

ICE Faces

Es un framework de código abierto para construir aplicaciones web con AJAX tipo RIA (Rich Internet Application).

Permite al programador incluir una serie de Ajax-tags en sus JSP o xhtml de tal manera que el código Ajax es generado por el propio framework automáticamente.

ICEFaces aísla completamente al desarrollador de AJAX. No hacen falta etiquetas especiales: se ponen los controles en la pantalla e ICEFaces se encarga de enviar sólo la información necesaria entre cliente y servidor. Es decir, ya no se envían los formularios a la antigua usanza, en un POST de HTTP, sino que sólo se envían los cambios que ha hecho el usuario del cliente al servidor, y los cambios en la pantalla del servidor al cliente. Además, con la inclusión de la librería Scriptaculous en ICEFaces, se dispone de arrastrar+soltar y de efectos (fundidos, parpadeos, apariciones, ...) para los controles.

Esta ficha presenta el contenido acerca de ICEFaces, se muestran a continuación enlaces directos a distintos aspectos:

- [Características](#)
- [Ventajas e inconvenientes](#)
- [Versiones recomendadas](#)
- [Requisitos e incompatibilidades](#)
- [Modo de empleo](#)
- [Enlaces de interés](#)
- [Buenas prácticas](#)
- [Aplicación de ejemplo](#)

Características.

ICEfaces es considerado un framework que integra funcionalidad AJAX y permite a los desarrolladores Java EE crear aplicaciones RIA (Rich Internet Applications) de una manera sencilla.

Las aplicaciones desarrolladas en ICEfaces no necesitan plugins de navegador o applets para ser vistas.

Estas aplicaciones están basadas en JavaServer Faces (JSF), así que permite el desarrollo de aplicaciones Java EE con la posibilidad de utilizar de forma fácil desarrollos basados en JavaScript.

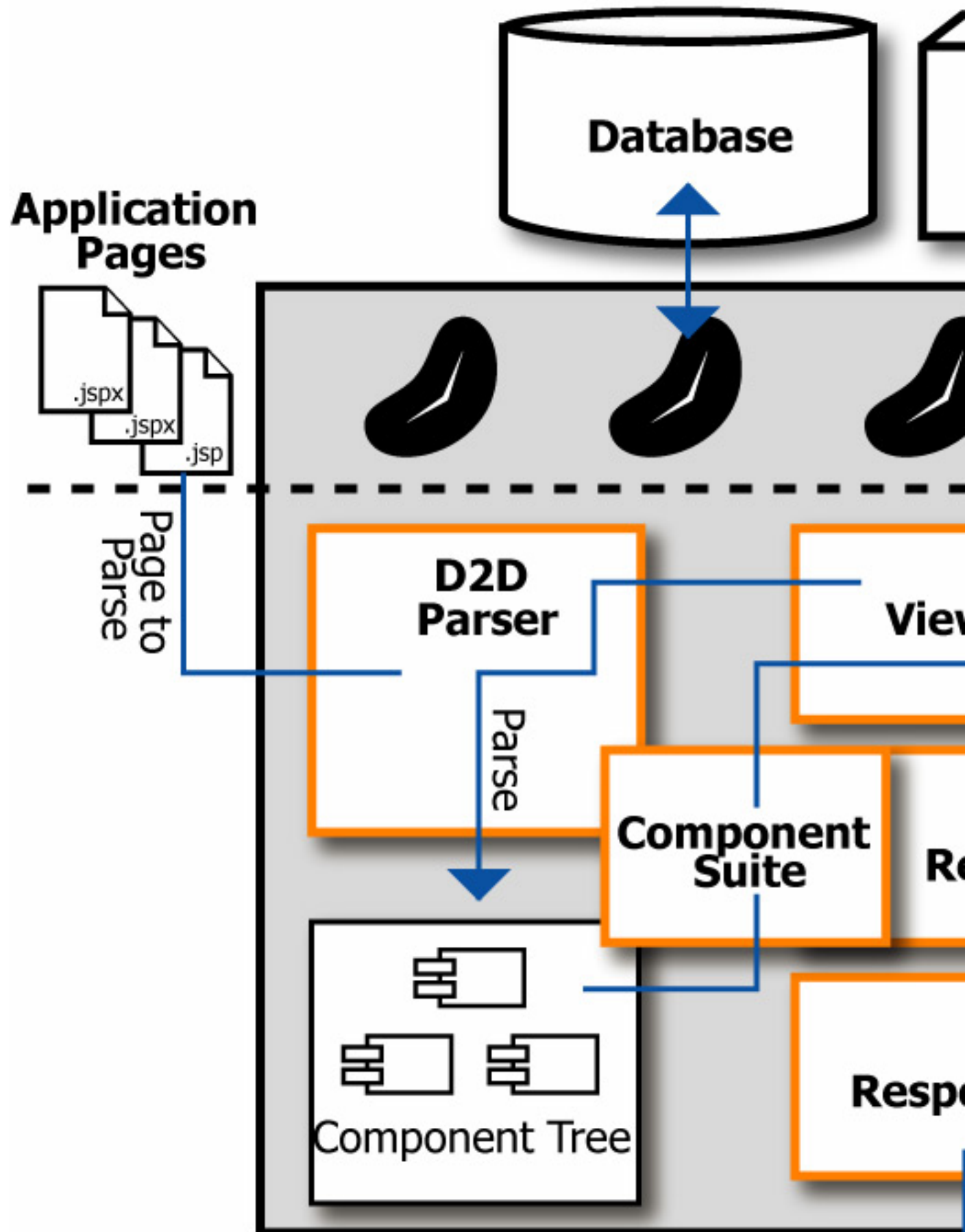
Entorno a AJAX han surgido varios frameworks (Prototype, DWR, GWT, ...) que, si bien aportaban facilidad de uso, no acababan de convencer a la comunidad de programadores. Algunos porque sólo eran clientes Javascript, otros porque, si bien integraban la parte de servidor con la de cliente, no eran realmente frameworks, sino librerías de comunicación. Además, no estaba claro cómo juntarlos con la arquitectura JEE.

