duro SATA de 500GB 7200 RPM de 3.5"
* Gráficos integrados Intel®

A lo cual indicamos que no hay problema alguno con la aplicación que se desarrollara , podrá ejecutarse bien y no abra problemas en cuanto la funcionalidad.

Marco Teórico

A continuación se presentara la información de las herramientas y conceptos que se utilizan con anterioridad, para la comprensión del usuario.

Scene buider

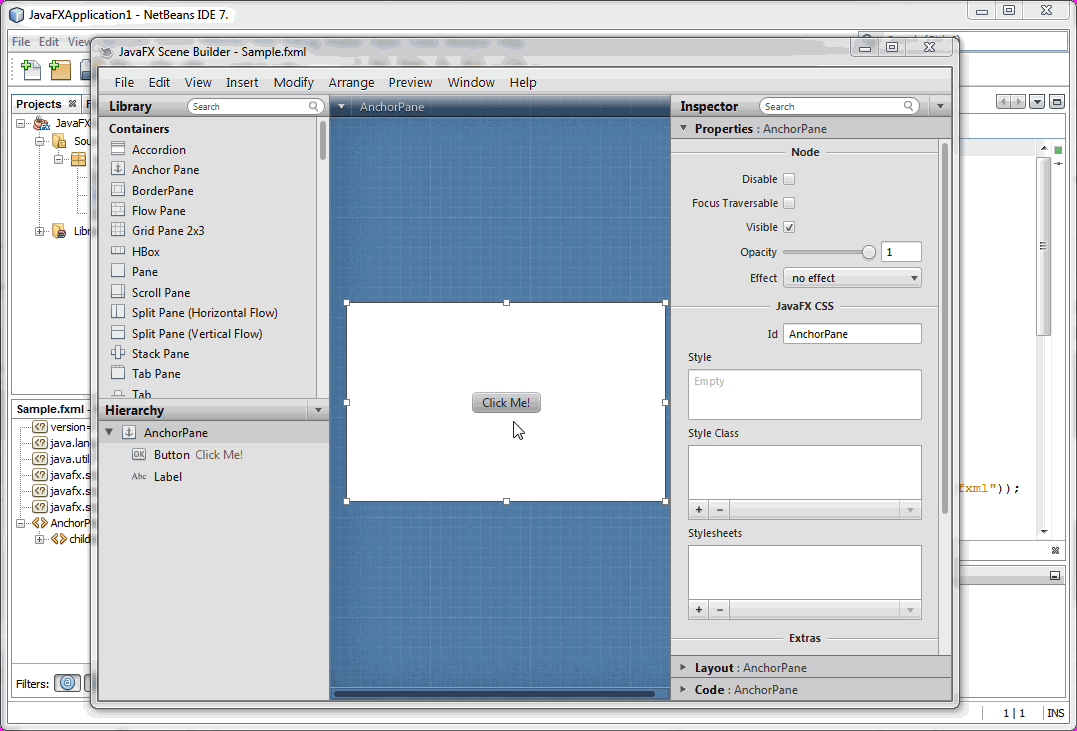
scene JavaFX Builder es una herramienta de diseño visual que permite a los usuarios diseñar rápidamente interfaces de usuario de aplicaciones JavaFX, sin necesidad de programación. Los usuarios pueden arrastrar y soltar los componentes de interfaz de usuario a un área de trabajo, modificar sus propiedades, solicitar las hojas de estilo, y el código FXML para la disposición de que están creando se genera automáticamente en segundo plano. El resultado es un archivo FXML que luego se puede combinar con un proyecto de Java mediante la unión de la interfaz de usuario a la lógica de la aplicación

Figura 11.11‑1 Scene builder

Ilustración 1 Scene Builder.

Paquetería Java FX

La plataforma JavaFX es la evolución de la plataforma del cliente Java diseñada para permitir a los desarrolladores de aplicaciones para crear y desplegar aplicaciones ricas de Internet (RIAs) que se comportan de forma consistente en múltiples plataformas. Construido sobre la tecnología Java, la plataforma JavaFX proporciona un rico conjunto de API de gráficos y contenidos multimedia de alto rendimiento con aceleración por hardware de gráficos y los motores de los medios de comunicación que simplifican el desarrollo de aplicaciones basadas en datos de la empresa cliente.

Historia de JavaFX

En la conferencia JavaOne 2007, Sun Microsystems presentó la plataforma JavaFX para ayudar a los desarrolladores de contenido y desarrolladores de aplicaciones crear aplicaciones ricas en contenido para dispositivos móviles, escritorios, televisores y otros dispositivos de consumo. La oferta inicial consistió en la plataforma JavaFX Mobile y el lenguaje JavaFX Script. Varios comunicados públicos fueron entregados después del anuncio inicial, la versión 1.3 fue lanzado el 22 de abril de 2010.

Después de la adquisición de Oracle de Sun Microsystems, Oracle ha anunciado durante la conferencia JavaOne 2010 que el apoyo al lenguaje JavaFX Script se suspendería. Sin embargo, también se anunció que el API de JavaFX Script será portado a Java y sería puesto en libertad como parte de los productos JavaFX 2.0. Este anuncio significa que las capacidades JavaFX estará disponible para todos los desarrolladores de Java, sin la necesidad de aprender un lenguaje de programación nuevo. Con este anuncio, Oracle se ha comprometido a JavaFX el medio ambiente de primera para aplicaciones de cliente enriquecido.

