

Descripción del proyecto

BLOCKBUSTER



García Rodríguez Samuel

Paez Vázquez José Eduardo

Pérez Sánchez Oscar

Venegas Morgado José María

10 de septiembre de 2015

TICSI 4°A

Equipo 1

**Contenido**

[**1 Introducción** 2](#_Toc430643271)

[**2** **Descripción de productos y entregables** 3](#_Toc430643272)

[**3 Objetivos** 4](#_Toc430643273)

[**4 Justificación** 5](#_Toc430643274)

[**5 Alcances y limitaciones** 5](#_Toc430643275)

[**6 Herramientas** 6](#_Toc430643276)

[**7 Marco Teórico** 6](#_Toc430643277)

[**8 Bibliografía** 16](#_Toc430643278)

# **1 Introducción**

* 1. **Antecedentes de la empresa**

La primera tienda Blockbuster abrió el 19 de octubre de 1985 en la ciudad de Dallas, Texas, Estados Unidos. El fundador de la compañía fue el empresario norteamericano de origen holandés Wayne Huizenga.

Para 1987 la compañía tenía ya 17 tiendas en los Estados Unidos.

La expansión de la cadena Blockbuster inicio en 1989 al abrir su primera tienda fuera de Estados Unidos en Londres Inglaterra, continuando esta con la apertura de 12 sucursales en Canadá en 1990.

En 1992 se abren las primeras tiendas en Australia, Chile, España y Venezuela.

En 1993 la empresa abrió su tienda número 3000 en Nueva York.

La compañía fue adquirida en 1994 por Viacom Inc. por 8.4 billones de dólares.

En 1995 abre nuevas sucursales en Italia, Argentina y Nueva Zelanda.

En 1996 la Blockbuster Entertainment Corporation cambió su nombre a Blockbuster Inc., y sus tiendas entonces llamadas Blockbuster Video, pasaron a ser solo Blockbuster; sin embargo, las tiendas anteriores a este año conservaron su antiguo nombre.

Para 1998 ya contaba con operaciones en Perú, Tailandia, Colombia, Portugal, Brasil, Panamá, El Salvador, Uruguay, Taiwán y Dinamarca.

En 2002 Blockbuster adquirió Movie Trading Company, una compañía radicada en Dallas, Texas dedicada a comprar, vender y promocionar películas y videojuegos. En ese mismo año la compañía adquirió a Gamestation, una cadena inglesa de tiendas de cómputo y consolas de videojuegos con 64 sucursales.

En 2004 Blockbuster introdujo el concepto de suscripción vía internet conocido como Blockbuster Online; además inicio en 450 tiendas de Estados Unidos y el en extranjero el concepto de store-in-store (tienda dentro de otra), con sus Game Rush, dedicados a la renta y venta especializada de videojuegos.

Actualmente Blockbuster está presente en 24 países alrededor del mundo y emplea a más de 128,000 personas. Pero desgraciadamente la falta de adaptación a las nuevas tecnologías digitales, y los elevados costes de personal y alquiler de locales, les hizo perder mucha cuota de mercado a favor de los pequeños videoclubs de barrio y las máquinas 24 horas de alquiler. La piratería muy contrariamente de lo que se dijo en su momento sólo supuso el 10% de sus pérdidas, todo lo demás fue debido a su mala gestión y a su falta de adaptación al mercado europeo. Todo esto provocó que cerrara todas sus tiendas en España en marzo de 2006.

El 23 de septiembre de 2010, Blockbuster se acogió al Capítulo 11 de protección de quiebra, debido a las pérdidas difíciles de $ 900 millones en deuda, y la fuerte competencia de Red Box, Netflix, y video bajo demanda, la piratería de películas bajadas de Internet y la venta callejera habían aumentado en los últimos años. Blockbuster dijo que mantendrá sus 3.300 tiendas en funcionamiento por el momento. Sin embargo, se espera que cerca de más de 900 tiendas cierren a finales de 2010. Movie Gallery / Video Hollywood acogió al Capítulo 11 de bancarrota a principios de este año del cierre y liquidación de Blockbuster.