Con la llegada de JSF, se empezó a vislumbrar posibilidades de integración. Si JSF permitía al desarrollador aislarse de la arquitectura web y ver sus aplicaciones como algo parecido a una aplicación de escritorio, debería entonces ser sencillo utilizar AJAX para hacer estos controles más funcionales. Y así fue, empezaron a aparecer AJAX4JSF, ICEFaces, Tobago, ...

MADEJA - ICEFaces

Sin embargo, de estas propuestas, ICEFaces fue una de las más acogidas ya que aísla completamente al desarrollador de AJAX. No hacen falta etiquetas especiales: se ponen los controles en la pantalla e ICEFaces se encarga de enviar entre cliente y servidor sólo la información necesaria.

Se presenta una figura con la arquitectura de una aplicación en JSF integrada con ICEFace:



Los principales elementos de la arquitectura ICEfaces incluyen:

- Persistent Faces Servlet: Las URLs con extensión ".iface" son mapeadas por el servlet 'Persistent Faces Servlet'. Cuando se realiza una petición de la página inicial en la aplicación, este servlet se hace responsable de la ejecución del ciclo de vida JSF para petición asociada.
- Blocking Servlet: Se encarga de la gestión de todas las peticiones de bloqueo y no-bloqueo después de las primeras páginas.
- D2D ViewHandler: Se encarga de establecer el Direct-to-DOM, incluyendo la inicialización de la 'DOM Respuesta Writer'. El ViewHandler también invoca al Parser para analizar el árbol de componentes JSF en la página inicial.
- Parseador D2D: Responsable del montaje de un componente de documentos JSP. El Parser ejecuta la etiqueta de JSP de procesamiento del ciclo de vida con el fin de crear el árbol, pero lo hace sólo una vez para cada página. La compilación del estándar JSP y el proceso de análisis no es compatible con ICEfaces.
- DOM Response Writer: Se encarga de la escritura en el DOM. También inicia la serialización DOM para la primera prestación, y desbloquea el DOM Updater para actualizaciones incrementales.
- DOM Serializer: Responsable de la serialización del DOM de la página inicial.
- DOM Updater: Se encarga de conjuntar las de las 'DOM mutations' en una única actualización DOM.
- Component Suite: Ofrece un conjunto de componentes 'rich JSF' con influencia AJAX y características del puente, proporcionando los elementos básicos para aplicaciones ICEfaces.
- Client-side AJAX Bridge: Responsable de la actualización DOM en curso generada por la solicitud y la respuesta del proceso. También es el encargado de centrar la gestión y de presentar el proceso.

Ventajas e inconvenientes con otros productos similares

Las ventajas del uso de ICEfaces para enriquecer las aplicaciones web de escritorio son numerosas. Debido a la cantidad de frameworks existentes, se procede a mostrar una serie de características diferenciadoras respecto a otros frameworks:

- Experiencia de usuario enriquecedora: crea una experiencia de usuario superior además de utilizar las ventajas de aplicaciones Java EE. Esto se consigue gracias a los componentes que vienen incluidos dentro de la distribución de ICEfaces.
- Está basado en código abierto: ICEfaces es un framework basado en Ajax bajo licencia de código abierto. La comunidad de desarrolladores de ICEfaces incluye cerca de 20.000 desarrolladores en 36 países.
- Basado en estándares: ICEfaces es una solución basada en Java, así que los desarrolladores pueden continuar trabajando de la misma forma que lo hacen. Hay multitud de plugins desarrollados para que ICEfaces sea integrado con multitud de IDEs Java.
- El Ajax es transparente: ICEfaces aporta a los programadores un desarrollo con mínimo esfuerzo en la sección JSF.
- Compatibilidad: ICEfaces soporta todos los servidores de aplicaciones, aporta plugins para los distintos IDEs y efectos javascript de librerías de cualquier empresa que haya desarrollado Ajax del mercado.
- Seguridad: ICEfaces es una de las soluciones Ajax más seguras del mercado. Es compatible con SSL, previene los scripts de cross-site, inyección de código malicioso. Es una solución Ajax basada en servidor, la cual no utiliza datos de usuarios, además es especialmente efectivo en la prevención de fallos en los submits de los formularios y el ataque SQL por inyección.
- Escalabilidad y clustering: El servidor asíncrono HTTP (AHS) aporta una alta escalabilidad para aplicaciones ICEfaces y pueden ser utilizadas por un gran número de usuarios concurrentes, además aporta despliegue en clúster (un requisito crítico que algunas soluciones no aportan).
- Carga de páginas incremental con edición de secciones y sin recargas de página completas.
- Se preserva el contexto del usuario durante la actualización de la página, incluyendo posición del foco y scroll.
- En aplicaciones de tiempo real, las recargas de páginas son asíncronas.

En una comparativa con un producto similar, se opta por compararlo con Rich Faces. Es la competencia de Rich Faces, tiene tantos o más componentes que ésta y visualmente tiene la misma calidad. La documentación posiblemente es peor que la de RichFaces. El modo en que RichFaces hace funcionar Ajax permite interactuar

con otras librerías e incluso añadirle Ajax a componentes que no lo tuvieran, mientras que ICEfaces limita a trabajar con sólo aquellos componentes para los que da soporte.

Versiones recomendadas

La versión más estable lanzada por la empresa propietaria de ICEfaces (IceSoft) es la versión 1.6 del framework. No obstante, ya se encuentra una nueva versión ICEfaces v1.7.0 con un conjunto de mejoras y arreglos (nuevo popup contextual, google maps, soporte para funciones adicionales...) aunque actualmente se encuentra en fase de pruebas y debe ser considerada no apta para su uso en producción.

Requisitos e Incompatibilidades.

Requisitos.

A continuación identificamos las plataformas que soportan el desarrollo basado en ICEfaces.

