**Integrantes:**

* David Culcay Orellana
* Edward Holguín
* Carlos Cerezo
* Ricardo Campozano

**Profesor:**

* Ing. Gonzalo Luzardo

Contenido

[Objetivo 3](#_Toc316838488)

[**1.** **Diseño lógico de su aplicación donde muestre la integración de servlets, filtros, JSP y Beans y que reflejen como el modelo MVC2 fue utilizado en el proyecto.** 3](#_Toc316838489)

[**2.** **Diseño de la base de datos.** 5](#_Toc316838490)

[**3.** **Descripcion de las soluciones encontradas a cada detalle técnico relevante de su proyecto con las respectivas referencias bibliográficas** 5](#_Toc316838491)

[**4.** **Una descripción detallada de como instalar todos los componentes de su aplicación sin la necesidad de utilizar un IDE como netbeans.** 5](#_Toc316838492)

[Instalación de Tomcat 5](#_Toc316838493)

[Instalar JSTL 6](#_Toc316838494)

[Añadir las librerías *hibernate:* 6](#_Toc316838495)

[Instalar PostgreSQL 7](#_Toc316838496)

[14](#_Toc316838497)

[15](#_Toc316838498)

# Objetivo

Elaborar un sitio web aplicando tecnologías modernas en el desarrollo del diseño de sitios

Web. Utilizar las siguientes tecnologías:

* XML para intercambio de datos.
* Modelo MVC2 (Orientado a páginas Web).
* CSS3 para layout.
* HTML5 para las páginas.
* Bases de Datos.
* AJAX para evitar full-page-reloads.
* Filtros.
* Parámetros de inicialización de servlets y filtros.
* Menú dinámico

## Sitio Web

Cholo Machine

Tienda de Zapatos

## Propósito

CholoMachine.com es una tienda de zapatos donde podrás ver nuestros mejores productos, con una variedad de marcas ya sea para vestir o practicar tu deporte favorito. Además si tienes alguna inquietud o comentario puedes enviarnos un correo desde la misma página y te responderemos.

Proceso de Desarrollo.

1. Descripción de cómo pudo integrar la característica de redes sociales en el proyecto.
2. Diseño de la base de datos.
3. Descripción de las soluciones encontradas a cada detalle técnico relevante de su proyecto con las respectivas referencias bibliográficas.
4. Una descripción detallada de cómo instalar todos los componentes de su aplicación sin la necesidad de utilizar un IDE como NetBeans.

## Descripción de cómo pudo integrar la característica de redes sociales en el proyecto.

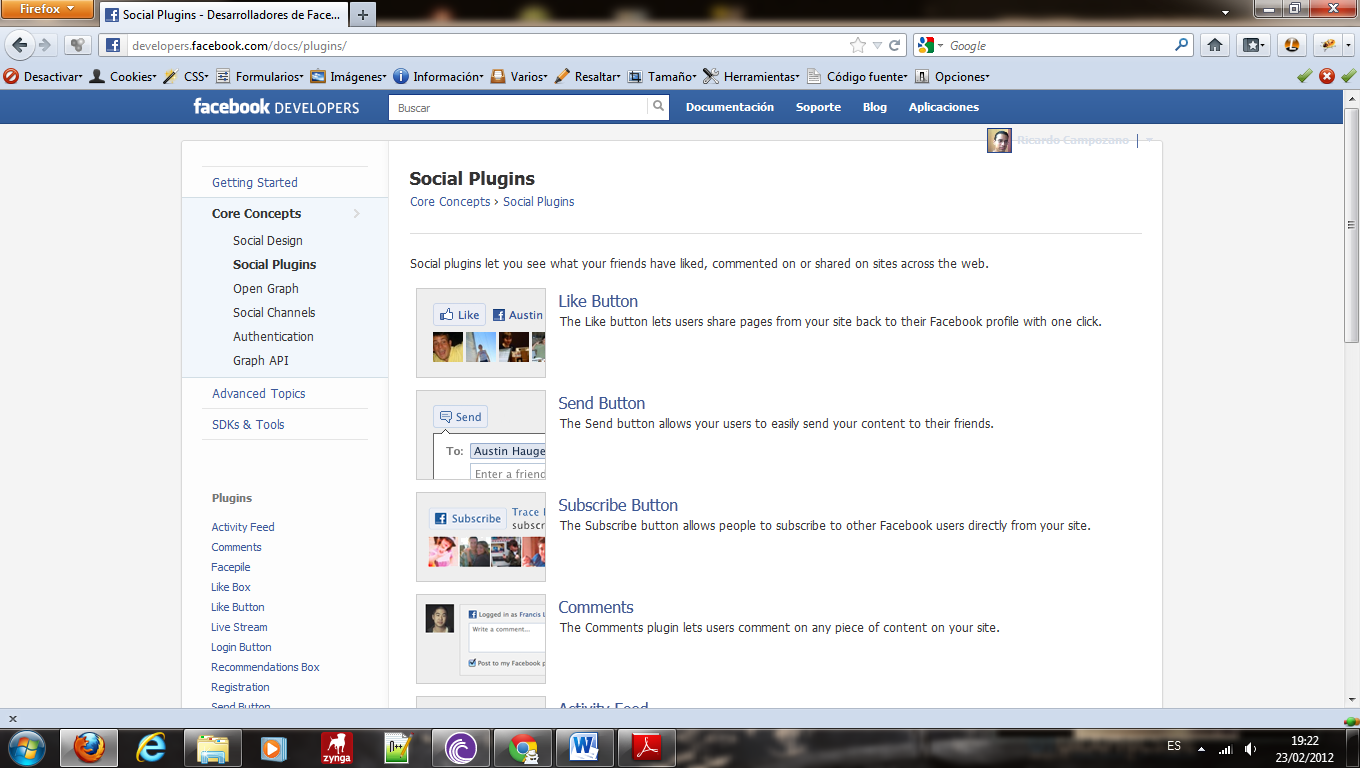
Primero creamos la página del proyecto, “Cholo Machine Comunnity”, en Facebook**.**