* 1. **Descripción de la problemática**

BlockBuster es una empresa dedicada al entretenimiento para ser más específico renta y venta de películas como negocio principal reconocida en todo el país, dicha empresa cuenta con una sucursal ubicada en la ciudad de Córdoba Veracruz, esta sucursal además de tener que llevar el control de inventario, entrega, socios y demás de sus películas, cuentan con una sección especializada a los videojuegos la cual entiende aspectos como compras, rentas, ventas, apartados, consolas, accesorios, suvenires, debido a que la empresa es nacionalmente conocida se vieron en la necesidad de implementar desde hace tiempo una solución informática para facilitar el control de los aspectos mencionados anteriormente, más sin embargo aunque el sistema con él cuenta la empresa para manejar esta sección de la misma ha sido funcional hasta ahora se han notado algunas características que son mejorables como la interfaz gráfica de usuario, manejo de promociones y los catálogos de productos, debido a las carencias de dicho sistemas en los aspectos mencionados con anterioridad los usuarios del sistema (empleados) han hecho mención de que puede ser mejorado.

**Solución**

Desarrollar una aplicación de escritorio echa a medida que cumpla con las funciones principales que el sistema actual ha brindados hasta el momento, las cuales son el manejo de existencia de los productos del área de videojuegos lo que conlleva los aspectos como compra, venta y renta, implementando así nuevas tecnologías para poder mejorar algunas carencias que el sistema actual tiene como la interfaz de usuario, el manejo de las promociones y la manera de manejar las consultas y modificaciones de los productos ya existentes.

# **Descripción de productos y entregables**

|  |  |
| --- | --- |
| **Entregables** | **Descripción** |
| Diccionario de base de datos | Modelo que especifica el valor de los campos de la base de datos, con la finalidad de asignar el tipo de dato correcto en la base de datos. |
| Maquetación | Elaboración de bocetos de las interfaces de usuario, creados en papel o en herramientas informáticas para el diseño. |
| Formularios | Interfaces de diseño y uso eficiente. En las cuales el usuario interactuará con las funciones del software. |
| Manual de usuario | Documento con información recabada sobre el contenido y funcionamiento del software, con la finalidad de capacitar al usuario para el uso del mismo. |
| Manual de instalación | Documento con pasos e instrucciones a seguir para realizar la instalación optima del software entregado al cliente. |
| Punto de venta Game-Rush | Software administrativo para el control de una tienda dedicada a la compra y veta de videojuegos, accesorios y productos gamer. |
| Modulo Empleados | Modulo donde se lleva acabo el registro, consulta, edición y eliminación de empleados. Siendo almacenados en una base de datos. |
| Modulo Productos | Modulo donde se registran, consultan, editan y eliminan los productos del punto de venta, almacenando los cambios en una base de datos. |
| Modulo Venta | Modulo donde se realizan, registran y consultan las ventas realizadas por los empleados. |
| Modulo Compras | Modulo donde se realizan, registran y consultan la compras realizadas. |
| Modulo Clientes | Modulo dónde se registran, consultan y editan clientes con membresía a la tienda. |

# **3 Objetivos**

**3.1 General**

Desarrollar una aplicación de escritorio con una interfaz gráfica mejor a la ya existente que permita a la empresa BlockBuster gestionar los productos que maneja en el área videojuegos, de una manera más eficiente para así mismo agilizar sus procesos de compra, venta y renta.

**3.2 Específicos**

1. Levantar la información necesaria, mediante entrevistas realizada al encargado de la sucursal BlockBoster y observación directa de los procesos que se realizan en el lugar.
2. Analizar la información recopilada, seleccionándola y organizándola detalladamente, determinando así el alcance y las necesidades de la aplicación a desarrollar
3. Diseñar la aplicación con la información y las especificaciones dadas por parte de la empresa BlockBoster.
4. Elaborar las pruebas o correcciones necesarias, conjuntamente con el personal de BlockBoster, con la finalidad de verificar que cumpla con las expectativas deseadas.
5. Instalar la aplicación para que pueda ser utilizada, realizando la respectiva capacitación de los usuarios finales.

# **4 Justificación**

Se desarrollara esta aplicación de escritorio debido a que la Empresa BlockBoster cuenta con un sistema el cual hasta el momento les ha sido de gran ayuda para su manejo de productos en el área videojuegos, pero con el inconveniente de que el sistema carece de algunos aspectos, por lo que se busca hacer una mejora en la interfaz gráfica la cual le haga sentirse al usuario más cómodo al estar haciendo uso del sistema, al igual se pretende mejorar la manera en que se gestionan los productos para así agilizar los procesos de registro de las ventas , compras y renta de los mismos.

