**UNIVERSIDAD DE EL SALVADOR**

**FACULTAD DE INGENIERIA Y ARQUITECTURA**

**ESCUELA DE INGENIERIA EN SISTEMAS**

**DISEÑO DE SISTEMAS II**

**Conclusiones sobre la metodología**

**Sistema de “Manejo y control de notas del Colegio PALAS ATENEA”**

**Docente:** Ing. Marvin de Rosario Ortiz

**Grupo Teórico:** 01

**Integrantes*:***

***Carnet Apellidos Nombres***

*CR05074 Cardoza Rodríguez Jorge Alberto*

*GH06019 García Henríquez Eufemia Elizabeth*

*PA06019 Pineda Ayala Iris Yamileth*

*BD03006 Beltrán Domínguez José Eduardo*

*VV10031 Velasco Rivas Edwin Josué*

**Grupo N°:**05

Ciudad Universitaria, 1 de Diciembre de 2014

# Índice.

[Índice. 2](#_Toc405110384)

[Introducción. 3](#_Toc405110385)

[Eclipse 4](#_Toc405110386)

[Apache Tomcat 9](#_Toc405110387)

[Hibérnate 11](#_Toc405110388)

[MYSQL 15](#_Toc405110389)

[Spring 16](#_Toc405110390)

[IReport 21](#_Toc405110391)

[jdk 22](#_Toc405110392)

[Conclusiones sobre la metodología 23](#_Toc405110393)

[Recomendaciones para nuevas versiones 25](#_Toc405110394)

[Conclusiones 26](#_Toc405110395)

# Introducción.

El presente documento tiene como propósito, dar a conocer la tecnología que se utilizó para el desarrollo del proyecto en la materia de Diseño de Sistemas II. Dicho proyecto consiste en el análisis y el diseño de un registro y control de notas del colegio Palas atenea por lo cual se deberá hacer uso de la tecnología siguiente:

* **JDK 7**
* **IDE Eclipse Juno**
* **Apache Tomcat**
* **MySQL**
* **Hibernate**
* **Spring Framework y Security**
* **IReport para la elaboración de los reportes.**

# Eclipse

## Eclipse: es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", o puesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

## Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado: Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse. Y que son usados también para desarrollar el mismo Eclipse; Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.



## Arquitectura

La base para Eclipse es la Plataforma de cliente enriquecido **(del Inglés Rich Client Platform RCP).** Los siguientes componentes constituyen la plataforma de cliente enriquecido:

* **Plataforma principal - inicio de Eclipse, ejecución de plugins**
* **OSGi - una plataforma para bundling estándar.**
* **El Standard Widget Toolkit (SWT) - Un widget toolkit portable.**
* **JFace - manejo de archivos, manejo de texto, editores de texto**
* **El Workbench de Eclipse - vistas, editores, perspectivas, asistentes**

Los widgets de Eclipse están implementados por una herramienta de widget para Java llamada Standard Widget Toolkit, a diferencia de la mayoría de las aplicaciones Java, que usan las opciones estándar Abstract Window Toolkit (AWT) o Swing. La interfaz de usuario de Eclipse también tiene una capa GUI intermedia llamada JFace, la cual simplifica la construcción de aplicaciones basadas en SWT.

El entorno de desarrollo integrado **(IDE)** de Eclipse emplea módulos **(en inglés plug-in)** para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente para permitirle a Eclipse extenderse usando otros lenguajes de programación como son: **C/C++ y Python,** permite a Eclipse trabajar con lenguajes para procesado de texto como: **LaTeX,** aplicaciones en red como: **Telnet y Sistema de gestión de base de datos.**

La arquitectura **plugin** permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse. Y no tiene por qué ser usado únicamente con estos lenguajes, ya que soporta otros lenguajes de programación.

La definición que da el proyecto Eclipse acerca de su software es: **"una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".**

**En cuanto a las aplicaciones clientes,** Eclipse provee al programador con frameworks muy ricos para el desarrollo de: **aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc.** Por ejemplo, GEF **(Graphic Editing Framework - Framework para la edición gráfica)** es un plugin de Eclipse para el desarrollo de editores visuales que pueden ir desde **procesadores de texto** **wysiwyg** hasta **editores de diagramas UML, interfaces gráficas para el usuario (GUI), etc.** Dado que los editores realizados con GEF "viven" dentro de Eclipse, además de poder ser usados conjuntamente con otros plugins, hacen uso de su interfaz gráfica personalizable y profesional.