Integración con IDEs.

ICEfaces aporta integración básica con una alto número de interfaces de desarrollo adaptados por la comunidad de desarrolladores. Destacamos los siguientes:

- Eclipse (Web Tools Platform + JSF tools).
- Sun NetBeans.
- MyEclipse Enterprise Workbench.
- BEA Workshop Studio 3, Workshop Studio for JSF.
- Oracle JDeveloper 10g Studio Edition.

Servidores de aplicaciones.

ICEfaces proporciona integración con los siguientes servidores de aplicaciones J2EE:

- Apache Tomcat.
- BEA Weblogic Server.
- JBoss Application Server.
- IBM Websphere Application Server.
- Oracle Application Server Container for J2EE (OC4J).
- SAP NetWeaver.
- Sun GlassFish.
- Sun Java System Application Server.
- Webtide Jetty.

Portales y frameworks Java EE.

ICEfaces soporta los siguientes portales y frameworks de desarrollo JEE:

- Liferay Portal 4.3.
- JBoss Seam 1.3.
- JavaServer Faces (JSF) 1.1, 1.2.
- Facelets.
- Spring Web Flow.

Navegadores soportados.

Se ha comprobado la compatibilidad de ICEfaces en los siguientes navegadores:

- Internet Explorer 6, 7
- Firefox 1.x, 2.0
- Mozilla 1.7.x and Netscape 7.x
- Safari 1.3
- Opera 9.x

Incompatibilidades.

En este punto describiremos los errores más comunes detectados en la release que se recomienda en MADEJA.

- La conexión asíncrona entre el navegador y el servidor se pierde debido a una interrupción de la conexión, la recarga de la página restablece el estado previo de la aplicación.
- La inclusión en ICEfaces de contenido desde JSP no es compatible con MyFaces.
- Los intentos de recuperar un parámetro de la Request usando la siguiente sintaxis no está soportado (devuelve nulo):

```
FacesContext.getCurrentInstance().getExternalContext().getRequestParameterMap().get(paramName);
```

Una alternativa puede ser usar la sintaxis siguiente:

```
((HttpServletRequest)FacesContext.getCurrentInstance().getExternalContext().getRequest()).getParameter(paramName);
```

* Si se especifica el parámetro de configuración `com.icesoft.faces.connectionLostRedirectURI` en el fichero `web.xml` y no se coloca una comilla simple a la URL, el resultado es una cadena de conexión corrupta.

Modo de empleo

Instalación

A continuación se va a describir el proceso para realizar la instalación del plugin que se encargará de la integración con eclipse. Para realizar esta integración, se han utilizado los siguientes componentes, los cuales se pueden descargar directamente desde la web de ICEfaces (sección download):

- Sistema Operativo: Guadalinex V4.0
- Eclipse SDK versión 3.3.1.1
- Librerías de ICEfaces versión 1.6.1 para eclipse (ICEfaces-1.6.1-libs-Eclipse.zip).
- Paquete con el plugin de integración de eclipse (ICEfaces-Eclipse-IDE-v3.0.0.zip).

Una vez que todos estos paquetes se encuentran disponibles, se procede a descomprimirlos en una carpeta para utilizarlos en la instalación del plugin para eclipse:

- Crear una carpeta que será la que contendrá los archivos descomprimidos de estos archivos:

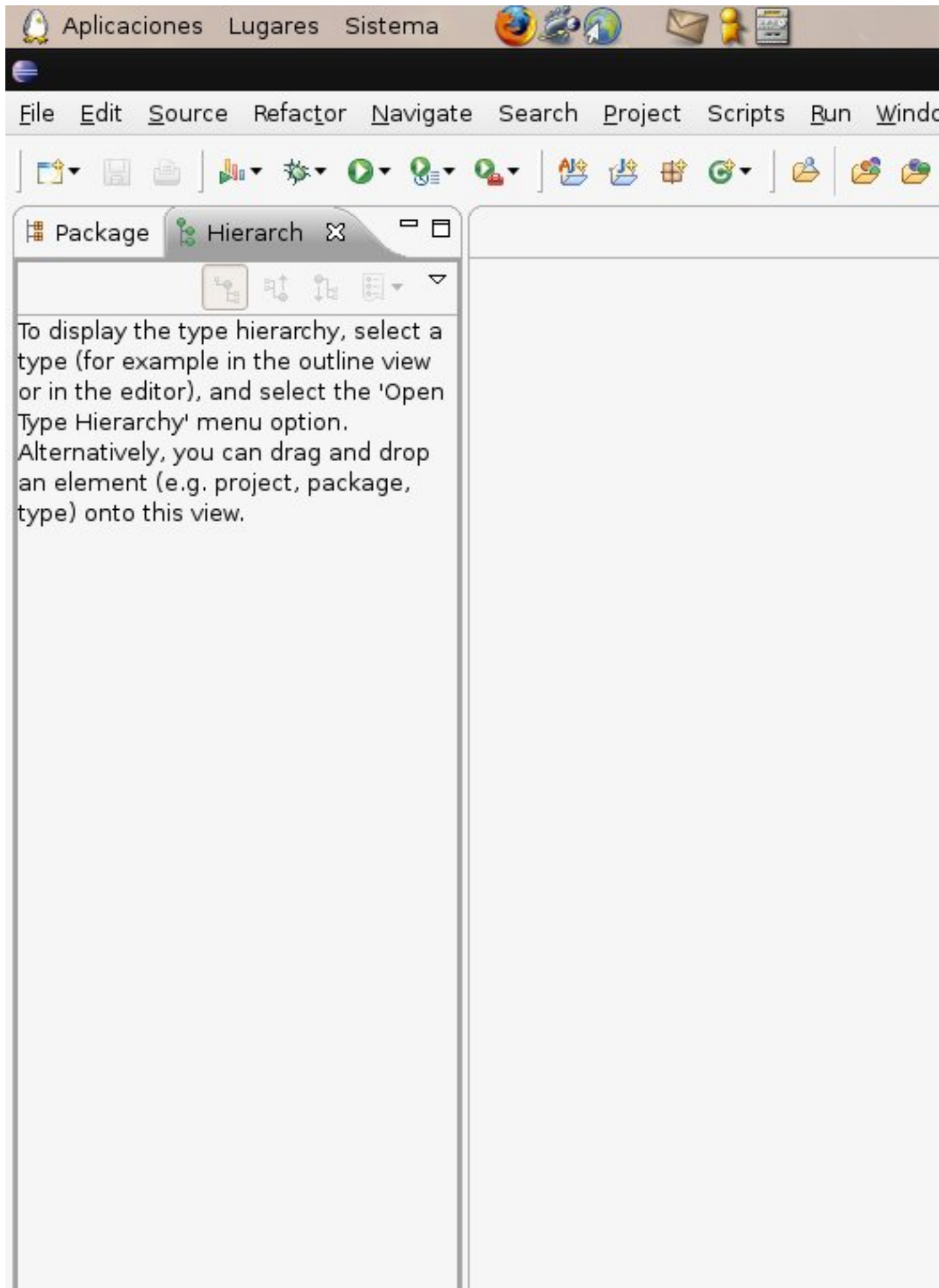
```
mkdir EclipseLocalSite
```

MADEJA - ICEFaces

* Descomprimir los archivos en esa carpeta:

```
unzip ICEfaces\1.6.1\libs\Eclipse.zip  
unzip ICEfaces\Eclipse\IDE\v3.0.0.zip
```

* Arrancar el eclipse y pulsar sobre Help/Software Update/Find and Install



* Pulsar sobre nuevas características a instalar.



Install/Update

Feature Updates

Choose the way you want to search for features to install

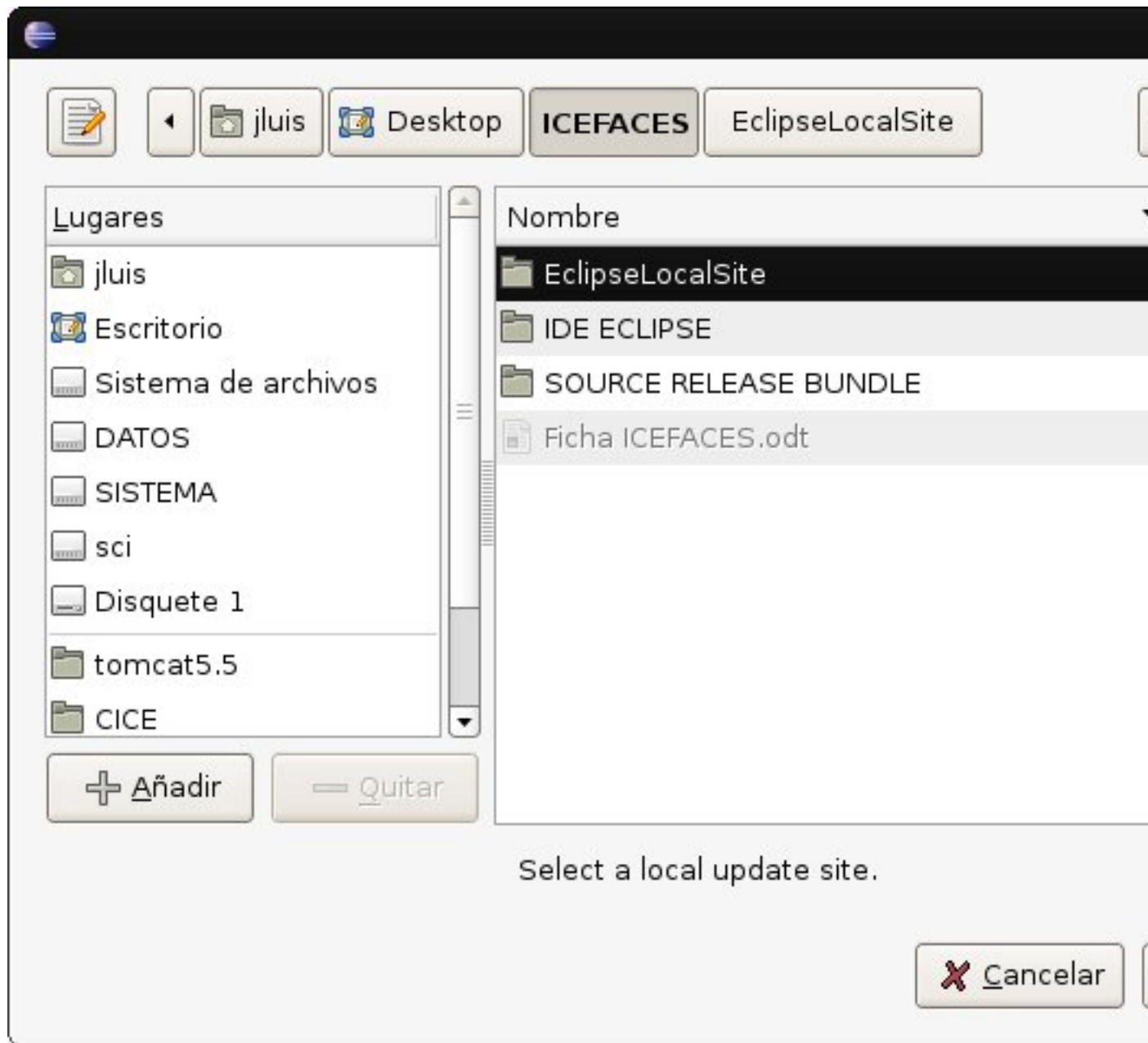
☐ Search for updates of the currently installed features

Select this option if you want to search for updates of the features you already installed.