# **5 Alcances y limitaciones**

* 1. **Alcances**
* La aplicación contara con un control de cuentas de usuario para administrar y de esta forma poder reportar las operaciones hechas por los empleados.
* El software contara con distintos módulos tales como rentas, ventas, apartados, compras algunas de estas categorías cuentan con una cantidad extra de módulos para realizar distintas operaciones como por ejemplo de consulta.
* Se manejara un panel de administrador en el cual se podrán generar reportes y obtener información alusiva a los empleados, ventas, rentas y compras.
* Se creara un módulo para el registro de los clientes, mediante esto se llevara un manejo más detallado de las sanciones por renta.
* Existirá un módulo llamado papelera en el cual se contendrán todos los artículos dados de baja en el sistema para poder ser reactivados.
* La interfaz gráfica del sistema le proporcionara un desplazamiento sencillo a través de este.
  1. **Limitaciones**
* La aplicación necesitara que el equipo donde sea usada cuente con el gestor de base de datos para funcionar.
* El software solo cubrirá aspectos relacionados con el control de los procesos de compra, venta y renta hablando de una manera administrativa y no operacional.
* La aplicación está diseñada para funcionar únicamente con el entorno de GameRush y no con BlockBuster (Renta y Venta de Películas).
* El lenguaje de programación a utilizar será JavaFX y como gestor de base de datos será utilizara Postgresql.
* La aplicación no será ejecutada mediante una arquitectura cliente servidor.
* El control de desarrollo ser llevara a cabo solamente para la sucursal Córdoba Veracruz mediante sus necesidades y especificaciones.
* El software contendrá únicamente el logo de la empresa GamerRush.

# **6 Herramientas**

**Características del equipo de cómputo a utilizar**

Marca: Lenovo

Modelo: Lenovo G480

Sistema operativo: Windows 8

Espacio de disco duro: 1 TB

Memoria RAM: 4GB

Procesador: Intel 1.80 ghz

Arquitectura: 64 bits

**Herramientas a ocupar**

1. pgAdmin
2. Eclipse Luna
3. JavaFX Scene Builder
4. Creately
5. MySQL Workbench 5.1 OSS
6. Java Runtime Environment 1.8.0\_51
7. Java Development Kit (jdk-8u45 de 64 bit)

**Lenguaje de programación**

1. JavaFX
2. SQL
3. CSS

**Sistema gestor de base de datos:**

PostgreSQL 9.4

# **7 Marco Teórico**

**UML**

El lenguaje UML comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía Rational fundada por Booch (dos reputados investigadores en el área de metodología del software. .  
El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (Object Modelling Tool). El primer borrador apareció en octubre de 1995. En esa misma época otro reputado investigador, Jacobson, se unió a Rational y se incluyeron ideas suyas. Estas tres personas son conocidas como los “tres amigos”. Además, este lenguaje se abrió a la colaboración de otras empresas para que aportaran sus ideas. Todas estas colaboraciones condujeron a la definición de la primera versión de UML.

Es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (Object Modeling Technique) o Booch sí definen procesos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión

**PostgresSQL**

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 robustos del mercado. Su desarrollo comenzó 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 al sistema.

**Historia**

Comienza como un proyecto de investigación del grupo del Profesor Michael Stonebraker en Berkeley. El proyecto original se llamó *Ingres*.

En 1987, Postgres introdujo reglas, procedimientos, tipos definibles y conceptos de orientación a objetos. Postgres fue comercializado con el nombre de Illustra y forma parte actualmente del Universal Server de Informix.

Una nueva versión fue lanzada en 1995 como Postgres95 y en ese momento el desarrollo pasó a manos de la comunidad de Internet. Postgres95 evolucionó hasta lo que es ahora PostgreSQL.

En 1996 se hizo evidente que el nombre Postgres95 no resistiría el paso del tiempo. Eligieron un nuevo nombre, PostgreSQL, para reflejar la relación entre el Postgres original y las versiones más recientes del SQL.

A continuación alguna de las características más importantes y soportadas por PostgreSQL:

**Generales**

* Es una base de datos 100% [ACID](http://es.wikipedia.org/wiki/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)
* Múltiples métodos de autentificación
* Acceso encriptado vía 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 programación, 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
* Índices compuestos, únicos, parciales y funcionales en cualquiera de los métodos de almacenamiento disponibles, B-tree, R-tree, hash o 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

**Java**

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

**Historia**

Java se creó como una herramienta de programación para ser usada en un proyecto de set-top-box en una pequeña operación denominada the Green Project en Sun Microsystems en el año 1991. El equipo (Green Team), compuesto por trece personas y dirigido por James Gosling, trabajó durante 18 meses en Sand Hill Road en Menlo Park en su desarrollo.