**El SDK de Eclipse** incluye las herramientas de desarrollo de Java, ofreciendo un **IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java.** Esto permite técnicas avanzadas de refactorización y análisis de código. Mediante diversos plugins estas herramientas están también disponibles para otros lenguajes como: **C/C++ (Eclipse CDT) y en la medida de lo posible para lenguajes de script no tipados como PHP o Javascript.** El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadatos en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente.

## Características

* Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con: **JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, test, etc.,** **y refactorización.**
* Asimismo, a través de **"plugins"** libremente disponibles es posible añadir control de versiones con **Subversion** e integración con **Hibernate.**
* Eclipse comenzó como un proyecto de [IBM](http://es.wikipedia.org/wiki/IBM) [Canadá](http://es.wikipedia.org/wiki/Canad%C3%A1). Fue desarrollado por OTI ([Object Technology International](http://es.wikipedia.org/w/index.php?title=Object_Technology_International&action=edit&redlink=1" \o "Object Technology International (aún no redactado))) como reemplazo de [VisualAge](http://es.wikipedia.org/w/index.php?title=VisualAge&action=edit&redlink=1" \o "VisualAge (aún no redactado)) también desarrollado por OTI. En noviembre del [2001](http://es.wikipedia.org/wiki/2001), se formó un consorcio para el desarrollo futuro de Eclipse como código abierto. En [2003](http://es.wikipedia.org/wiki/2003), fue creada la fundación independiente de IBM.

**Resumen de las versiones de Eclipse:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Versión** | | | **Fecha de lanzamiento** | **Versión de plataforma** | | **Proyectos** |
| Mars | | | Junio de 2015 (planeado) | ? | | [Mars Project](http://wiki.eclipse.org/Mars) |
| Luna | | | 25 junio 2014 | 4.4 | | [Luna projects](http://wiki.eclipse.org/Luna) |
| Kepler | | | 26 de junio de 2013 | 4.3 | | [Kepler projects](http://wiki.eclipse.org/Kepler) |
| Juno | | | 27 de junio de 2012 | 4.2 | | [Juno projects](http://wiki.eclipse.org/Juno) |
| Indigo | | | 22 de junio de 2011 | 3.7 | | [Indigo projects](http://wiki.eclipse.org/Indigo) |
| Helios | | | 23 junio de 2010 | 3.6 | | [Helios projects](http://wiki.eclipse.org/index.php/Helios) |
| Galileo | | | 24 de junio de 2009 | 3.5 | | [Galileo projects](http://wiki.eclipse.org/Galileo) |
| Ganymede | | | 25 junio de 2008 | 3.4 | | [Ganymede projects](http://wiki.eclipse.org/index.php/Ganymede_Simultaneous_Release) |
| Europa | | | 29 de junio de 2007 | 3.3 | | [Europa projects](http://wiki.eclipse.org/index.php/Europa_Simultaneous_Release) |
| Callisto | | | 30 de junio de 2006 | 3.2 | | [Callisto projects](http://www.eclipse.org/callisto/callistoprojects.php) |
| Eclipse 3.1 | | | 28 de junio 2005 | 3.1 | |  |
| Eclipse 3.0 | 28 de junio de 2004 | 3.0 |  | |

## C:\Users\EDU-BELTRAN\Desktop\eclipsejuno2.png

## 

## Lenguajes de programación utilizados en Eclipse

|  |  |  |
| --- | --- | --- |
| Lenguaje | Líneas de código | % |
| [Java](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java) | 1.911.693 | 92,66% |
| [ANSI C](http://es.wikipedia.org/wiki/ANSI_C) | 133.263 | 6,46% |
| [C++](http://es.wikipedia.org/wiki/C%2B%2B) | 10.082 | 0,49% |
| [JSP](http://es.wikipedia.org/wiki/JSP) | 3.613 | 0,18% |
| [sh](http://es.wikipedia.org/wiki/Bourne_Shell) | 2.066 | 0,10% |
| [perl](http://es.wikipedia.org/wiki/Perl) | 1.468 | 0,07% |
| [php](http://es.wikipedia.org/wiki/Php) | 896 | 0,04% |
| [sed](http://es.wikipedia.org/wiki/Sed_(inform%C3%A1tica)) | 2 | 0,00% |

# Apache Tomcat

Apache Tomcat (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation (aunque creado por Sun Microsystems).

