

JAVA SERVER FACES

- **¿Qué es?**

Java Server Faces es un marco de trabajo de interfaces de usuario del lado del servidor para aplicaciones Web.

- **Principales Componentes:**

- API e implementación de referencia para representar componentes de interfaz de usuario y manejar su estado, manejo de eventos, validación del lado del servidor y conversión de datos; también permite definir las reglas de navegación entre páginas.
- Librería de etiquetas JavaServer Pages, personalizadas para dibujar componentes de interfaz dentro de una página JSP.

- **¿Qué es una aplicación JavaServer Faces?**

Es una aplicación como cualquier otra, que se ejecuta dentro de un contenedor Servlet Java y típicamente contiene:

- JavaBeans, para contener datos y funcionalidades específicas de la aplicación.
- Oyentes de eventos.
- Páginas (JSP)
- Beans, clases de utilidad del lado del servidor.
- Librería de etiquetas personalizadas para dibujar componentes UI en una página.
- Librería de etiquetas personalizadas para representar manejadores de eventos, validadores y otras acciones.
- Validadores, manejadores de eventos y manejadores de navegación.

- **Estructura de las aplicaciones JSF.**

- **/JavaSource:** Fuentes de las clases Java.
- **/ant:** Scripts de creación, (build.xml)
- **/WebContent:** Archivos de la aplicación web usados por el servidor de aplicaciones.
- **/Web-INF:** archivos que se utilizan como parte de la rutina de la aplicación Web.
 - /classes
 - /lib : jsf-impl.jar, jsf-api.jar
 - web.xml
 - faces-config.xml
- **/pages:** archivos jsp para la presentación de la interfaz.

- **Pasos de desarrollo para una aplicación JSF**

1. Desarrollar los objetos del modelo: éstos son los objetos que contienen los datos.
2. Añadir las declaraciones del bean controlado al archivo de configuración de la aplicación
3. Crear las páginas
4. Definir la navegación entre páginas, ésta se define en el archivo faces.

- **Ciclo de vida estándar de procesamiento de peticiones**

- Reconstruir el árbol de componentes: la implementación JavaServer Faces construye el árbol de componentes de la página, conecta los manejadores de eventos y los validadores y graba el estado.
- Aplicar valores de petición: una vez construido el árbol, cada componente extrae su nuevo valor desde los parámetros de petición con su método decode. Entonces el valor es almacenado localmente en el componente.
- Procesar Validaciones: Durante esta fase, la implementación JavaServer Faces procesa todas las validaciones registradas con los componentes del

árbol. Examina los atributos del componente que especifican las reglas de validación y compara esas reglas con el valor local almacenado en el componente.

- Actualizar los valores del Modelo: una vez que se determina que el dato es válido, se configuran los valores del objeto del modelo correspondiente con los valores locales de los componentes.
- Invocar aplicación: durante ésta fase, la implementación JavaServer Faces maneja cualquier evento a nivel de aplicación, como enviar un formulario o enlazar a otra página.
- Renderizar la respuesta: durante ésta fase, la implementación JavaServer Faces invoca las propiedades de codificación de los componentes y dibuja los componentes del árbol de componentes grabado en el FacesContext.

- **Ciclo de vida de una página JavaServer Faces**

El ciclo de vida de una página JavaServer Faces, es similar al ciclo de vida de una página cualquiera, pero puede tener algunas variaciones dependiendo de si la página es generada como una aplicación JavaServer Faces, y si la respuesta del servidor da una página Faces, por eso se pueden dar los siguientes escenarios de procesamiento:

- Respuesta Faces: Una respuesta servlet que se generó mediante la ejecución de la fase renderizar respuesta del ciclo de vida de procesamiento de la respuesta.
- Respuesta No-Faces: una respuesta de un servlet que no incorpora componentes JavaServer Faces.
- Petición Faces: una petición servlet que fue enviada desde una Respuesta Faces previamente generada.
- Petición No-Faces: una petición que fue enviada a un componente de la aplicación que no es JavaServer Faces.