La intención de Sun era crear un lenguaje con una estructura y una sintaxis similar a C y C++, aunque con un modelo de objetos más simple y eliminando las herramientas de bajo nivel.

Los pilares en los que se sustenta Java son cinco: la programación orientada a objetos, la posibilidad de ejecutar un mismo programa en diversos sistemas operativos, la inclusión por defecto de soporte para trabajo en red, la opción de ejecutar el código en sistemas remotos de manera segura y la facilidad de uso.

Lo habitual es que las aplicaciones Java se encuentren compiladas en un bytecode (un fichero binario que tiene un programa ejecutable), aunque también pueden estar compiladas en código máquina nativo.

Sun controla las especificaciones y el desarrollo del lenguaje, los compiladores, las máquinas virtuales y las bibliotecas de clases a través del Java Community Process. En los últimos años, la empresa (que fue adquirida por Oracle) ha liberado gran parte de las tecnologías Java bajo la licencia GNU GPL.

Referencias

<http://alvearjofre.galeon.com/>

**Características de java**

* **Es orientado a objetos:** si bien existen detractores de esta modalidad, la programación orientada a objetos resulta muy conveniente para la mayoría de las aplicaciones, y es esencial para los videojuegos. Entre las ventajas más evidentes que ofrece se encuentra un gran control sobre el código y una mejor organización, dado que basta con escribir una vez los métodos y las propiedades de un objeto, independientemente de la cantidad de veces que se utilicen.
* **Es muy flexible:** Java es un lenguaje especialmente preparado para la reutilización del código; permite a sus usuarios tomar un programa que hayan desarrollado tiempo atrás y actualizarlo con mucha facilidad, sea que necesiten agregar funciones o adaptarlo a un nuevo entorno.
* **Funciona en cualquier plataforma:** a diferencia de los programas que requieren de versiones específicas para cada sistema operativo (tales como Windows o Mac), las aplicaciones desarrolladas en Java funcionan en cualquier entorno, dado que no es el sistema quien las ejecuta, sino la máquina virtual (conocida como Java Virtual Machine o JVM).
* **Su uso no acarrea inversiones económicas:** programar en Java es absolutamente gratis; no es necesario adquirir ninguna licencia, sino simplemente descargar el kit de desarrollo (Java Development Kit o JDK) y dar riendas sueltas a la imaginación.
* **Es de fuente abierta:** Java ofrece el código de casi todas sus librerías nativas para que los desarrolladores puedan conocerlas y estudiarlas en profundidad, o bien ampliar su funcionalidad, beneficiándose a ellos mismos y a los demás.
* **Es un lenguaje expandible:** continuando con el punto anterior, cada programador tiene la libertad de revisar y mejorar el código nativo de Java, y su trabajo puede convertirse en la solución a los problemas de muchas personas en todo el mundo. Infinidad de desarrolladores han aprovechado esta virtud del lenguaje y continúan haciéndolo.

**Java FX**

**¿Qué es JavaFX?**

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.

Invertir en la plataforma JavaFX proporciona las siguientes ventajas para los desarrolladores de Java y las empresas que forman parte del ecosistema de Java:

* Dado que la plataforma JavaFX está escrito en Java, los desarrolladores de Java pueden aprovechar sus habilidades existentes y las herramientas para desarrollar aplicaciones JavaFX.
* Debido a que Java es ampliamente utilizado, es fácil encontrar desarrolladores Java con experiencia que puede convertirse rápidamente en la creación de aplicaciones JavaFX productiva.
* Mediante el uso de un conjunto homogéneo de las tecnologías Java para el servidor y las plataformas de cliente, la plataforma JavaFX reduce el riesgo de la inversión mediante la reducción de la complejidad de las soluciones de negocio.
* Los costos de desarrollo también se han reducido debido a las ventajas antes mencionadas.
* La plataforma JavaFX proporciona a los desarrolladores un marco de desarrollo y el medio ambiente runtime para crear aplicaciones empresariales y de negocios que se ejecutan en múltiples plataformas que soportan Java.