**Estado de su desarrollo**

Tomcat es desarrollado y actualizado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software License. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 7.x, que implementan las especificaciones de Servlet 3.0 y de JSP 2.2. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

## Entorno

Tomcat es un contenedor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

**Estructura de directorios**

**La jerarquía de directorios de instalación de Tomcat incluye:**

* bin - arranque, cierre, y otros scripts y ejecutables.
* common - clases comunes que pueden utilizar Catalina y las aplicaciones web.
* conf - ficheros XML y los correspondientes DTD para la configuración de Tomcat.
* logs - logs de Catalina y de las aplicaciones.
* server - clases utilizadas solamente por Catalina.
* shared - clases compartidas por todas las aplicaciones web.
* webapps - directorio que contiene las aplicaciones web.
* work - almacenamiento temporal de ficheros y directorios.

## Versiones de Apache Tomcat

|  |  |  |
| --- | --- | --- |
| Versión | Fecha de Lanzamiento | Descripción |
| 3.0.x. (Lanzamiento inicial) | 1999 | Construido a partir de código donado por Sun basado en las especificaciones Servlet 2.2 y JSP 1.1. |
| 3.3.2 | 2004-03-09 | Último lanzamiento de la rama 3.x. Versión discontinuada. |
| 4.1.31 | 2004-10-11 |  |
| 4.1.36 | 2008-03-24 |  |
| 4.1.39 | 2008-12-03 |  |
| 4.1.40 | 2009-06-25 | Último lanzamiento de la rama 4.x. Versión discontinuada. |
| 5.0.0 | 2002-10-09 |  |
| 5.0.24 | 2004-05-09 |  |
| 5.0.28 | 2004-08-28 |  |
| 5.0.30 | 2004-08-30 | Último lanzamiento de la rama 5.0.x. Versión discontinuada. |
| 5.5.0 | 2004-08-31 |  |
| 5.5.1 | 2004-09-07 |  |
| 5.5.4 | 2004-11-10 |  |
| 5.5.7 | 2005-01-30 |  |

# Hibérnate

## Hibérnate: es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para (.Net) con el nombre de NHibérnate que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.Hibérnate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

## Historia

Hibérnate fue una iniciativa de un grupo de desarrolladores dispersos alrededor del mundo conducidos por Gavin King.Tiempo después, JBoss Inc. **(empresa comprada por Red Hat)** contrató a los principales desarrolladores de Hibérnate y trabajó con ellos en brindar soporte al proyecto.

La rama actual de desarrollo de Hibérnate es la 3.x, la cual incorpora nuevas características, como una nueva arquitectura Interceptor/Callback, filtros definidos por el usuario, y —opcionalmente— el uso de anotaciones para definir la correspondencia en lugar de **(o conjuntamente con)** los archivos XML. Hibérnate 3 también guarda cercanía con la especificación EJB 3.0 (aunque apareciera antes de la publicación de dicha especificación por Java Community Process) y actúa como la espina dorsal de la implementación de EJB 3.0 en JBoss.

Con la conceptualización anterior, se puede llegar a definir Hibernate como: “un Framework para persistir objetos en una base de datos relacional”. La idea de Hibernate la tuvo Gavin King, Ingeniero actual Grupo de JBoss, el cual cansado de la ineficiencia y complejidad de los sistemas de persistencia de la época, ideo un sistema base que fue apoyado por un inmenso grupo de programadores alrededor del mundo. Esta idea de King se transformo en un proyecto robusto, con licencia libre que actualmente es el más utilizado, en cuanto a Framework dedicados a persistencia se refiere.