Características de JavaFX

\* La creación de aplicaciones JavaFX se hace a través del lenguaje JavaFX Script.

\* Las aplicaciones de escritorio pueden ejecutarse en Windows XP, Vista y 7, Mac OS, GNU/Linux y OpenSolaris.

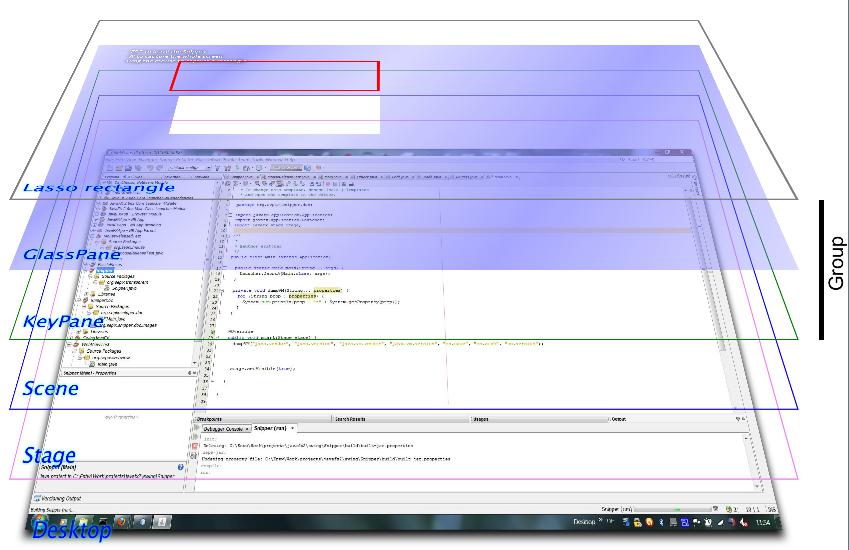
\* Al estar integrado con el JRE (Java Runtime Environment), las aplicaciones de JavaFX se pueden ejecutar en cualquier navegador que posean el JRE, como así también móviles que dispongan de Java ME.

Figura 11.11‑2 Paqueteria javaFX visualizacion

Eclipse

Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plug-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.

Principales características

Perspectivas, editores y vistas: en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una preconfiguración de ventanas y editores, relacionadas 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 preconfigurada 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.

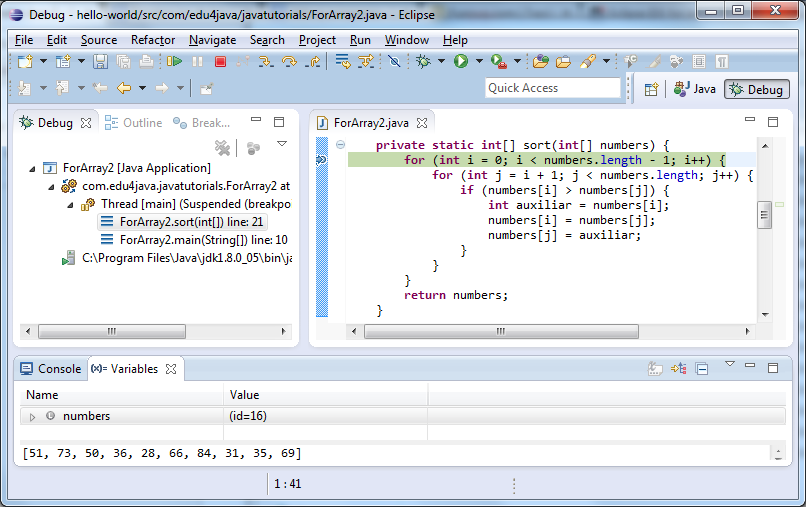
Extensa colección de plug-ins: están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros. Al haber sido un estándar de facto durante tanto tiempo (no el único estándar, pero sí uno de ellos), la colección disponible es muy grande. Los hay gratuitos, de pago, bajo distintas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

Figura 11.11‑3 Eclipse , entorno de desarrollo

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Características

La última serie de producción es la 9.3. Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzo hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez a el sistema.

características más importantes y soportadas por PostgreSQL:

Generales

* Es una base de datos 100% ACID
* Integridad referencial
* Tablespaces
* Nested transactions (savepoints)
* Replicación asincrónica/sincrónica / Streaming replication - Hot Standby
* Two-phase commit
* PITR - point in time recovery
* Copias de seguridad en caliente (Online/hot backups)
* Unicode
* Juegos de caracteres internacionales
* Regionalización por columna
* Multi-Version Concurrency Control (MVCC)
* Multiples métodos de autentificación
* Acceso encriptado via SSL
* Actualización in-situ integrada (pg\_upgrade)
* SE-postgres
* Completa documentación
* Licencia BSD
* Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
* Programación / Desarrollo
* Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programacion, entre otros PL/pgSQL (similar al PL/SQL de oracle), PL/Perl, PL/Python y PL/Tcl
* Bloques anónimos de código de procedimientos (sentencias DO)
* Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc
* Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido, ...)
* APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.
* SQL
* SQL92,SQL99,SQL2003,SQL2008
* Llaves primarias (primary keys) y foráneas (foreign keys)
* Check, Unique y Not null constraints
* Restricciones de unicidad postergables (deferrable constraints)
* Columnas auto-incrementales
* Indices compuestos, únicos, parciales y funcionales en cualquiera de los metodos de almacenamiento disponibles, B-tree, R-tree, hash ó GiST
* Sub-selects
* Consultas recursivas
* Funciones 'Windows'
* Joins
* Vistas (views)
* Disparadores (triggers) comunes, por columna, condicionales.
* Reglas (Rules)
* Herencia de tablas (Inheritance)
* Eventos LISTEN/NOTIFY



Figura ‑ Gestor de base de datos PostgreSQ

Bitbucket

es un servicio de alojamiento basado en la web para proyectos que utilizan el sistema de control de versiones Mercurial y Git. Fue lanzado en el año 2008 por la empresa Atlassian Software y esta escrito en Python mediante el framework web Django.

Bitbucket ofrece cuentas gratuitas y comerciales. Las gratuitas cuentan con número ilimitado de repositorios privados y cinco usuarios, aunque tienen la opción de llegar a un total de 8 si invitas a 3 a unirse al servicio. Los repositorios privados van desde 10$ al mes por 10 usuarios hasta planes ilimitados. Estos no se muestran en las páginas de perfil, es decir, si un usuario sólo tiene repositorios privados el sitio web indicara que no tiene.

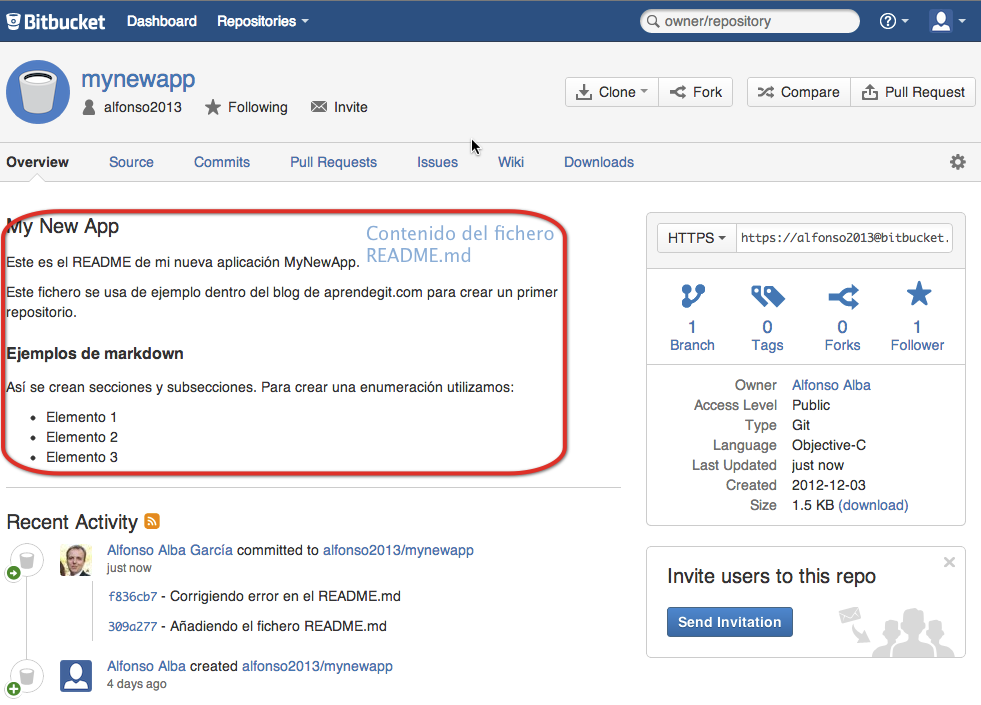
Entre sus características más llamativas están su integración con Jira, un bug tracking system muy usado actualmente. Gracias a esto se puede seguir cada commit de un problema en este popular gestor de incidencias. También permite controlar las actividades por branch, agregar keys de Google Analytics y seguir otros repositorios mediante la opción watch. Tienen un cliente gratuito para Windows y Mac llamado SourceTree.

Figura 11.11‑5 Servidor bit-bucket

SourceTree

· Ayuda mercurial y de Git en un uso - es común actualmente parecer los proyectos que eligen diversos sistemas de control de versión, así que SourceTree le deja trabajar con los depósitos mercuriales y de Git sin el uso cambiante. El interfaz de la base es igual para ambos, pero se ha adaptado la ayuda de modo que el comportamiento (y la terminología) sea constantes con las características y las capacidades únicas de cada uno.

· Refresco automático - con SourceTree, usted no necesita golpear el `restaura' para ver los últimos cambios; SourceTree supervisa sus carpetas y automáticamente cambios de trabajo de las exhibiciones inmediatamente.

· Señales con el resumen del estado - usted puede marcar una dirección de la Internet carpetas de trabajo con frecuencia usadas y ver de un vistazo cuál es su estado, incluso antes de que usted los abre.