Aquí subiremos imágenes o comentarios relacionados al nuestraempresa



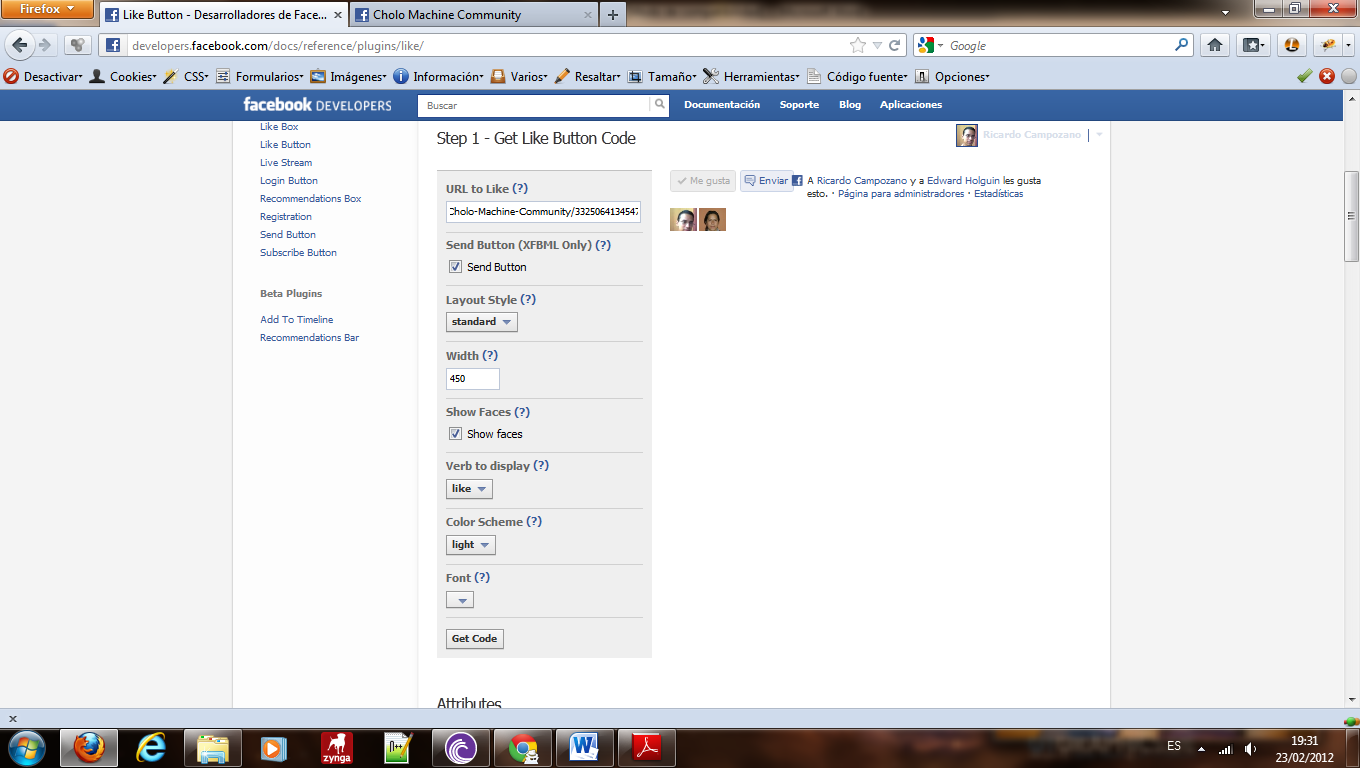