3.1 Características

Hibernate es un potente servicio de persistencia Objeto – Relacional de alto rendimiento que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación. Ofrece persistencia automatizada y transparente de objetos a tablas en una base de datos relacional, utilizando metadata que describe el mapeo entre objetos y la base de datos en si. Entre las características más importantes de destacan:

• Licencia LGPL

• Ofrece su propio lenguaje de consulta HQL

• Implementado con XML o Anotaciones (JPA)

• Excelente Documentación

• Fácil de aprender

• Comunidad Activa

• Soporte para Windows (NHibernate)

3.2. Ventajas

• Productividad: Evita mucho del código confuso de la capa de persistencia, permitiendo centrarse en la lógica de negocio.

• Mantenibilidad: Por tener pocas líneas de código permite que el código sea más claro. Al dividir la capa de persistencia se puede identificar los errores muy fácilmente.

• Rendimiento: Existe la tendencia a pensar que una solución “manual” es más eficiente que una “automática”. Hibérnate tiene un buen desempeño pero todo depende realmente de como se realicen las consultas y como se configure el Framework.

• Indepencia del proveedor: Una solución ORM te abstrae del SGBD. Permite desarrollar en local con bases de datos ligeras sin implicación en el entorno de producción.

3.3. Componentes básicos

Para utilizar Hibérnate es importante saber que este se compone de un conjunto de librerías reunidas en varios Apis, en donde, dependiendo del problema se debe escoger y configurar, las librerías necesarias en el proyecto. (https://www.hibernate.org/)

Figura 4. API de Hibérnate

De todo el conjunto de Apis de Hibérnate, existen varias clases que permiten el trabajo básico con el Framework. Entre estas se encuentran:

• Session: Corresponde con un objeto que representa una unidad de trabajo con la base de datos (transacción). Además representa el gestor de persistencia, ya que dispone de la API básica para poder cargar y guardar objetos.

• Transaction: La API de Hibérnate contiene utilidades para demarcar la transaccionalidad de operaciones de manera programática.

• Query: Este interfaz permite crear consultas y enlazar argumentos a parámetros de la consulta (binding). Permite definir consultas en HQL (Hibernate Query Language) o en SQL.

• SessionFactory: Es una factoría de sesiones. Proporciona objetos Session. Es thread-safe. Permite concurrencia.

## Características

* Como todas las herramientas de su tipo, Hibérnate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: **el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional).** Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibérnate le permite a la aplicación manipular los datos en la base de datos operando sobre objetos, con todas las características de la [POO](http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos). Hibérnate convertirá los datos entre los tipos utilizados por Java y los definidos por [SQL](http://es.wikipedia.org/wiki/SQL).
* Hibérnate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.
* Hibérnate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.
* Hibérnate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una [API](http://es.wikipedia.org/wiki/Application_Programming_Interface) para construir las consultas programáticamente (conocida como "criteria").
* Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones [Java EE](http://es.wikipedia.org/wiki/Java_EE), mediante el componente **Hibérnate Annotations** que implementa el estándar [JPA](http://es.wikipedia.org/wiki/Java_Persistence_API), que es parte de esta plataforma.

# MYSQL

MySQL es un sistema de gestión de bases de datos: **relacional, multihilo y multiusuario**, con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C. Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, **MySQL es patrocinado por una empresa privada**, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

## Lenguajes de programación

Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo: **C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl;** cada uno de estos utiliza una interfaz de programación de aplicaciones específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

Aplicaciones

* MySQL es muy utilizado en [aplicaciones web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web), como [Drupal](http://es.wikipedia.org/wiki/Drupal" \o "Drupal) o [phpBB](http://es.wikipedia.org/wiki/PhpBB" \o "PhpBB), en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a [PHP](http://es.wikipedia.org/wiki/PHP), que a menudo aparece en combinación con MySQL.
* MySQL es una [base de datos](http://es.wikipedia.org/wiki/Base_de_datos) muy rápida en la lectura cuando utiliza el motor no transaccional [MyISAM](http://es.wikipedia.org/wiki/MyISAM" \o "MyISAM), pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación

# Spring

## Spring Framework y security

Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

La primera versión fue escrita por Rod Johnson, quien lo lanzó junto a la publicación de su libro Expert One-on-One J2EE Design and Development (Wrox Press, octubre 2002). El framework fue lanzado inicialmente bajo la licencia Apache 2.0 en junio de 2003. El primer gran lanzamiento fue la versión 1.0, que apareció en marzo de 2004 y fue seguida por otros hitos en septiembre de 2004 y marzo de 2005. La versión 1.2.6 de Spring Framework obtuvo reconocimientos Jolt Awards y Jax Innovation Awards en 2006. Spring Framework 2.0 fue lanzada en 2006, la versión 2.5 en noviembre de 2007, Spring 3.0 en diciembre de 2009, y Spring 3.1 dos años más tarde. El inicio del desarrollo de la versión 4.0 fue anunciado en enero de 2013. La versión actual es 4.1.1.

Si bien las características fundamentales de Spring Framework pueden ser usadas en cualquier aplicación desarrollada en Java, existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. A pesar de que no impone ningún modelo de programación en particular, este framework se ha vuelto popular en la comunidad al ser considerado una alternativa, sustituto, e incluso un complemento al modelo EJB (Enterprise JavaBean).

## Historia

## Spring Security comenzó a finales de 2003 como " El Sistema Acegi Seguridad para la primavera” . Una pregunta fue planteada en la lista de correo de la Primavera de los desarrolladores preguntando si había habido alguna consideración a una aplicación de seguridad basada en Spring. En el momento en que la comunidad de Primavera fue relativamente pequeño (sobre todo en comparación con el tamaño de hoy!), Y luego, la propia primavera sólo había existido como un proyecto SourceForge desde principios de 2003. La respuesta a la pregunta es que era un área que vale la pena, aunque la falta de tiempo actualmente impedido su exploración.

## Con esto en mente, una aplicación de seguridad simple fue construido y no depurados. Unas semanas más tarde, otro miembro de la comunidad de Primavera preguntó acerca de la seguridad, y en ese momento se le ofreció este código para ellos. Varias otras solicitudes siguieron, y para enero de 2004 una veintena de personas estaban usando el código. Estos usuarios pioneros fueron acompañados por otros que sugirieron un proyecto SourceForge estaba en orden, la cual fue debidamente establecidos en marzo de 2004.

## En aquellos primeros días, el proyecto no tiene ninguna de sus propios módulos de autenticación. Container Managed Seguridad fue invocada por el proceso de autenticación, con Acegi Seguridad lugar de centrarse en la autorización. Este era adecuado al principio, pero a medida que más y más usuarios solicitaron apoyo contenedor adicional, la limitación fundamental de interfaces de reino de autenticación específica del contenedor se hizo evidente. También hubo un problema relacionado de añadir nuevos archivos JAR a classpath del contenedor, que era una fuente común de confusión en el usuario final y una mala configuración.

## Servicios de autenticación Acegi específicas de seguridad se introdujeron posteriormente. Alrededor de un año después, Acegi Seguridad se convirtió en un subproyecto oficial de Spring Framework. La versión final 1.0.0 se publicó en mayo de 2006 - después de más de dos años y medio de uso activo en numerosos proyectos de software de producción y muchos cientos de mejoras y contribuciones de la comunidad.

## Acegi Seguridad se convirtió en un proyecto oficial de Primavera Portfolio hacia el final de 2007 y fue renombrado como " Spring Security” .

## Hoy Spring Security cuenta con una comunidad de código abierto fuerte y activa. Hay miles de mensajes sobre Spring Security en los foros de soporte. Hay un núcleo activo de los desarrolladores que trabajan en el propio código y una comunidad activa que también comparte regularmente parches y apoyar a sus compañeros.