- **¿Qué se necesita para trabajar con JSF?**

Se necesita:

- JDK 1.4 o más avanzado.
- Tomcat 5.0, 4.1.x , o JBoss 3.2.x en adelante.
- Ant

- **Herramientas que ayudan a la creación y utilización de JSF:**

- MyEclipse: ésta herramienta nos permite generar aplicaciones J2EE, y generar aplicaciones JavaServer Faces.

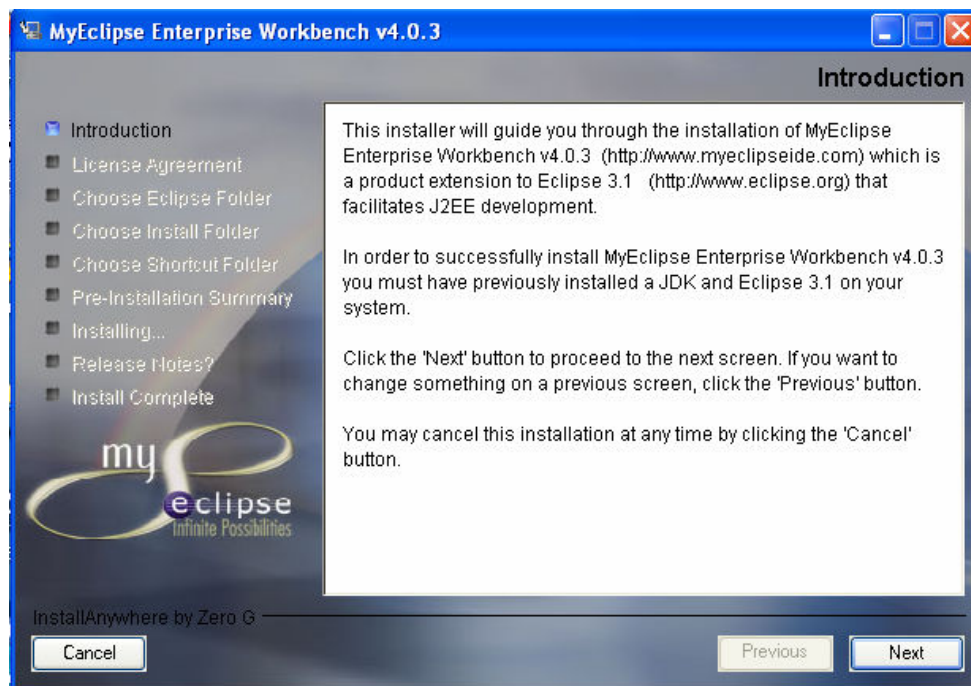
Para su instalación ver el Anexo.

En ésta dirección podemos encontrar un tutorial para la creación de un JSF usando MyEclipse:

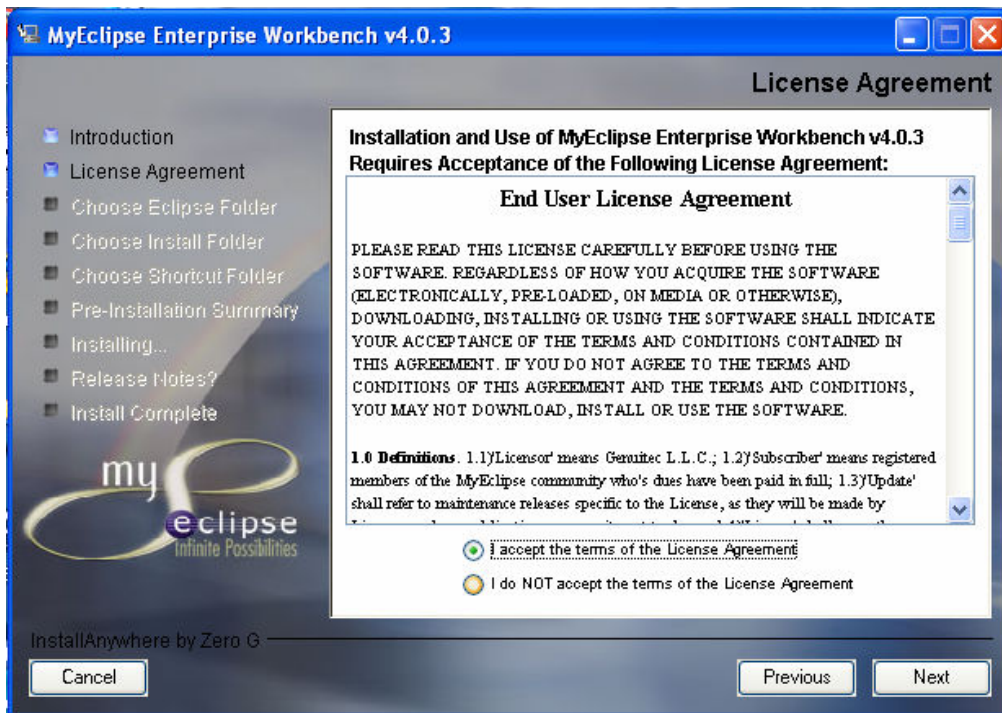
www.laliluna.de/first-java-server-faces-tutorial-es.html

ANEXO (INSTALACIÓN DE MYECLIPSE)

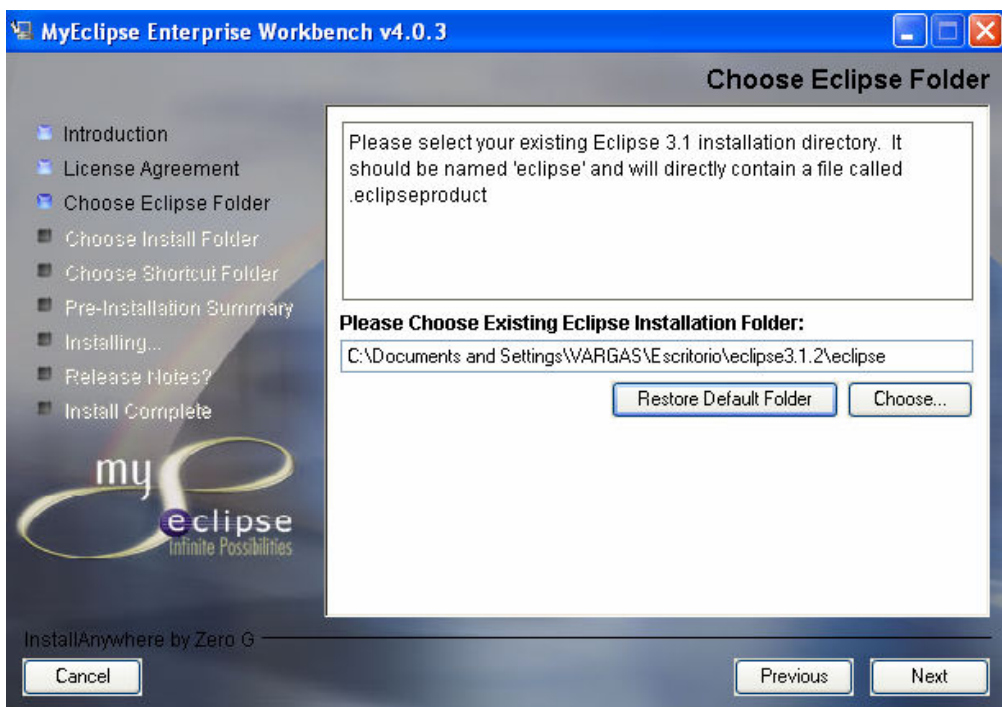
1. Registrarse para obtener una licencia de 30, en:
<http://www.myeclipseide.com/ContentExpress-display-ceid-10.html>
2. Aceptar la licencia.
3. Una vez que se tiene un usuario y contraseña, iniciar una sesión.
4. Seleccionar la versión que se desea descargar de MyEclipse.
5. Una vez descargada, instalarla como se muestra a continuación:
 - a. Ejecutar el archivo de instalación