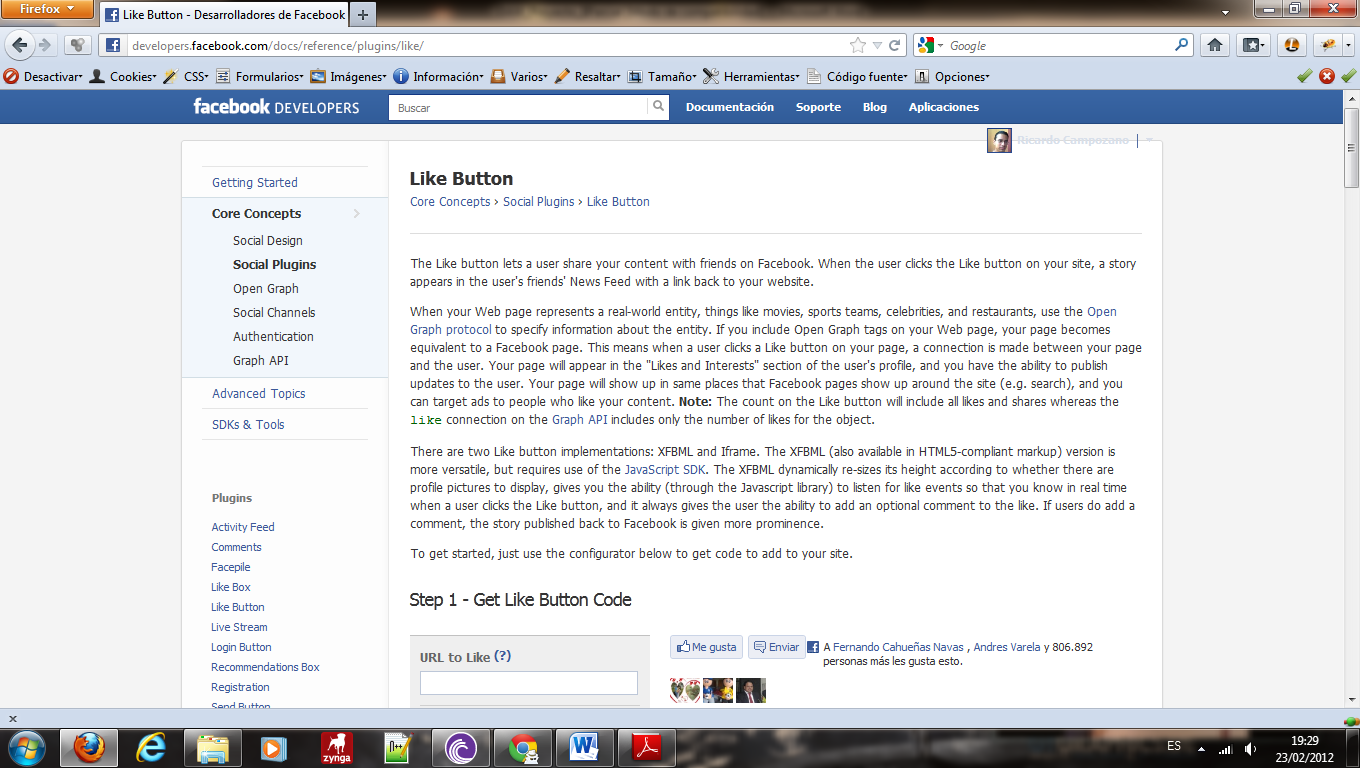
Luego de esto haremos uso de los plugins sociales.