☒ Search for new features to install

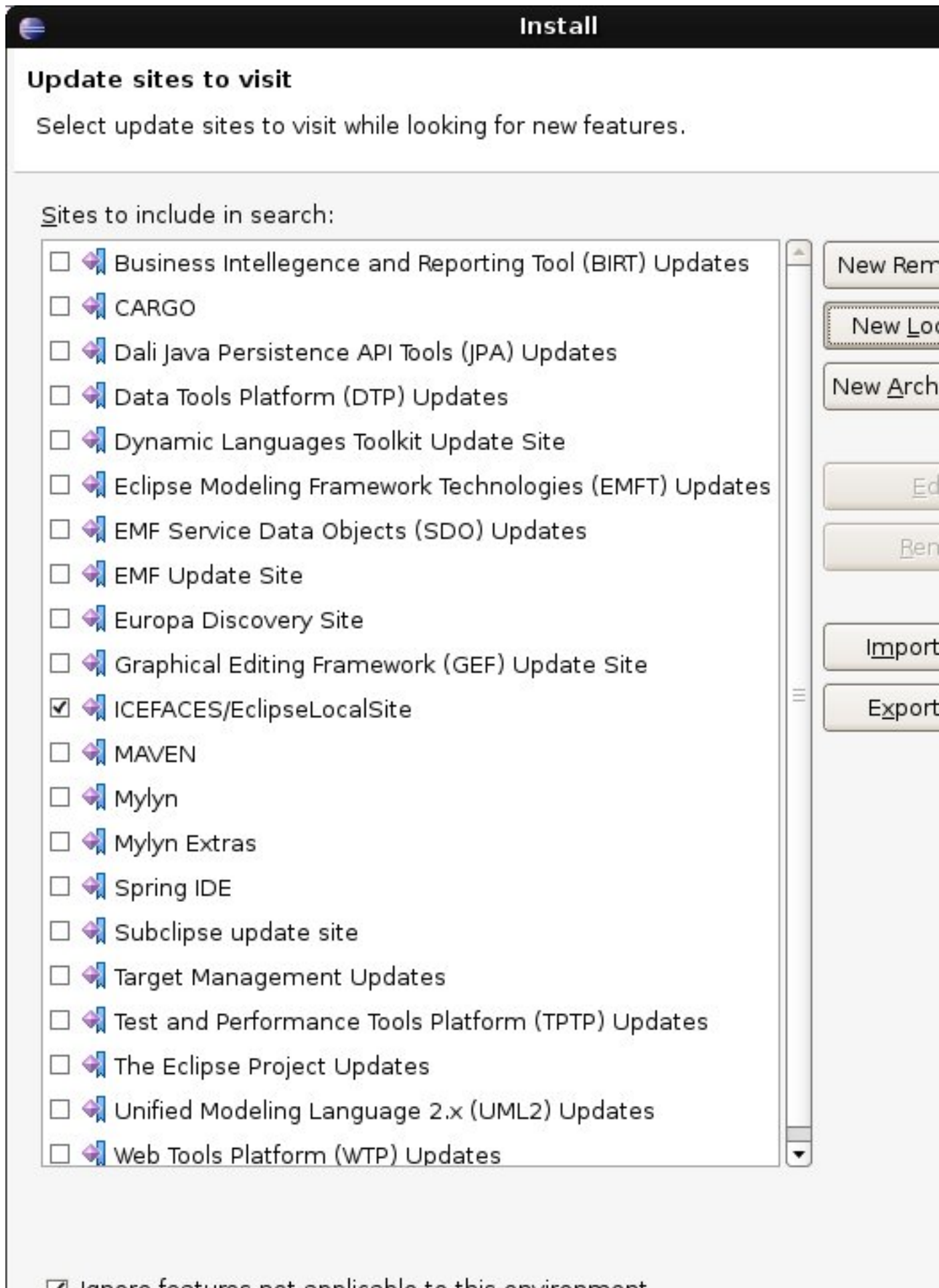
Select this option if you want to install new features from existing or new update sites. Some sites may already be available. You can add new update site URLs to the list.

* Tras esto, pulsar sobre New local site y seleccionar la carpeta en la que se descomprimieron los dos archivos anteriores:

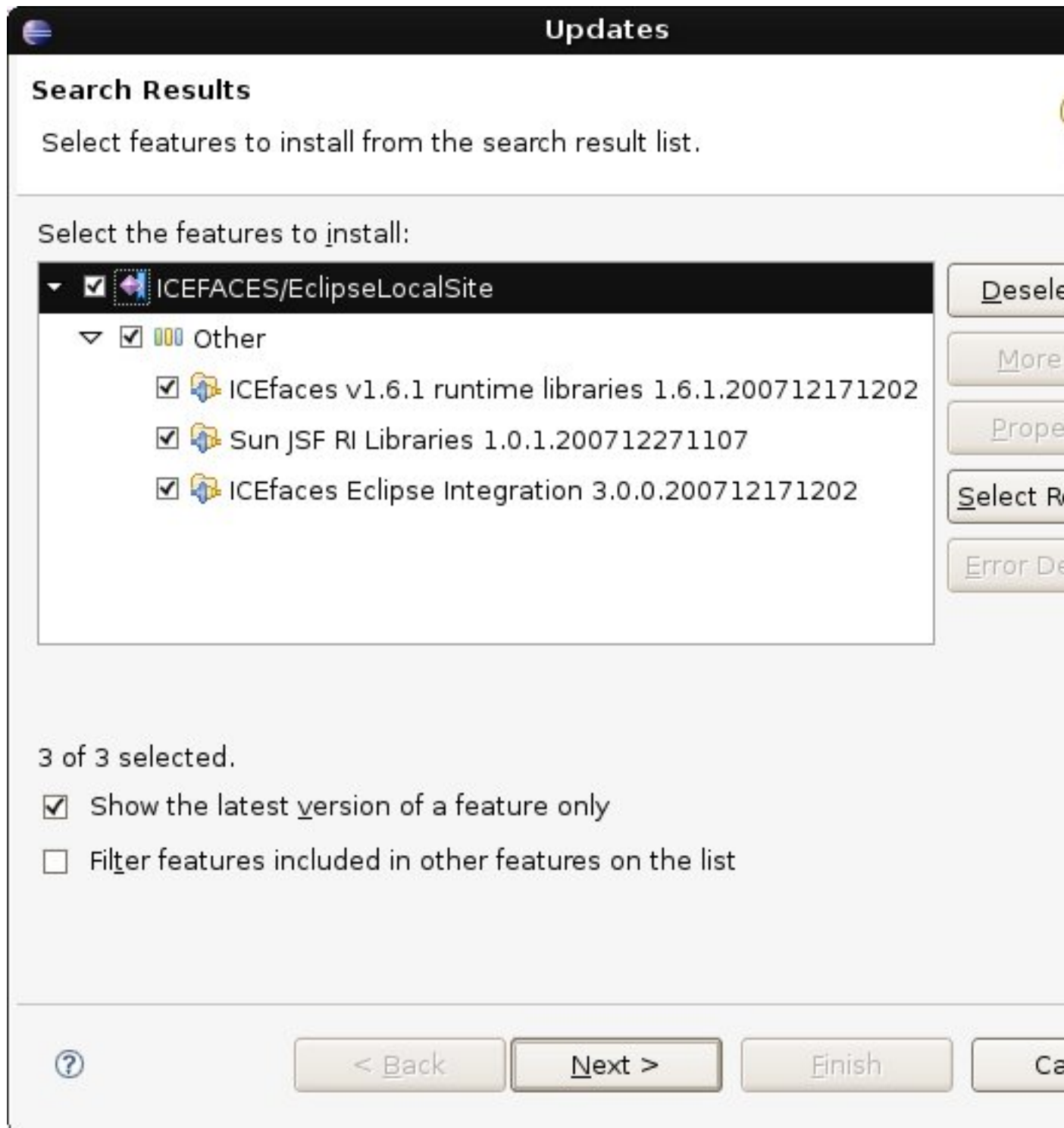


* Indicar el nombre con el que identificar a este repositorio y pulsar ok.

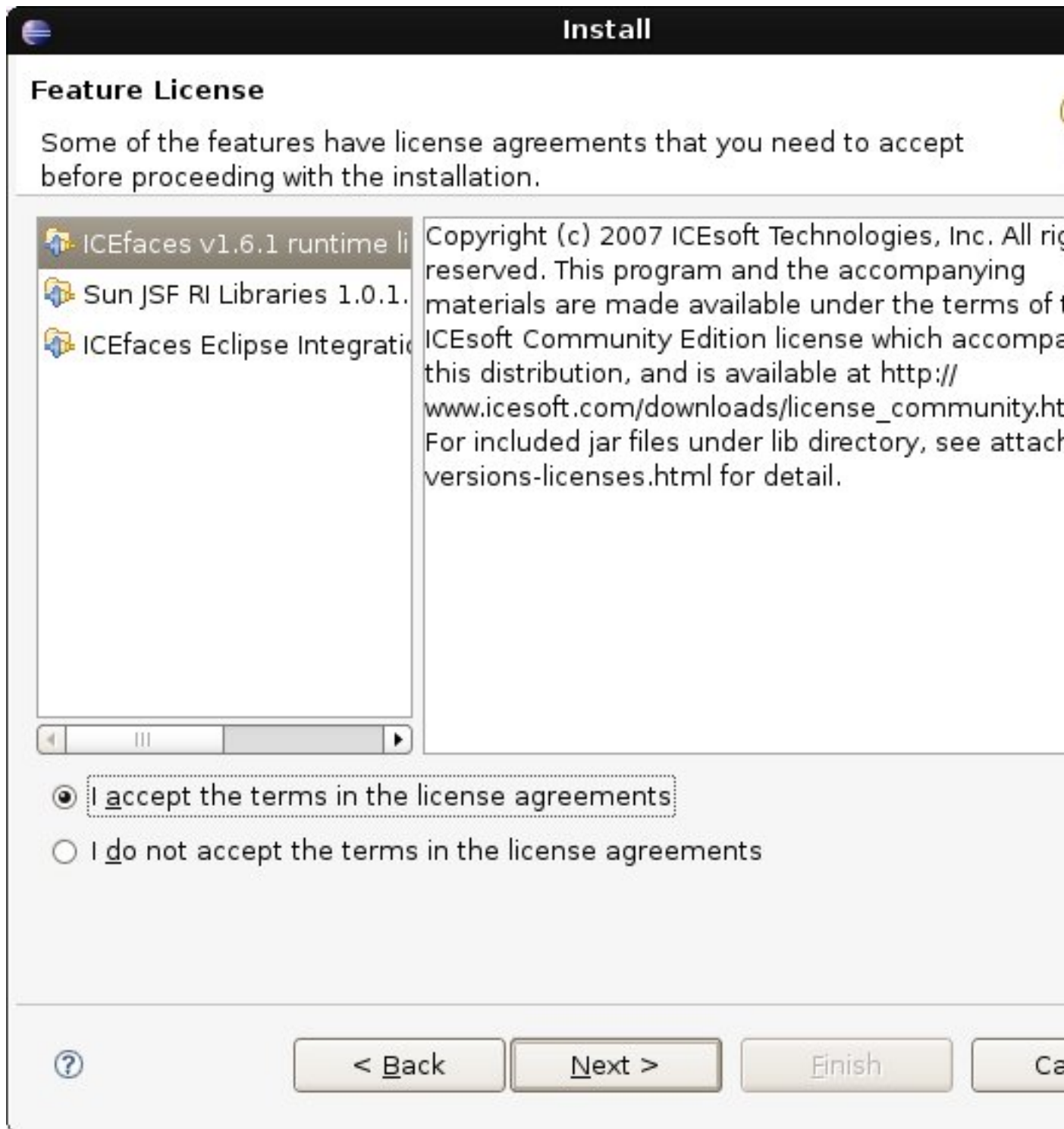
- Seleccionar el nuevo sitio agregado y pulsar en Finish:



* Tras realizar el paso anterior, aparecerá una nueva ventana en la que se mostrarán los plugins que se pueden instalar desde la ubicación seleccionada, acto seguido se procede a seleccionar todo y pulsar Next:



* A continuación aceptar el contrato de licencia y pulsar sobre siguiente. Por último finalizar la instalación:



* Cuando se acabe de instalar el plugin, será necesario reiniciar el eclipse, tras lo cual ya se encontrará el plugin instalado.