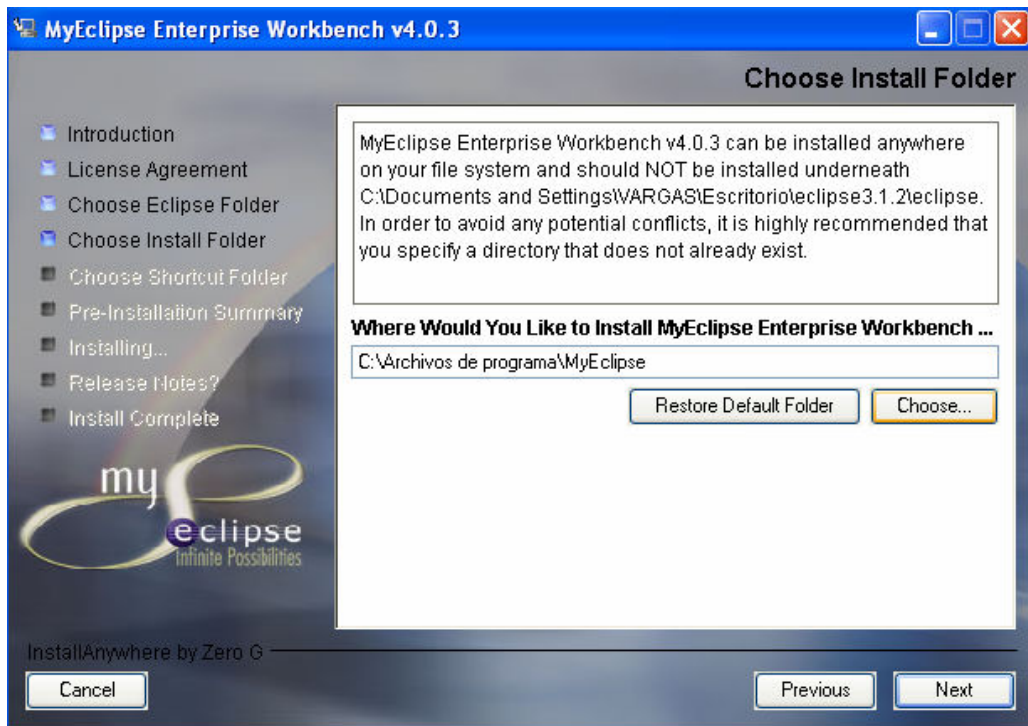
- b. Seleccionar Next



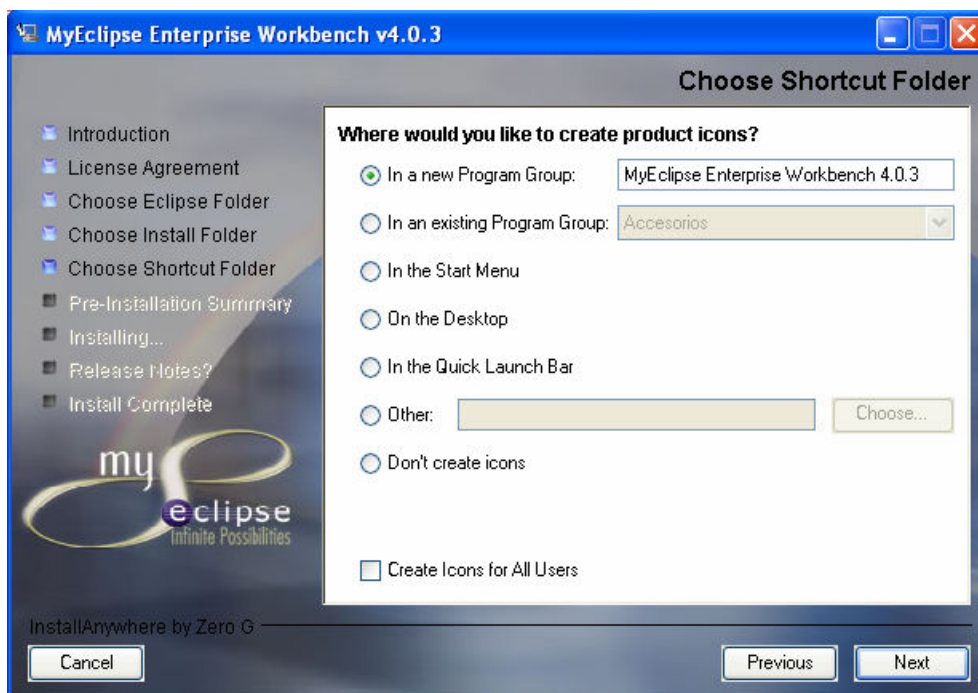
c. Aceptar la licencia