· Opinión avanzada del estado del archivo - la opinión del estado del archivo le demuestra el estado de su carpeta del proyecto inmediatamente, con una multiplicidad de maneras de explorar y de perforar abajo en ella. Elegir de plano, de árbol y de 3 opiniones de la columna, o utilizar una caja de la búsqueda que filtre mientras que usted mecanografía para encontrar lo que usted necesita rápidamente. En Git, usted tiene la opción para utilizar la zona de espera del `' si usted trabaja esa manera, o la oculta lejos si usted prefiere un acercamiento más simple.

· Opinión de la historia - demostraciones de la opinión de la historia de SourceTree usted el gráfico de cambios usted ha venido esperar de un sistema de control distribuido de versión, demostrando ramas, etiquetas y una exhibición inmediata de detalles y de cambios del archivo mientras que usted navega. Incluso demuestra le el efecto de confiar sus cambios actuales en el gráfico, así que le puede visualizar lo que usted está haciendo más fácilmente.

· Herramientas externas de Diff y de la fusión - SourceTree viene preconfigurado para utilizar FileMerge, fusión de Araxis, DiffMerge o KDiff3 para exhibir diferencias y la fusión de la resolución está en conflicto, y otras herramientas se pueden utilizar con una cierta configuración del usuario.

· Rama, fusión, etiqueta - SourceTree proporciona le un interfaz de gran alcance pero simple a ocuparse de las ramas múltiples del desarrollo, y a marcar con etiqueta de versiones específicas.

· Colaborar fácilmente con otros - SourceTree incluye toda la funcionalidad que usted necesita colaborar con los equipos cerrados o con una audiencia más ancha (tal como fuente abierta); los sistemas alejados se pueden configurar fácilmente dentro de SourceTree y de los cambios intercambiados por otros en varias formas.

· Proceso del remiendo - una tarea común especialmente en proyectos de la fuente abierta es generar y consumir remiendos. SourceTree le da las herramientas para hacer que fácilmente - incluso explorará un archivo de remiendo y calculará hacia fuera los parámetros requeridos para aplicarlo, y se apoyan qué opciones (e.g. importando un lleno confían o simplemente aplicándolo a su carpeta de trabajo).

· Resolución de conflicto - mercurial y Git ser muy bueno en la combinación, pero los conflictos son a veces inevitables. SourceTree provee de usted las herramientas que usted necesita trabajar a través de esos casos, usar las herramientas de la fusión externa si está deseado.

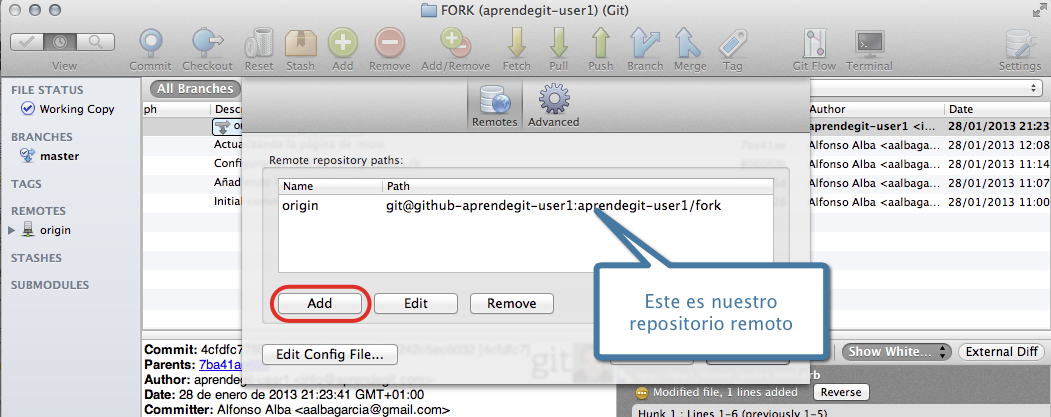


Figura 11.11‑6 Cliente- Source-tree

Lenguaje Java

Java es un lenguaje de programación que fue desarrollado a principios de los años 90's

El creador de este exitoso programa fue Patrick Naughton, ingeniero de Sun MicroSystems que trabajo con sus compañeros de trabajo llamando al proyecto "El proyecto verde". Él y otros ingenieros trabajaron sin descansar durante 18 meses.

Intentaban crear un lenguaje de programación fácil de usar y de utilizar, al principio consideraban el más útil C++ pero no todos estuvieron de acuerdo y lo encontraban inadecuado como Gosling.

Gosling intentaba modificar C++, pero al final termino abandonando el proyecto y sus cambios y empezo desde cero al que llamo Oak, similar a C,C++. Luego Oak cambió el nombre a Java porque resultaba que ya existia un lenguaje llamado Oak, se dice que Java es originario de Asia, lo nombraron así mientras tomaban café.

En Agosto del 1991 Java ya tenía sus propios programas. Para 1992, el equipo o grupo desarrollo un sistema llamado Star7, parecido a PDA, cuyo nombre viene de la combinacion del "Proyect Green" o "Proyecto Verde" como mencionamos antes, que permitía a los usuarios responder al teléfono desde cualquier lugar.