Para esto iremos a: Editar la Pagina 🡪Recursos 🡪Usa los Plugins sociales

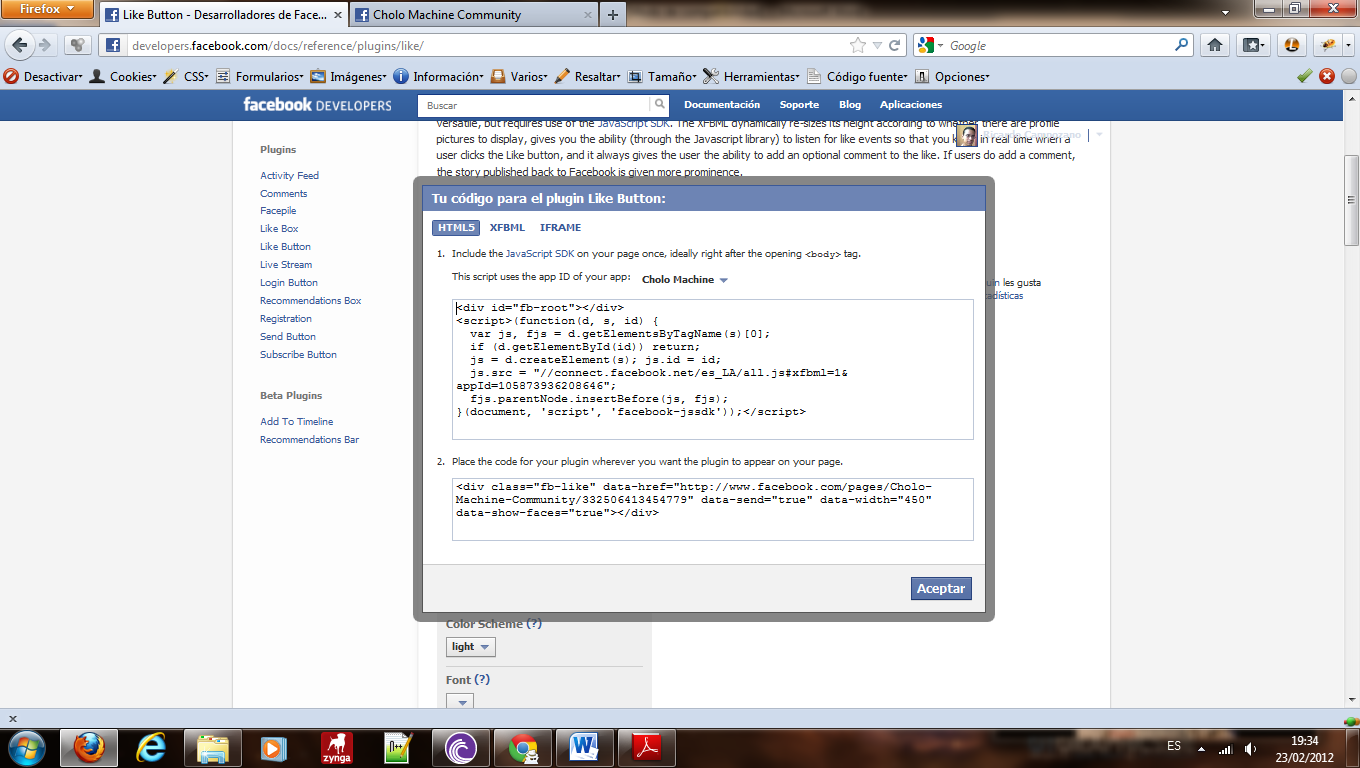


Como podemos ver en la imagen, aquí encontramos las características más comúnmente usadas como son: el botón me gusta, comentarios, suscribirse, etc.

Para poder integrar estas características en nuestra aplicación, daremos click sobre el plugin de nuestro interés. A continuación se mostrara una descripción del mismo y un cuadro donde podemos conocer los parámetros necesarios para su funcionamiento.



Una vez que hayamos definido los parámetros adecuadamente podremos hacer uso de la opción “Get code” que nos mostrara el código necesario y donde debería ser colocado.



Luego de haber hecho esto podemos disfrutar de las características de Facebook en nuestra página web.

## Diseño lógico de su aplicación donde muestre la integración de servlets, filtros, JSP y Beans y que reflejen como el modelo MVC2 fue utilizado en el proyecto.

**FiltroProducto.class**

**FiltroContenido.class**

**FiltroUsuario.class**

**..**

Filtros

Navegador WEB

**Usuario.class**

**Rol.class**

**Permiso.class**

**Articulo.class**

**Producto.class**

**Tema.class**

Base de Datos

EntidadesBD (Beans)

**index.jsp**

**administrar\_Producto.jsp**

**administrar\_Temas.jsp**

**administrar\_Articulos.jsp**

**administrar\_Usuarios.jsp**

Modelo (Servlets)