## Módulos

Spring Framework comprende diversos módulos que proveen un rango de servicios:

* **Contenedor de inversión de control:** permite la configuración de los componentes de aplicación y la administración del ciclo de vida de los objetos Java, se lleva a cabo principalmente a través de la inyección de dependencias.
* **Programación orientada a aspectos:** habilita la implementación de rutinas transversales.
* **Acceso a datos:** se trabaja con RDBMS en la plataforma java, usando Java Database Connectivity y herramientas de Mapeo objeto relacional con bases de datos NoSQL.
* **Gestión de transacciones:** unifica distintas APIs de gestión y coordina las transacciones para los objetos Java.
* **Modelo vista controlador:** Un framework basado en HTTP y servlets, que provee herramientas para la extensión y personalización de aplicaciones web y servicios web REST.
* **Framework de acceso remoto:** Permite la importación y exportación estilo RPC, de objetos Java a través de redes que soporten RMI, CORBA y protocolos basados en HTTP incluyendo servicios web (SOAP).
* **Convención sobre Configuración:** el módulo Spring Roo ofrece una solución rápida para el desarrollo de aplicacionesa basadas en Spring Framework, privilegiando la simplicidad sin perder flexibilidad.
* **Procesamiento por lotes:** a framework for high-volume processing featuring reusable functions including registro/trazado, transaction management, job processing statistics, job restart, skip, and resource management.
* **Autenticación and Autorización:** procesos de seguridad configurables que soportan un rango de estándares, protocolos, herramientas y prácticas a través del subproyecto Spring Security (antiguamente Acegi).
* **Administración Remota:** Configuración de visibilidad y gestión de objetos Java para la configuración local o remota vía JMX.
* **Mensajes:** Registro configurable de objetos receptores de mensajes, para el consumo transparente desde la a través de JMS, una mejora del envío de mensajes sobre las API JMS estándar.
* **Testing:** Soporte de clases para desarrollo de unidades de prueba e integración.

**¿Qué es Spring Security?**

Spring Security proporciona servicios integrales de seguridad para aplicaciones de software empresarial basado en J2EE. Hay un énfasis especial en el apoyo a proyectos construidos utilizando el marco de la Primavera, que es la solución líder en J2EE para el desarrollo de software empresarial. Si usted no está utilizando Spring para el desarrollo de aplicaciones empresariales, acogemos invitamos a echar un vistazo más de cerca. Algunos familiaridad con la primavera - y en los principios de inyección de dependencia particulares - le ayudará a ponerse en marcha con Spring Security con mayor facilidad.

La gente usa la primavera de Seguridad por muchas razones, pero la mayoría se sienten atraídos por el proyecto después de encontrar los elementos de seguridad de de J2EE Servlet o EJB Especificación Especificación carecen de la profundidad requerida para los escenarios típicos de aplicaciones empresariales. Mientras que mencionar estas normas, es importante reconocer que no son portátiles a nivel WAR o EAR. Por lo tanto, si cambia los entornos de servidores, típicamente es un montón de trabajo para volver a configurar la seguridad de la aplicación en el nuevo entorno de destino. El uso de Spring Security supera estos problemas, y también le trae docenas de otros, las características de seguridad personalizables útiles.

Como usted probablemente sabe dos grandes áreas de seguridad de las aplicaciones son " autenticación " y " autorización " (o " control de acceso " ). Estas son las dos áreas principales que los objetivos de seguridad de primavera. " autenticación " es el proceso de establecimiento de un director es quien dice ser (un " director " generalmente significa que un usuario, dispositivo o algún otro sistema que puede realizar una acción en su aplicación). " Autorización " se refiere al proceso de decidir si se permite que un director para realizar una acción dentro de su aplicación. Para llegar al punto donde se necesita una decisión de autorización, la identidad del principal ya ha sido establecida por el proceso de autenticación. Estos conceptos son comunes, y para nada específico para la primavera de Seguridad.

En un nivel de autenticación, Spring Security es compatible con una amplia gama de modelos de autenticación. La mayoría de estos modelos de autenticación o bien son proporcionados por terceros, o son desarrollados por los órganos pertinentes de normalización, tales como el Grupo de Trabajo de Ingeniería de Internet. Además, Spring Security ofrece su propio conjunto de características de autenticación. En concreto, Spring Security actualmente soporta la integración con la autenticación de todas estas tecnologías:

Cabeceras HTTP autenticación básica (un estándar basado en RFC IEFT)

Encabezados de autenticación HTTP Digest (un estándar basado en RFC IEFT)

Intercambio de certificados de cliente HTTP X.509 (un estándar basado en RFC IEFT)