Enlaces de interés

Página oficial:

<http://www.icefaces.org/main/home/index.jsp>

Página de descarga de ICEfaces:

<http://www.icefaces.org/main/downloads/os-downloads.iface>

Integración con eclipse:

<http://www.icefaces.org/JForum/posts/list/6688.page>

Enlaces a otros

Integración de JSF 1.2, Facelets e ICEfaces en Tomcat 6:

http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=tomcat6_icefaces

Documentos de referencia.

Página con las características de la última release:

http://www.icefaces.org/docs/v1_6_2/ReleaseNotes.html

Documentos de formación

Documentación oficial de ICEfaces:

- ICEfacesAsyncServer.pdf
- ICEfacesDevelopersGuide.pdf
- ICEfacesGettingStartedGuide.pdf

Buenas prácticas

A continuación se muestran algunas buenas prácticas en la validación de formularios:

Habilitando el botón de submit

Hay dos opciones principales cuando se habla de habilitar, o no, botones de submit: ancho de banda y usabilidad. No es recomendable abusar de un reenvío parcial de submit para verificar por ejemplo que el nombre del usuario introducido en un típico formulario de entrada a una aplicación es correcto. En las aplicaciones web típicas, cuando se realiza un submit, se verifica que el campo usuario ha sido rellenado, esto puede realizarse

mediante ICEfaces. La otra opción es implementar una validación directa del usuario, pero si el usuario no esta habituado al uso de este tipo de aplicaciones, esta validación puede resultar engorrosa si no ven un botón de login en la página.

Mensajes de error y validación

Suele ser una fuente de errores el que los mensajes de validación o error desaparezcan debido a que esos campos no tengan el atributo immediate situado a true. Este problema es debido a que sólo se hace submit de una parte del formulario.

JSF permite que se modifiquen el texto de los mensajes de error. A continuación se muestra cómo se hace:

- Se indica la lista de mensajes que se pretenden reemplazar, por ejemplo para validar la longitud:

```
javax.faces.validator.LengthValidator.MAXIMUM_MESSAGE_ID = Tu mensaje
```

* Por último sólo es necesario agregar en el faces-config.xml el bundle de los mensajes:

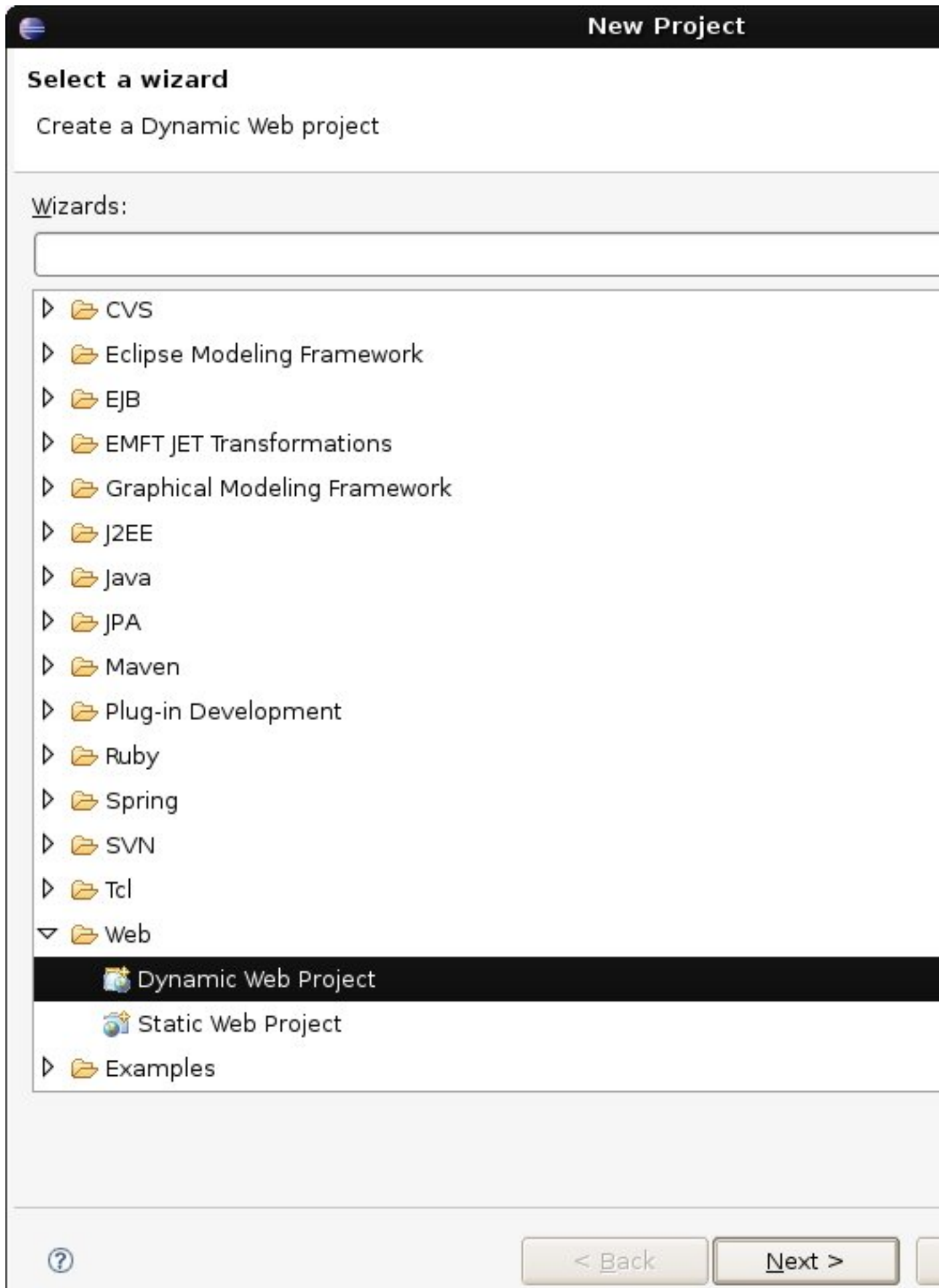
```
<application>
<message-bundle>com.yourcompany.resources</message-bundle>
</application>
```