Vistas

**AdministrarProducto.class**

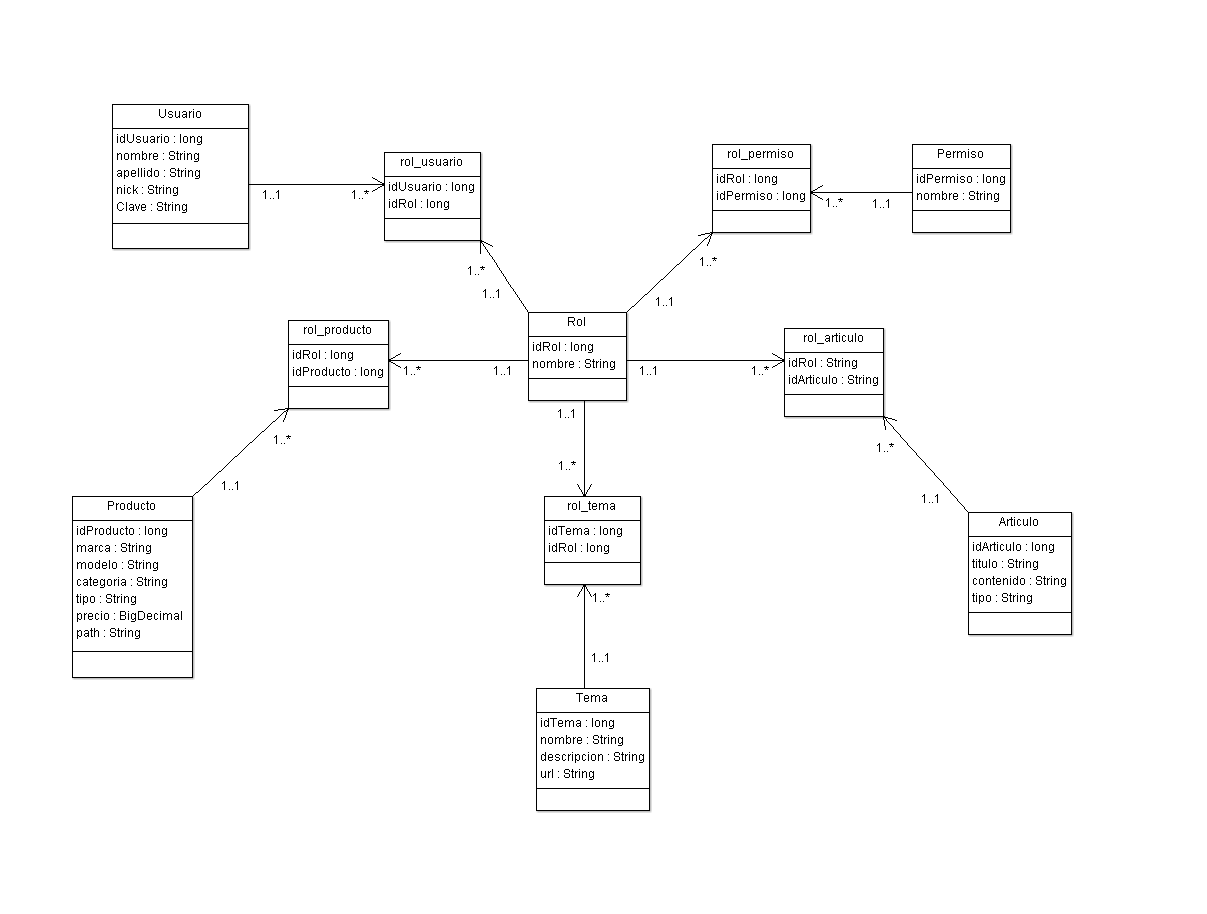
**AdministarUsuario.class**

**AdministrarTemas.class**

**AdministrarArticulos.class**

Controlador **Controlador.class**

## Diseño de la base de datos.



## Descripción de las soluciones encontradas a cada detalle técnico relevante de su proyecto con las respectivas referencias bibliográficas

### Hibernate

Trabajar con software orientado a objetos y bases de datos relacionales puede hacernos invertir mucho tiempo en los entornos actuales.  Hibernate es una herramienta que realiza el *mapping* entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java.  El término utilizado es ORM (object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

Hibernate no solo realiza esta esta transformación sino que nos proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que nos reducen el tiempo de desarrollo.

Hibernate funciona asociando a cada tabla de la base de datos un Plain Old Java Object (POJO, a veces llamado Plain Ordinary Java Object).  Un POJO es similar a una Java Bean, con propiedades accesibles mediante métodos setter y getter. Para poder asociar el POJO a su tabla correspondiente en la base de datos, Hibernate usa los ficheros hbm.xml.

Para la clase Cat se usa el fichero Cat.hbm.xml para mapearlo con la base de datos. En este fichero se declaran las propiedades del POJO y sus correspondientes nombres de columna en la base de datos, asociación de tipos de datos, referencias, relaciones x a x con otras tablas etc.

Hibernate nos proporciona además un lenguaje con el que realizar consultas a la base de datos.

Este lenguaje es similar a SQL y es utilizado para obtener objetos de la base de datos según las condiciones especificadas en el HQL.

El uso de HQL nos permite usar un lenguaje intermedio que según la base de datos que usemos y el dialecto que especifiquemos será traducido al SQL dependiente de cada base de datos de forma automática y transparente.

### JSP 2.0 y EL (Expresion Languaje)

La especificación 2.0 de JSP ha introducido una nueva librería estandar de etiquetas, denominada JSTL.

Estas etiquetas tratan de abstraer la complejidad de introducir código Java (scriptlet) dentro de JSP, del mismo modo que trata de evitar que cada equipo de desarrollo cree un juego de etiquetas no estándar para las mismas labores.

Dentro de estas etiquetas, se utiliza un lenguaje llamado EL, lenguaje de expresiones, que pretende ser un lenguaje más sencillo que Java, para realizar operaciones.

## Una descripción detallada de como instalar todos los componentes de su aplicación sin la necesidad de utilizar un IDE como netbeans.

### Instalación de Tomcat

1. Bajar la versión binaria de Tomcat en: <http://tomcat.apache.org/download-70.cgi>.
2. Descomprimir el archivo de Tomcat.
3. Posteriormente se debe definir una variable ambiental la cual le indicará al sistema la ubicación de Tomcat, para instalaciones Windows esta variable ambiental puede ser definida de la misma manera que CLASSPATH, descrita en las instrucciones del JDK (es decir la dirección donde se encuentra el jdk1.7.0).
4. Del archivo que se descomprimió dentro de la carpeta “bin” hacemos doble clic sobre el archivo startup.wbf (windows batch file).