LDAP (un enfoque muy común para la autenticación entre la plataforma necesita, especialmente en entornos de gran tamaño)

Autenticación basada en formularios (para las necesidades de la interfaz de usuario simple)

Autenticación OpenID

Autenticación basada en las cabeceras de solicitud preestablecida (como Computer Associates Siteminder)

JA-SIG Central Authentication Service (también conocido como CAS, que es una fuente abierta de sesión único populares en el sistema)

Propagación de contexto de autenticación transparente para invocación de método remoto (RMI) y HttpInvoker (un protocolo de comunicación remota de primavera)

Automático "Recuerdo-me" de autenticación (por lo que puedo marcar una casilla para evitar la re-autenticación para un periodo de tiempo predeterminado)

Autenticación anónima (permitiendo que cada llamada a asumir automáticamente una identidad particular de la seguridad)

Gestión como la autenticación (que es útil si una llamada debe proceder con una identidad de seguridad diferente)

JAAS (JAAS)

JEE autenticación contenedor (lo que aún puede utilizar Container Managed autenticación si se desea)

Kerberos

Java Open Source Single Sign On (JOSSO) \*

# IReport

**iReport:** es una herramienta opensource construida por Jaspersoft; escrita en java y basada en Netbeans que permite generar reportes haciendo uso de la librería JasperReports de forma visual, gracias a esta herramienta tendremos una interface gráfica que permitirá construir reportes de manera más fácil y cómoda.

## características de iReport

Las características más destacadas de esta herramienta son:

* **Editor WYSIWYG:** es el acrónimo de (What You See Is What You Get), en español, **"lo que ves es lo que obtienes"**. Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final.
* Vista previa integrada (permite visualizar el reporte en diferentes formatos a través del Preview).
* Soporte completo a las etiquetas de la librería JasperReports.
* Administración de errores.
* Gráficas en HTML5.
* Soporte para construcción de tablas anidadas.
* Soporte para todas las bases de datos accesibles por JDBC.
* Soporte de templates.

Un template: **es un conjunto de archivos que determinan la estructura y el aspecto visual de un sitio web, y tiene como ventaja principal disminuir tiempos y costos de desarrollo.**

* Librería de estilos.
* Drag-and-drop sobre los componentes.

# jdk

Se trata de un conjunto de herramientas (programas y librerías) que permiten desarrollar (compilar, ejecutar, generar documentación, etc.) programas en lenguaje Java.

Existen versiones del JDK para prácticamente todos los Sistemas Operativos y existen también distintos programas comerciales

Sun distribuye gratuitamente el JDK “oficial” para los siguientes sistemas operativos: Windows, Solaris y Linux

Se pueden conseguir en Internet (http://www.javasoft.com/)

Los profesores de la asignatura los facilitarán a los alumnos que lo deseen en un CD-ROM prestado para la instalación.

Los JDK incorporan una herramienta de Debugger (detener la ejecución de un programa en la línea que se desee y poder conocer el valor de las variables en ese momento)

La versión JDK1.2 se ha empezado a denominar también Java 2

JRE (Java Runtime Environment)

Versión reducida del JDK que permite ejecutar código Java

No permite compilar ni utilizar el Debugger

Viene incluido en el JDK pero también puede instalarse separadamente

Es de libre distribución

¿Cómo puedo empezar a desarrollar programas Java con Java Development Kit (JDK)?

La escritura de aplicaciones y applets de Java necesita herramientas de desarrollo como JDK. JDK incluye Java Runtime Environment, el compilador Java y las API de Java. Familiarizarse resulta fácil para los programadores nuevos y con experiencia.

Los desarrolladores también pueden consultar Oracle Technology Network para desarrolladores Java para obtener toda la información necesaria sobre la tecnología Java, incluyendo documentación y formación.

# conclusiones sobre la metodología

Las políticas de desarrollo son una etapa crucial de la etapa de diseño del proceso de planificación del diseño en cuestión, ya que al orientar la planificación le permiten alcanzar sus objetivos o no.

En la actualidad a las políticas de desarrollo se las denomina entre ellas conforme al conocimiento y uso de algún método o diseño mediante el cual se le dé solución al problema planteado atraves del tiempo y recurso que poseo para poder llegar a la meta en este caso el diseño completo con los requerimientos que se debe tener a la hora de diseñar un sistema.