En Septiembre de 1992 se presentan los prototipos de bajo nivel del sistema, que continuaban desarrollando, incluyendo el sistema operativo. Y se termina a finales de Septiembre como el "Proyecto Verde"

De 1993 a 1994 el equipo de Naughton lanzó nuevas propuestas al mercado, las mismas que se fueron dando en el sistema operativo base. Sin embargo, este proyecto fracasó por sus ambiciones de ganar ofertas. Afortunadamente el Proyecto Verde coincidió con el fenómeno mundial WEB

Naughton procedió a la construcción del lenguaje Java y en Septiembre de 1994 se termina el prototipo de HotJava cuando se hace la demostración y se reconoce el potencial de Java y aceptan el proyecto.



Figura ‑ Logo del lenguaje de programacion Java

Toad Data Modeler™

le ayuda a crear modelos de datos de alta calidad. Esta herramienta de modelado de datos le permite implementar fácilmente cambios exactos a las estructuras de datos a través de más de 20 plataformas diferentes.

Este multi-plataforma de software de modelado de bases de datos ofrece las siguientes funcionalidades:

* Le permite construir modelos de datos lógicos y físicos de alta calidad
* Le permite comparar los modelos y luego sincronizarlos
* Hace generando compleja SQL / DDL rápido y fácil
* Le permite crear y modificar secuencias de comandos
* Le permite invertir y reenviar ingeniero ambas bases de datos y sistemas de almacenamiento de datos
* Le permite conectar múltiples bases de datos de forma nativa y al mismo tiempo; estos incluyen Oracle, Sybase, MySQL, SQL Server, PostgreSQL, DB2, Ingres, MS Access y más
* Le ayuda a crear estructuras de base de datos de calidad, además de proporcionar la documentación en múltiples plataformas
* Le permite generar nuevas estructuras de bases de datos o realizar cambios en los modelos existentes de forma automática
* Le permite construir modelos de relación de entidad complejos (lógico y físico)
* Le permite fácilmente revertir y bases de datos de ingeniería hacia adelante
* Los permisos de importación de modelos de secuencias de comandos SQL
* Le permite migrar las estructuras a través de plataformas de bases de datos
* Le proporciona informes detallados sobre las estructuras de bases de datos existentes
* Le permite agregar datos lógicos a sus diagramas para personalizar modelos
* Facilita la sincronización de un modelo con la base de datos existente mediante "ALTER" generación de scripts (Oracle y SQL Server) y el modelo se fusionan características
* Le permite ver las relaciones y el trabajo con atributos y columnas de entidades y tablas vinculadas
* Permite el acceso a datos rápido y sencillo por lo que le permite crear carpetas de proyecto a los modelos de tienda, los requerimientos del negocio, especificaciones y otros archivos

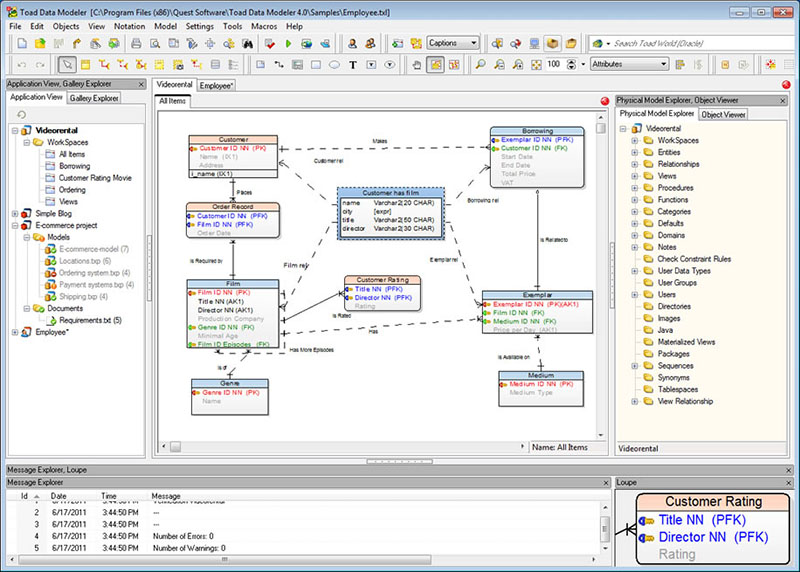


Figura 11.11‑8 Toad modeler vista

Metodología MOPROSOFT

### Introducción

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.

### Ventajas

• Su adopción no es costosa.

• Especifico para el desarrollo y mantenimiento del software.

• Practico de aplicar en organizaciones pequeñas.

• Orientado a mejorar los procesos para contribuir a los objetivos del negocio.

• Aplicable como norma Mexicana.

• Basada en normas ISO

• Facilita la compresión del modelo.

• Simplifica la relación entre el modelo de procesos y la organización.

• Cuenta únicamente con 9 procesos evitando la fragmentación que se presenta en otros modelos.

• Capacidad organizacional de gestión de procesos y proyecto.

### Desventajas

• Evaluaciones formales constantes

• No es práctico ni fácil de usar.