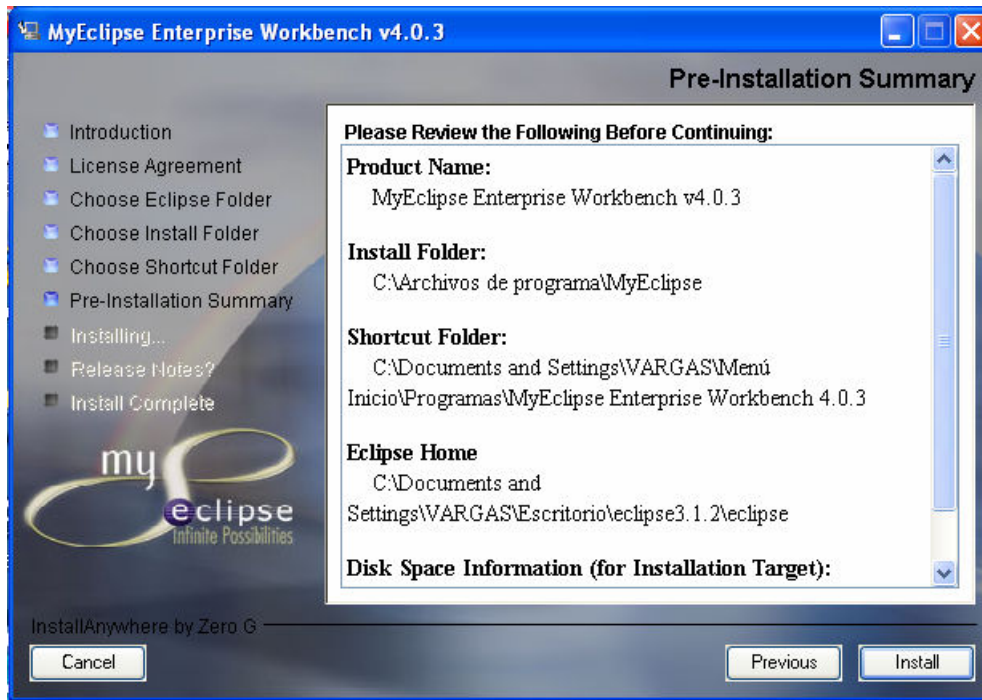
d. Seleccionar la carpeta donde se encuentra instalado eclipse, y seleccionar Next.