### Instalación de JRE

1. Ir a la [sección de descarga del JRE (Java Runtime Environment)](http://www.ayudabittorrent.com/descargar-jre) y descargarlo desde nuestros servidores.
2. Una vez descargado se debe hacer doble clic en el icono para iniciar la instalación.
3. Lo primero que debemos escoger es entre la instalación típica y la personaliza. Escogeremos la típica ya que es más sencilla y permite la ejecución de las aplicaciones hechas en Java.
4. A continuación aceptaremos la licencia pulsando el botón *Aceptar >*.
5. Si se pulsa en el botón *Finalizar* la instalación se habrá finalizado.

## Instalar JSTL

Para trabajar con JSTL necesitamos

* Un kit de desarrollo java como el Java Standard Edition (J2SE) de Sun.
* Un servidor que ejecute Servlets y JSP como Tomcat. JSTL 1.0 necesita JSP 1.2 y Servlets 2.3 implementadas por Tomcat 4. JSTL 1.1 requiere JSP 2.0 y Servlets 2.4, implementadas por Tomcat 5. O dicho de otro modo, si vas a trabajar con la última versión de JSTL (la 1.1) usa Tomcat 5.
* Una implementación de JSTL como la proporcionada por el proyecto Taglibs de Apache. La distribución (jakarta-taglibs-standard-1.1.0.zip) contiene documentación, ejemplos, la implementación en sí, y varios jars de dependencias. Si quieres ahorrar tiempo baja solo los jars siguientes: jstl-1.0.2.jar (20k), standard-1.0.4.jar (503k).

## Añadir las librerías *hibernate:*

Es curioso que algo tan básico como añadir las librerías de Hibernate a nuestros proyectos Java no suela aparecer por ahí explicado de forma clara y sencilla, de hecho no lo he encontrado ni en los tutoriales oficiales de [Hibernate](http://www.hibernate.org/docs) (JBoss Community).

Suponiendo que utilizamos el IDE Eclipse, el proceso es el siguiente :

* Seleccionamos “Project –> Properties” en el menú.
* En la opción Java Build Path, pestaña “Libraries” pulsamos el botón “Add Library”. Aparecerá una ventana de diálogo con una lista de varios tipos de librería. Seleccionamos “User Library” y pulsamos “Next”.
* En la siguiente ventana, pulsamos el botón “User Libraries”.
* En la siguiente pantalla pulsamos el botón “New” y le ponemos un nombre a la librería cuando se nos solicite, por ejemplo “Hibernate”. Ahora veremos que en la ventana “Defined User Libraries” aparece la librería que acabamos de crear.
* Tan sólo queda añadir los JARs; para ello pulsamos en el botón “Add JARs”. Cuando se ha instalado Hibernate como parte de las JBoss Tools, los jars de Hibernate podemos descargarlo de <http://sourceforge.net/projects/hibernate/files/hibernate4/4.0.1.Final/hibernate-release-4.0.1.Final.zip/download>.
* Finalmente selecionamos de golpe todos los JARs y le damos a OK; seguidamente hacemos click en el botón “Finish” y ya tenemos los jars asociados a nuestra librería “Hibernate” que quedará añadida al proyecto.

**Link de Referencia**

<http://www.youtube.com/watch?v=gTjlLC-jaVg&feature=related>

<http://www.youtube.com/watch?v=TiDvtPkGBuc&feature=player_embedded>

<http://www.programacion.com/articulo/jstl_1_0_estandarizando_jsp__179>

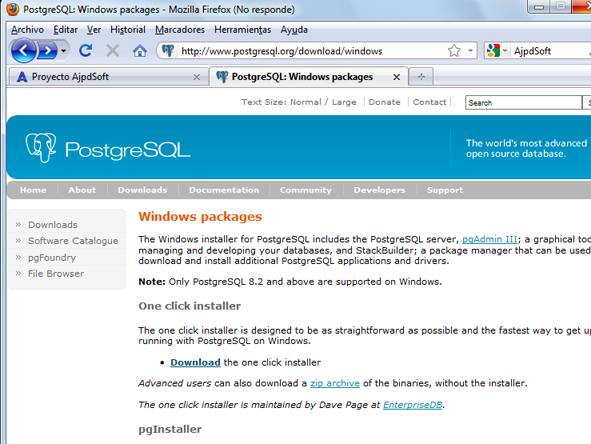
<http://www.9lessons.info/2008/09/jsp-login-page-to-connect-mysql-atabase.html>