• No es compresible para los modelos ISO 9000: 2000

• Mejora de procesos orientado al objetivo del negocio.

• Proyectos para largos plazos.

\*\* Existen empresas en México que implementaron y se certificaron en MoProSoft han logrado obtener niveles 3 y 5 en CMMI; lo que muestra que puede ser adoptado como un estrategia para crecer.

### 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, además de una investigación del giro de la empresa para comprender como se controla.

Productos generados en esta fase:

* Plan de desarrollo: cronograma, equipo de trabajo.
* Reporte de actividades: bitácora, minutas.

#### 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.

Productos generados en esta fase:

* Manual de usuario (preliminar).
* Plan de pruebas del sistema.
* Especificación de requerimientos en base a la norma IEEE 830
* Plan de pruebas unitarias

#### 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 de análisis y diseño y el plan de pruebas de integración.

Productos generados en esta fase:

* Plan de pruebas de integración
* Reporte de validación
* Reporte de actividades
* Diagramas: casos de uso, clases, secuencia, entidad-relación, relacional

#### Construcción

Conjunto de actividades para producir componentes de software que correspondan al análisis y diseño. Así como la realización de pruebas unitarias.

Como resultado se obtiene los componentes de software probados.

Productos generados en esta fase:

* Distribución de las tareas a los integrantes del equipo, según sus roles
* Modificación y construcción de software
* Reporte de actividades

#### 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 final

Como resultado se obtiene el software probado y documentado.

#### Cierre

Integración final de la configuración del software generada en las fases anteriores para su entrega. Identificación y documentación de las lecciones aprendidas. Generación del reporte de mediciones y sugerencias de mejora.

### Diagrama de la metodología MoProSoft

Figura 11.11‑9 Metodologia MoProSoft

Diagramas de caso de uso

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen qué es lo que debe hacer el sistema, pero no cómo.

Caso de uso

Un caso de uso describe, —desde el punto de vista de los actores—, un grupo de actividades de un sistema que produce un resultado concreto y tangible.

Los casos de uso son descriptores de las interacciones típicas entre los usuarios de un sistema y ese mismo sistema. Representan el interfaz externo del sistema y especifican qué requisitos de funcionamiento debe tener este (recuerde, únicamente el qué, nunca el cómo).

Cuando se trabaja con casos de uso, es importante tener presentes algunas secillas reglas:

Cada caso de uso está relacionado como mínimo con un actor

Cada caso de uso es un iniciador (es decir, un actor)

Cada caso de uso lleva a un resultado relevante (un resultado con «valor intrínseco»)

Los casos de uso pueden tener relaciones con otros casos de uso. Los tres tipos de relaciones más comunes entre casos de uso son:

<<include>> que especifica una situación en la que un caso de uso tiene lugar dentro de otro caso de uso

<<extends>> que especifica que en ciertas situaciones, o en algún punto (llamado punto de extensión) un caso de uso será extendido por otro.

Generalización que especifica que un caso de uso hereda las características del «super» caso de uso, y puede volver a especificar algunas o todas ellas de una forma muy similar a las herencias entre clases.

Actor

Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.

Los actores no representan a personas físicas o a sistemas, sino su rol. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores. Por ejemplo, una persona que proporciona servicios de atención telefónica a clientes y realiza pedidos para los clientes estaría representada por un actor «equipo de soporte» y por otro actor «representante de ventas».

Descripción de casos de uso

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso.