Ver la [arquitectura de JavaFX y Marco](http://download.oracle.com/javafx/2.0/architecture/jfxpub-architecture.htm) documento para conocer la arquitectura de la plataforma JavaFX y conceptos clave.

**Una breve 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 estarán 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.

**¿Qué hay de nuevo en JavaFX 2.0?**

Las áreas de enfoque principal para el lanzamiento de JavaFX 2.0 son las siguientes:

* Java API para JavaFX que proporcionan todas las características del lenguaje familiar (como los genéricos, anotaciones, y el multithreading) que los desarrolladores de Java están acostumbrados a usar. Las API están diseñados para ser amigable con alternativas idiomas JVM, como JRuby y Scala. Como las capacidades de JavaFX están disponibles a través de las API de Java, puede seguir utilizando sus herramientas favoritas de desarrolladores Java (como el IDE, la refactorización de código, depuradores y analizadores) para desarrollar aplicaciones JavaFX.
* Un nuevo motor gráfico para manejar las modernas unidades de procesamiento gráfico (GPU). La base de este nuevo motor es una aceleración por hardware pipeline de gráficos, llamado Prism, que está acoplado con una caja de herramientas de ventanas, llamado Glass. Este motor gráfico es la base de los avances actuales y futuros para hacer gráficos ricos simple, suave y rápido.
* Un nuevo motor de los medios de comunicación que soporta la reproducción de contenido multimedia en la Web. Proporciona una estable, de baja latencia marco de los medios de comunicación que se basa en el marco de GStreamer multimedia.
* Un componente web que da la capacidad de insertar contenido HTML dentro de una aplicación JavaFX utilizando la tecnología de rende rizado WebKit HTML.Representación de aceleración por hardware está disponible usando Prism.
* Un navegador actualizado plug-in para JavaFX 2.0, que permite la carga de applets JavaFX basadas en Prism.
* Una amplia variedad de controles integrados de interfaz de usuario, que incluyen gráficos, tablas, menús y paneles. Además, una API se proporciona para permitir a terceros que contribuyan controles de interfaz de usuario que la comunidad de usuarios puede utilizar.
* Aplicaciones de ejemplo que muestran las diferentes características de la tecnología JavaFX 2.0, junto con un gran número de ejemplos de código y los fragmentos.

**Novedades técnicas**

Perfil Común - JavaFX se basa en el concepto de 'perfil común' es la reutilización de gran parte del código en todos los dispositivos sean móviles o de escritorio. Esto permite a los desarrolladores utilizar modelos de programación comunes, ya que construir para el escritorio o dispositivos móviles. Para diferenciar las cualidades de cada uno de los dispositivos, por ejemplo, JavaFX 1.1 proporciona una API para escritorio que incluye media vuelta y efectos visuales avanzados.

Integración para la creación de programas en tercero - JavaFX incluye plugins para Adobe Photoshop y Adobe Illustrator que te permite crear gráficos avanzados para integrar directamente en las aplicaciones JavaFX. Los plugins generan código en JavaFX Script que conservar el diseño y la estructura de los gráficos. Los desarrolladores pueden añadir fácilmente animaciones y efectos para los gráficos estáticos importados. También hay un convertidor de gráficos SVG que le permite importar y revisión después de su conversión al formato JavaFX.

**MoProSoft**

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.

**Historia**

En el 2002 la Secretaría Económica comenzó a organizar grupos de trabajos para definir las estrategias del Programa de Desarrollo de la Industria del Software (PROSOFT). Alcanzar niveles internacionales en capacidad de procesos fue una de las estrategias del programa. Se realizaron investigaciones acerca las características básicas de la industria de software mexicana con respecto a su tamaño y las capacidades de sus procesos. Lo cual arrojo que la gran mayoría de las empresas son Micro y Pequeñas Empresas y que el promedio de capacidades de sus procesos está a nivel 1. Se realizaron entrevista a empresas con esas características y lo que ellas querían es un modelo de procesos y evaluación, que sea fácil de entender, practico y barato.

Se revisó los modelos de procesos disponibles como: ISO 9000, ISO 15504, SW-CMM, pero; ninguno de los estándares o modelos cumplía con los requisitos expresados por la industria mexicana, por lo cual se decidió crear un nuevo modelo de procesos y un método de evaluación a la medida de la industria mexicana.