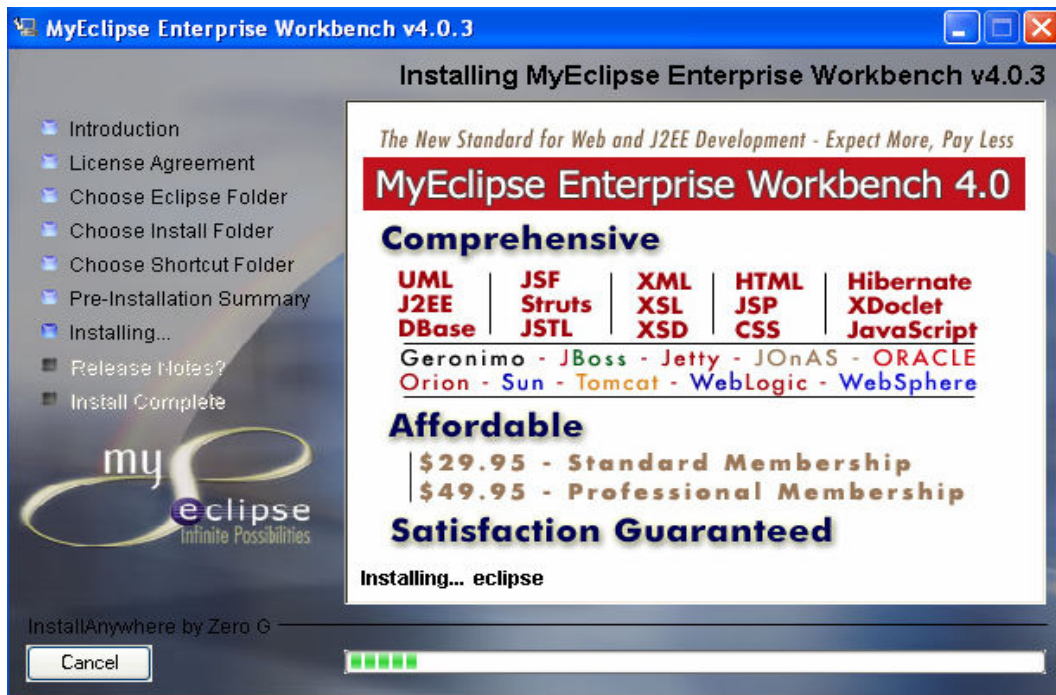
e. Seleccionar la carpeta donde se desea instalar MyEclipse

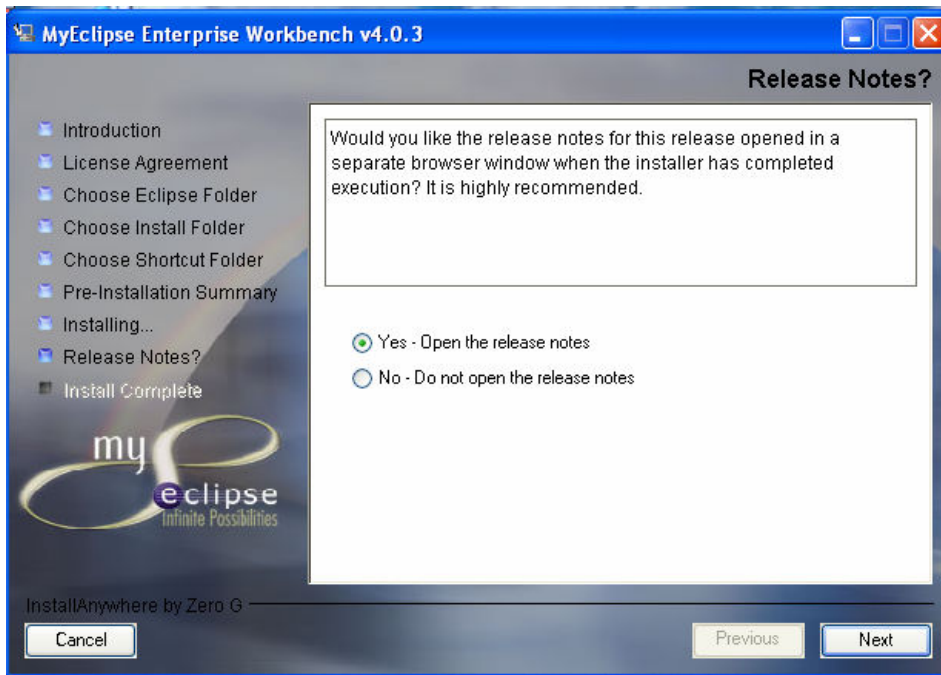


f. Llenar los campos solicitados y dar Next.