## Instalar PostgreSQL

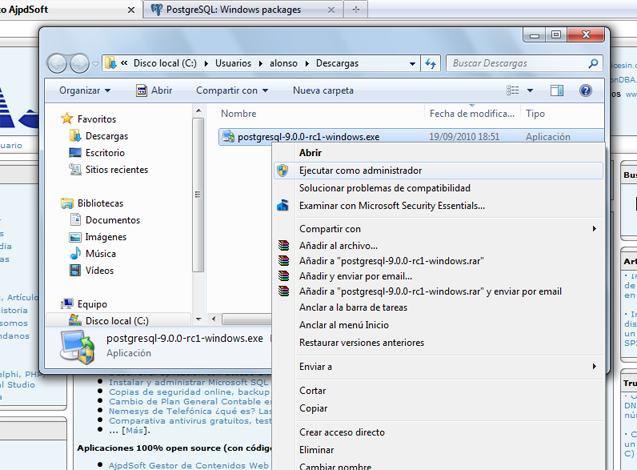
Abriremos un navegador web y accederemos a la URL:

*http://www.postgresql.org/download/windows*

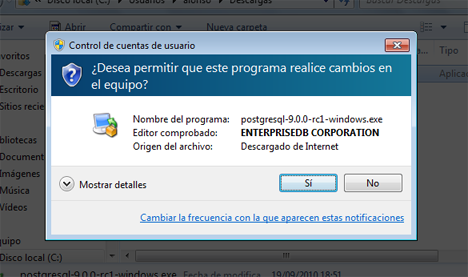
En nuestro caso descargaremos la versión aún no estable 9.0.0 RC1 para Windows de 32 bits (Win x86-32):



Una vez descargado el archivo de instalación de PostgreSQL (*postgresql-9.0.0-rc1-windows.exe* de 45,5 MB) pulsaremos con el botón derecho del ratón sobre él y seleccionaremos "Ejecutar como administrador":



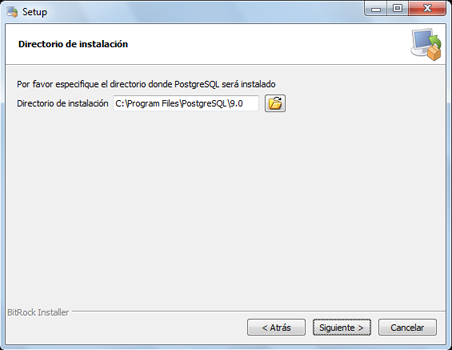
Si tenemos activado el control de cuentas de usuario nos mostrará una advertencia con el texto "¿Desea permitir que este programa realice cambios en el equipo?", pulsaremos "Sí" para continuar con la instalación de PostgreSQL:



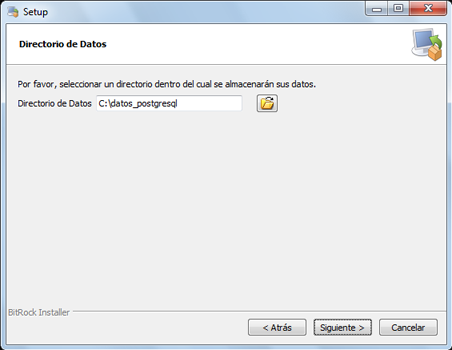
Se iniciará el asistente para instalar PostgreSQL, pulsaremos "Siguiente":



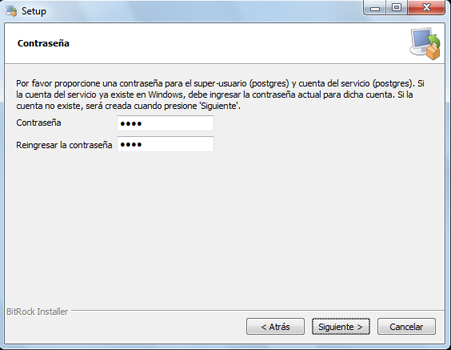
Indicaremos la carpeta de instalación de PostgreSQL, donde se guardarán los ejecutables, librerías y ficheros de configuración de PostgreSQL:



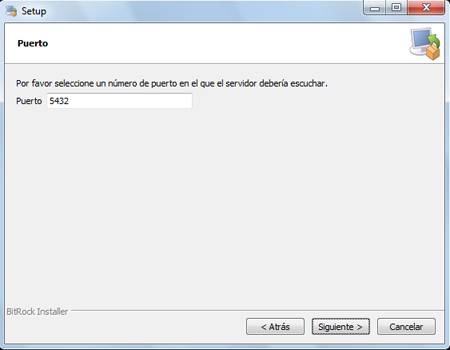
Indicaremos también la carpeta donde se guardarán los datos por defecto de PostgreSQL:



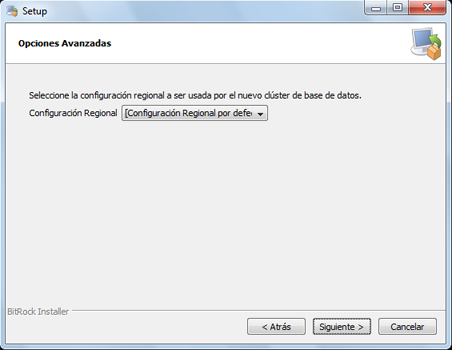
Introduciremos la contraseña para el superusuario "postgres" que será con el que iniciemos sesión para administrar la base de datos:



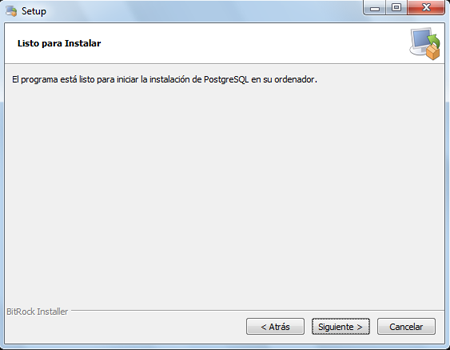
Introduciremos el puerto de escucha para la conexión con el servidor PostgreSQL, por defecto el 5432:



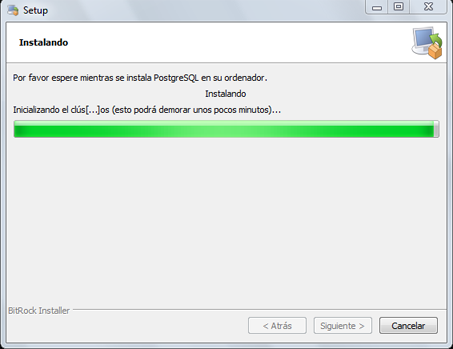
Seleccionaremos la configuración regional:



Pulsaremos "Siguiente" para iniciar la instalación definitiva del servidor PostgreSQL en [Microsoft Windows 7](http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=430):



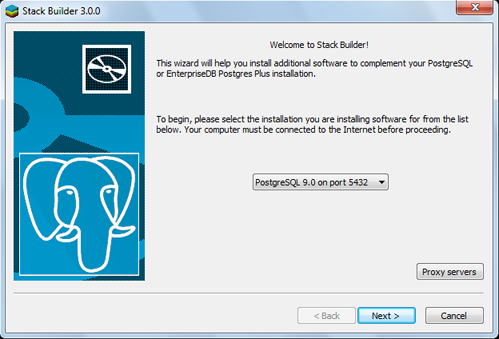
Se iniciará el asistente para instalar el motor de base de datos PostgreSQL, que creará las carpetas oportunas, copiará los ficheros necesarios y creará el servicio Windows para iniciar de forma automática el motor de base de datos:



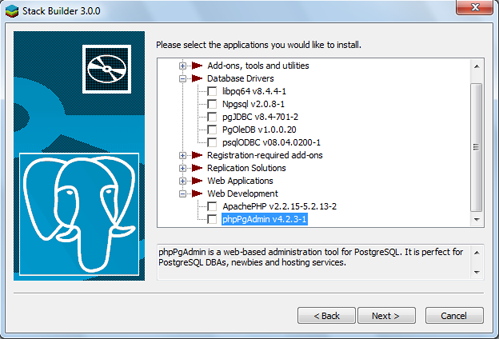
Una vez finalizada la instalación el asistente nos dará la posibilidad de ejecutar Stack Builder, aplicación que nos permitirá instalar otros componentes y herramientas para PostgreSQL:



Si hemos marcado la opción de Stack Builder, se iniciará, seleccionaremos "PostgreSQL 9.0 on port 5432" y pulsaremos "Next":



Seleccionaremos las aplicaciones, componentes y herramientas a instalar y pulsaremos "Next" (en nuestro caso cancelaremos Stack Builder pues no instalaremos más componentes):



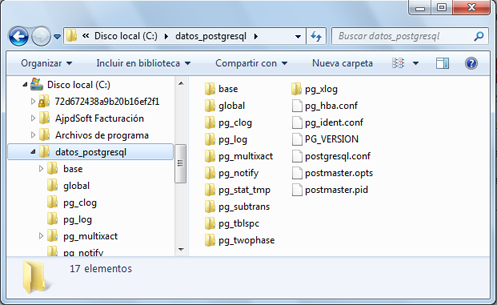
El asistente para instalar el servodpr PostgreSQL habrá creado un servicio que estará iniciado y en tipo de inicio automático llamado "postgresql-9.0":

## 

Y habrá creado la carpeta en archivos de programa "PosgreSQL" con las subcarpetas:

## 

Y la carpeta de datos:



Con esto habremos convertido nuestro equipo [Microsoft Windows 7](http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=430) en un servidor de base de datos PostgreSQL.

Si queremos que los equipos de nuestra red tengan acceso al servidor PostgreSQL y tenemos algún cortafuegos (firewall) deberemos abrir el puerto 5432. Y si queremos que se tenga acceso desde Internet al servidor PostgreSQL deberemos redireccionar (mapear) el puerto 5432 en el router o cortafuegos de nuestra empresa.