Entonces la Secretaría de Economía designo la elaboración del nuevo modelo a la Asociación Mexicana para la Calidad en Ingeniería del Software (AMCIS) en conjunto con la Universidad Nacional Autónoma de México (UNAM). Entre septiembre y diciembre del 2002 un grupo de 11 personas aportaron con su conocimiento y experiencia para la generación de MoProSoft (Modelo de Procesos para la Industria de Software). En junio de 2003 la SE lo hizo público a través del portal software.net.mx

Para completar la norma se necesitaba un método de evaluación basado en MoProSoft como modelo de procesos, para eso se reunió otro grupo de trabajo en el 2003, del cual surgió EvalProSoft (el método de Evaluación de Procesos de Software). Sin embargo faltaba probar que los dos modelos sirven en la práctica. Así se dio paso a un nuevo proyecto de Pruebas Controladas con cuatro empresas con el perfil promedio de la industria de software. Entre agosto y diciembre del 2004 se puso a prueba los modelos en las empresas seleccionadas, y se demostró que en un lapso de tiempo relativamente corto, las empresas pueden elevar sus niveles de capacidad.

Desde inicios del 2005 se buscó convertir los dos modelos en las normas mexicanas, la norma fue aprobada por el NYCE (Normalización Y Certificación en Electrónica) el 5 de julio y el 15 de agosto publicada en el Diario Oficial de la Federación. Su nombre completo es: Tecnología de la Información-Software-Modelos de procesos y evaluación para desarrollo y mantenimiento de software

**Procesos que maneja Moprosoft:**

**Alta Dirección (DIR)**

- Gestión de Negocio (GN): El propósito de Gestión de Negocio es 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.

**Gerencia (GER)**

- Gestión de Procesos (GPR): El propósito de Gestión de Procesos es 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 de Proyectos (GPY): El propósito de la Gestión de Proyectos es asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.

- Gestión de Recursos (GR): El propósito de Gestión de Recursos es 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. Su finalidad es apoyar el cumplimiento de los objetivos del Plan Estratégico de la organización.

* Recursos Humanos y Ambiente de Trabajo (RHAT): El propósito de Recursos Humanos y Ambiente de Trabajo es proporcionar los recursos humanos adecuados para cumplir las responsabilidades asignadas a los roles dentro de la organización, así como la evaluación del ambiente de trabajo.

* Bienes, Servicios e Infraestructura (BSI): El propósito de Bienes, Servicios e Infraestructura es proporcionar proveedores de bienes, servicios e infraestructura que satisfagan los requisitos de adquisición de los procesos y proyectos.
* Conocimiento de la Organización (CO): El propósito de Conocimiento de la Organización es mantener disponible y administrar la Base de Conocimiento que contiene la información y los productos generados por la organización.

**Operación (OPE)**

- Administración de Proyectos Específicos (APE): El propósito de la Administración de Proyectos Específicos es establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.

- Desarrollo y Mantenimiento de Software (DMS): El propósito de Desarrollo y Mantenimiento de Software es la realización sistemática de las actividades de obtención de requisitos, análisis, diseño, construcción, integración y pruebas de productos de software nuevo o modificado cumpliendo con los requisitos especificados.

Desarrollo y Mantenimiento de Software (DMS)

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. Se compone de una introducción y una descripción de requerimientos
* 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*. Este documento contiene la descripción textual y grafica de la estructura de los componentes de software.
* 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 softwareprobados*.*
* Integración y Pruebas. Conjunto de actividades para integrar y probar los componentesde 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.*

# **8 Bibliografía**

The Capability Maturity Model: Guidelines for Improving the Software Process. Carnegie Mellon University, Software Engineering Institute. 1994. Addison- Wesley.

ISO/IEC TR 15504 – 2:1998(E) Information Technology - Software process assessment. Part 2: A reference model for process and process capability, v. 3.3.

Software Engineering Institute, Capability Maturity Model Integrated (CMMI). CMU/SEI-2006-TR-008, Carnegie Mellon University, 2006.

ISO/IEC International Standard Organization/International Electrotechnical Commission, ISO/IEC 12207 Systems and software engineering —Software life cycle processes. ISO/IEC, 2008.

http://es.scribd.com/doc/80876237/Que-es-JavaFX#scribd

Java 2, Steven Holzner, Anaya Multimedia 2005 Segunda Edición 2004.

<http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=368:ique-es-java-concepto-de-programacion-orientada-a-objetos-vs-programacion-estructurada-cu00603b&catid=68:curso-aprender-programacion-java-desde-cero&Itemid=188>