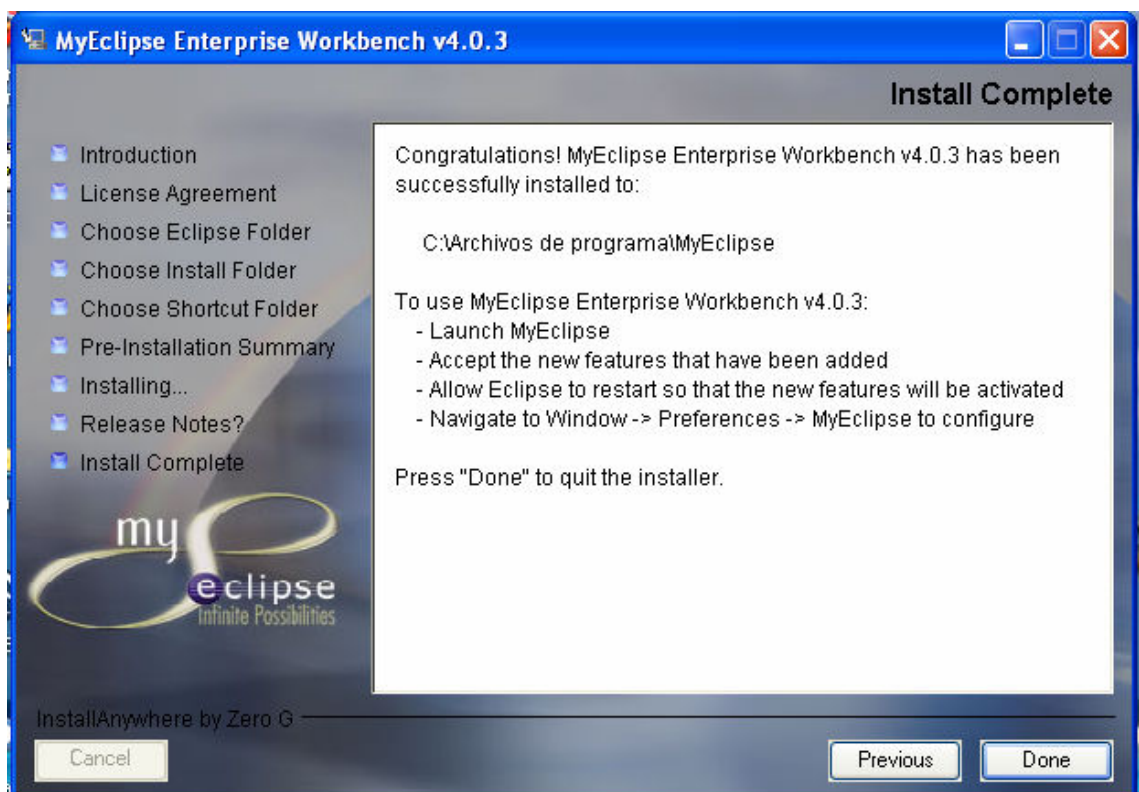


g. Seleccionar Install





h. Seleccionar Next



i. Hacer clic en Done.

MyEclipse ha sido instalado

Referencias

- <http://www.exadel.com/tutorial/jsf/jsftutorial-kickstart.html>
- <http://www.adictosaltrabajo.com/tutoriales/pdfs/jsf.pdf>
- http://www.programacion.com/java/tutorial/jsf_intro/
- www.jboss.org/wiki/Wiki.jsp?page=JBossFaces
- www.laliluna.de/first-java-server-faces-tutorial-es.html