Una posible explicación de falta de claridad es que la planificación en general y las políticas de desarrollo en particular que utilizamos, no se enmarcan en procesos recientes y en toma de decisiones no controladas.

Dicha metodología usada en el diseño de este sistema es y era nueva en muchos aspectos como programadores y diseñadores en un camino por el cual se ha recorrido de muchas maneras aprendiendo de la mano en cuanto a la metodología vista en clases de ingeniería de software por lo cual en el desarrollo de la aplicación ha ido de la mano con las enseñanzas vistas en clases expositivas y laboratorios de dicha materia.

Dichas situación influyo en el diseño y la implementación de un cronograma de trabajo o planificación ya que no se contaba con que dicha situación no afectaría tanto en el diseño de nuestra aplicación por lo cual estuvimos obligados a poner en práctica a la misma ves que nos enseñaban la teoría y la práctica.

El aprender una nueva herramienta de diseño e implementación no solo conlleva un poco más de trabajo sino también muchas horas prácticas de las cuales muchas veces nos vemos obligados a tomar decisiones cruciales pare le desarrollo del sistema.

* El uso e implementación de un nuevo Framework influyo en la planificación antes hecha
* El desarrollo de los módulos del sistema se hizo más extensa al configurar muchas de las aplicaciones necesarias para el diseño (servidor Tomcat, eclipse etc.)
* La programación se hiso d forma optimista sin contar con las consideraciones correspondientes
* Utilización de mucho recurso con respecto a las pc
* Tiempo de descarga de las aplicaciones usadas
* Aprendizaje de un nuevo lenguaje de programación por combinación de java y jsp, Hibernate etc.
* Versiones de software
* Control de versiones
* El avance en la materia ingeniería de software
* El diseño e implementación de la base de datos con sus respectivas relaciones y script de la base de datos
* Horas laborales
* Falta de coordinación
* Incompatibilidad de sistema
* Análisis de tutoriales
* Inconsistencia en el modelo de los datos
* Falta de información en las herramientas
* Aprendizaje en el uso del nuevo IDE eclipse
* El uso de un gestor de base de datos en conflicto con los puertos usados
* Aprendizaje del nuevo lenguaje de programación por parte de los integrantes
* El tiempo estimado del diseño es poco
* Nuevos requisitos no tomados en cuenta por el diseñador

# recomendaciones para nuevas versiones

En dicho apartado podremos exponer muchas de las situaciones de las cuales no se implementaran en este diseño del sistema por lo cual se deja a criterio del solicitante dichas sugerencias si decide buscar ayuda profesional para poder ejecutarlas.

* Llevar un registro de las operaciones realizadas por los usuarios como un tipo de reporte por usuario en dicho sistema.
* Verificar quien esta solvente o no en sus pagos de colegiatura dicho requerimiento no fue solicitado por el usuario.
* Llevar un control de asistencia a clases o materias impartidas.
* El diseño de implementación a nivel de los demás años académicos ejemplo el bachillerato
* Generar reporte de alumnos con promedios de edades
* Generar reporte de alumnos con promedios de género humano

# conclusiones

• En el mundo de software, el uso de Framework es una buena alternativa para los programadores de la actualidad, ya que mejoran sustancialmente su trabajo, dedicándose a otras tareas más significativas de la lógica del negocio.

• El uso de aplicaciones de persistencia de objetos en bases de datos relacionales es cada vez común. En el mercado existen muchos Framework dedicados a esta tarea que hacen el trabajo de almacenamiento de objetos de una forma transparente para el programador.

• Hibérnate es uno de los mejores Framework de persistencia de la actualidad ya que posee características in mejoradas como: buen soporte, facilidad de adaptación, categoría libre, soporte de java y .net, buena documentación.

Durante el desarrollo de dicho sistema ha habido complicaciones y eso es debido a nivel de aprendizaje que se tiene como programador o diseñador de una nueva herramienta de programación por tratar llevar de la mano o innovar por medio de herramientas tecnológicas las cuales tendrían que tener un poco más de tiempo de lo acostumbrado para su implementación