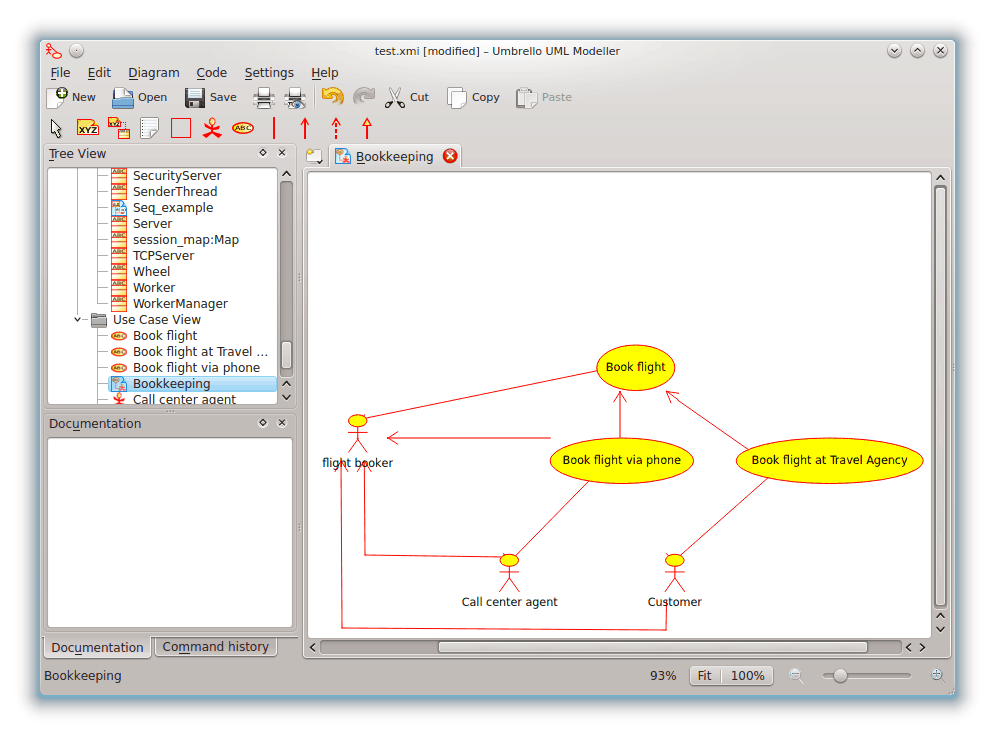


Imagen . Casos de uso

## Diagrama de clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

Clase

Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos (cada objetos tiene el suyo propio). En ocasiones se utiliza el término «tipo» en lugar de clase, pero recuerde que no son lo mismo, y que el término tipo tiene un significado más general.

Las clases están representadas por rectángulos, con el nombre de la clase, y también pueden mostrar atributos y operaciones de la clase en otros dos «compartimentos» dentro del rectángulo.

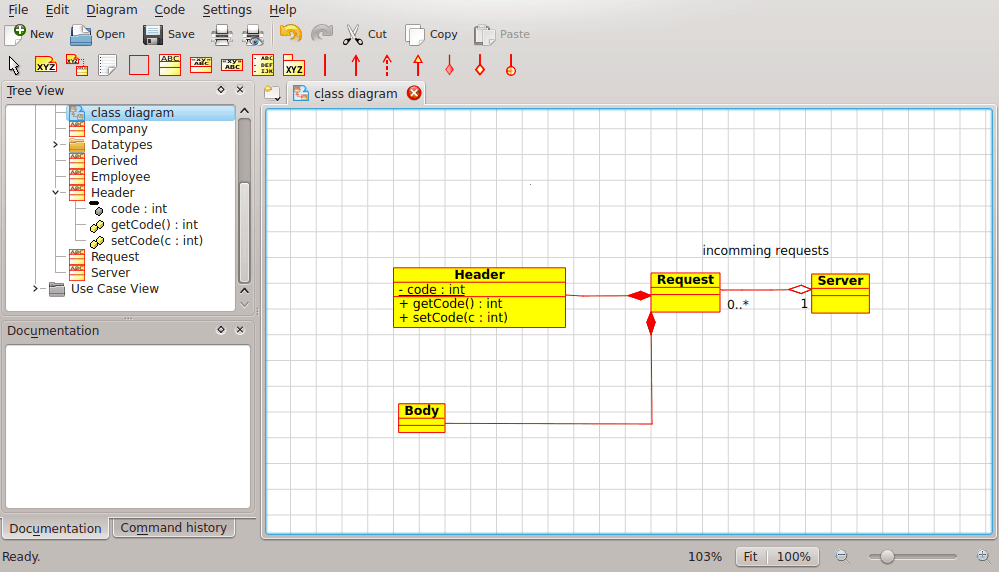


Imagen . Diagramas de clase

## Diagramas de secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

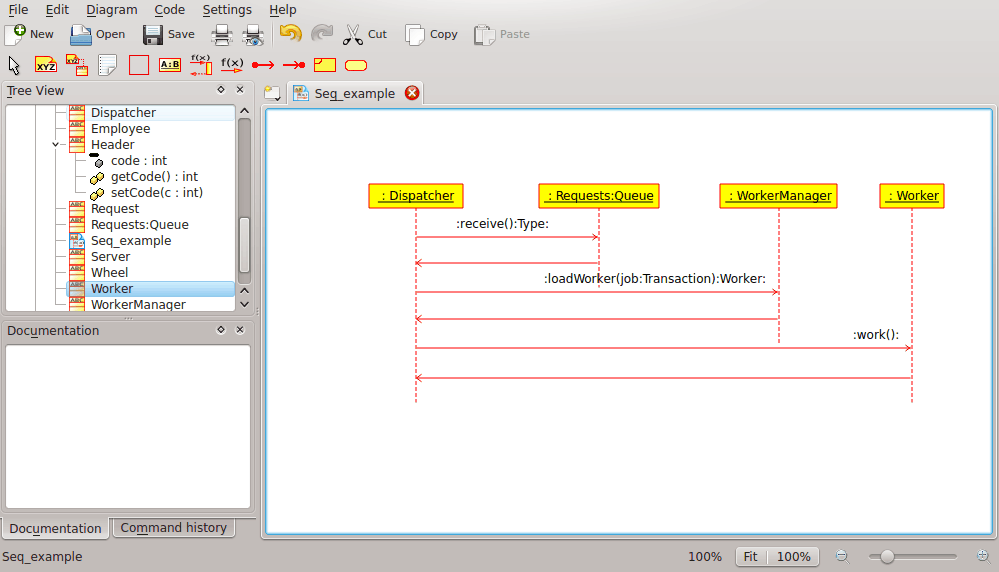


Imagen . Diagramas de secuencia

## Diagramas de relación de entidad

Los diagramas de relaciones de entidad (diagramas ER) muestran el diseño conceptual de las aplicaciones de bases de datos. Representan varias entidades (conceptos) en el sistema de información y las relaciones y restricciones existentes entre ellas. Una extensión de los diagramas de relaciones de entidad llamado «diagramas de relaciones de entidad extendida» o «diagramas de relaciones de entidad mejoradas» (EER), se utiliza para incorporar las técnicas de diseño orientadas a objetos en los diagramas ER.

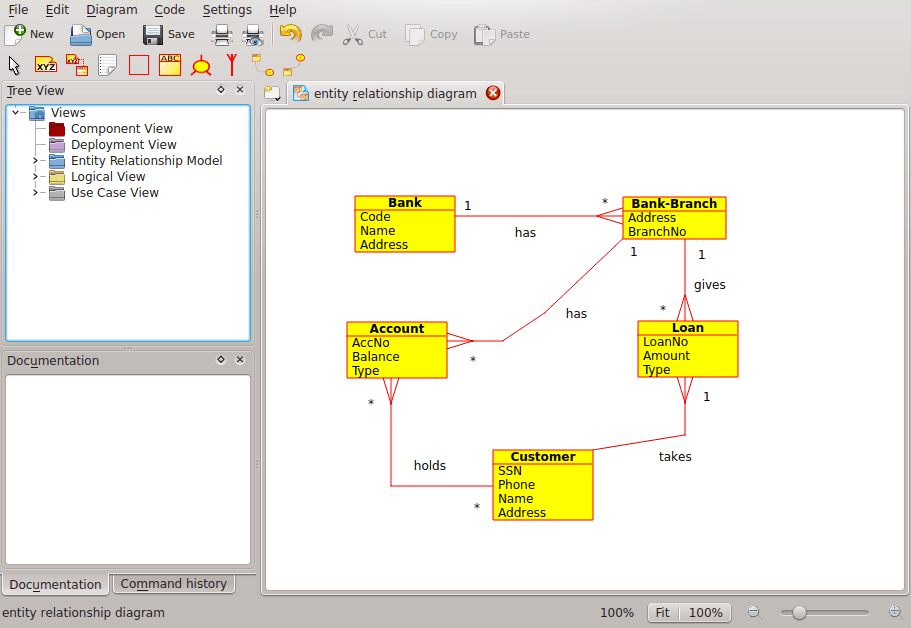


Imagen . Diagrama entidad - Relacion

# Bibliografía

(2005-2014). Obtenido de http://www.misrespuestas.com/que-es-php.html

Arráez Rodriguez, J., & Advani, K. (Mayo de 2011). *¿Qué es vitualbox y como se usa vitualbox?* Obtenido de http://drcaos.com/sobre-nosotros/

*Culturacion*. (s.f.). Obtenido de http://culturacion.com/

*Culturacion*. (s.f.). Obtenido de http://culturacion.com/que-es-apache/

*Culturacion*. (s.f.). Obtenido de http://culturacion.com/que-es-apache/

*Definicion.de*. (2008-2014).

*Definicion.de*. (2008-2014). Obtenido de http://definicion.de/html/

Grady Booh, I. J. (2000). *El proceso unificado de desarrollo de software.* Madrid: Person Education.

Gutierrez Mora, S. (4 de Marzo de 2004). *Conceptos básicos de FTP*. Obtenido de http://www.uam.es/servicios/ti/servicios/ftp/ftp.html

*Instituto Nacional del Derecho de Autor*. (2014). Recuperado el 4 de Noviembre de 2014, de http://www.indautor.gob.mx/?navegador2=%271%27&valor=

Luna Esquivel, R. (10 de Noviembre de 2010). *La historia del DNS*. Obtenido de http://faqhosting.com.ar/la-historia-de-los-dns/

Martinez, R. (2005-2014). *Mis respuestas.com*. Obtenido de http://www.misrespuestas.com/que-es-php.html

Martinez, R. (2009-2013). *PostgreSQL-es*. Obtenido de http://www.postgresql.org.es/sobre\_postgresql

Martinez, R. (2009-2014). *PostgresSQL-es*. Obtenido de http://www.postgresql.org.es/sobre\_postgresql

Martinez, R. (s.f.). *PostgresSQL-es*. Obtenido de http://www.postgresql.org.es/sobre\_postgresql

*mis respuestas.com*. (2005-2014). Obtenido de http://www.misrespuestas.com/que-es-php.html

*Mis respuestas.com*. (2005-2014). Obtenido de http://www.misrespuestas.com/que-es-php.html

Prados, C. (26 de Abril de 2011). *La importancia de la virtualización de servidores para la empresa*. Obtenido de http://blog.trevenque.es/sistemas/la-importancia-de-la-virtualizacion-de-servidores-para-la-empresa/

Smuller, J. (2001). *Aprendiendo UML en 24 horas.* California: Prentice Hall.

UML. (2008). *https://docs.kde.org/stable4/es/kdesdk/umbrello/uml-elements.html*. Obtenido de UML: https://docs.kde.org/stable4/es/kdesdk/umbrello/uml-elements.html

Zamora Carmona, D. (2013). *¿Qué es una página web?* Obtenido de http://www.quees.info/que-es-una-pagina-web.html