Aplicación de ejemplo

Creando un proyecto de ejemplo

En esta sección se realiza una aplicación de ejemplo utilizando eclipse y el plugin de ICEfaces instalado en pasos anteriores. A continuación se procede a realizar un tutorial paso a paso para realizar un nuevo proyecto con características de ICEfaces:

- Ejecutar el eclipse, seleccionar un workspace y crear un nuevo proyecto del tipo Dynamic Web Project:



* Indicar el nombre del proyecto, seleccionando en la sección de Configurations que va a ser un proyecto ICEfaces:



New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise

Project name: EjemploICE

Project contents:

☒ Use default

Directory: /home/jluis/Desarrollo/workspace/EjemploICE

Target Runtime

Apache Tomcat v5.5

Configurations

ICEfaces Project

Enables the project with ICEfaces capabilities.

EAR Membership

☐ Add project to an EAR


EAR Project Name: EAR



< Back

Next >









* Seleccionar siguiente y marcar las facets del proyecto:


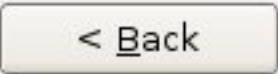

 **New Dynamic Web Project**

Project Facets

Select the facets that should be enabled for this project.

Configurations: ICEfaces Project

Project Facet	Version
 Axis2 Web Services	
<input checked="" type="checkbox"/>  Dynamic Web Module	2.3
<input checked="" type="checkbox"/>  ICEfaces	3.0
<input type="checkbox"/>  ICEfaces Facelets	3.0
<input checked="" type="checkbox"/>  Java	1.4
<input type="checkbox"/>  Java Persistence	1.0
<input checked="" type="checkbox"/>  JavaServer Faces	1.1
<input type="checkbox"/>  WebDoclet (XDoclet)	1.2.3

* Indicar la configuración del módulo web y hacer clic en siguiente:



New Dynamic Web Project

Web Module

Configure web module settings.

Context Root:

Content Directory:


Java Source Directory:



< Back

Next >

* Por último seleccionar las librerías que se van a utilizar, en este caso se indicaría que serán las Sun JSF ir v1.2_04 y agregar la librería referida a ICEfaces versión 1.6.1:


 **New Dynamic Web Project**


JSF Capabilities
Add JSF capabilities to this Web Project


JSF Libraries:

☐ Server Supplied JSF Implementation
☒ Sun JSF RI v1.2_04

Component Libraries

▶  ICEfaces Facelets library v1.6.1

▶  Sun EL RI v2.1

▶  Xerces v1.3.1

>

<

>>

<<

New...

JSF Configuration File:

/WEB-INF/faces-config.xml

JSF Servlet Name:


Faces Servlet

JSF Servlet Classname:

javax.faces.webapp.FacesServlet

URL Mapping Patterns:

/faces/*



< Back

Next >

Por último, indicar que las librerías de ICEfaces serán desplegadas con la aplicación, para realizar esto marcamos el check asociado a estas librerías en el wizard que apareció. Con esto la configuración del proyecto se daría por finalizada:



New Dynamic Web Project

ICEfaces Capabilities

Add one of ICEfaces libraries to this JSF Project.

Note:

1. Only one ICEfaces library can be selected.
2. Xerces library is required on JVM 1.4.x. only.
3. EL(Expression Language) RI library is not required on Tomcat6 and JavaEE5

ICEfaces libraries :



Deploy	Library Name
<input checked="" type="checkbox"/>	ICEfaces library v1.6.1




< Back

Next >

Creando una página de ejemplo










En esta sección se procede a crear una página de ejemplo y a mostrar algunas de las características que ofrece ICEfaces.

- Lo primero es abrir el proyecto de ejemplo denominado EjemploICE, a continuación pulsar sobre la carpeta WebContent, pulsar el botón derecho del ratón y seleccionar New y página JSP.
- En la ventana que aparece, seleccionar el nombre de la nueva página (con extensión .jspx) y su ubicación.


 **New JavaServer Page**

JavaServer Page
Create a new JavaServer Page.

Enter or select the parent folder:


  
▼  EjemploICE
  .settings
 ▶  build
  src
 ▶  **WebContent**
 ▶  Servers

File name:



MADEJA - ICEFaces

* Pulsar sobre siguiente y seleccionar el patrón para la página que va a utilizarse. Seleccionar New ICEfaces Facelets JSPX file o New ICEfaces JSPX File. Tras seleccionar la plantilla realizar clic sobre finalizar.



New JavaServer Page

Select JSP Template

Select a template as initial content in the JSP page.


☒ Use JSP Template

Templates are 'New JSP' templates found in the [JSP Templates](#) preference page.

Name	Description
New ICEfaces Facelets JSPX File (.jspx)	ICEfaces Facelets
New ICEfaces JSP File (.jsp)	ICEfaces JSP page
New ICEfaces JSPX File (.jspx)	ICEfaces JSPX page
New JSP File (html)	JSP with html markup
New JSP File (xml)	JSP with xml markup

Preview

```
<?xml version="1.0" encoding="${encoding}" ?>
<jsp:root version="1.2"
  xmlns:jsp="http://java.sun.com/JSP/Page"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ice="http://www.icesoft.com/icefaces/component">
<jsp:directive.page contentType="text/html; charset=${encoding}"
<f:view>
```

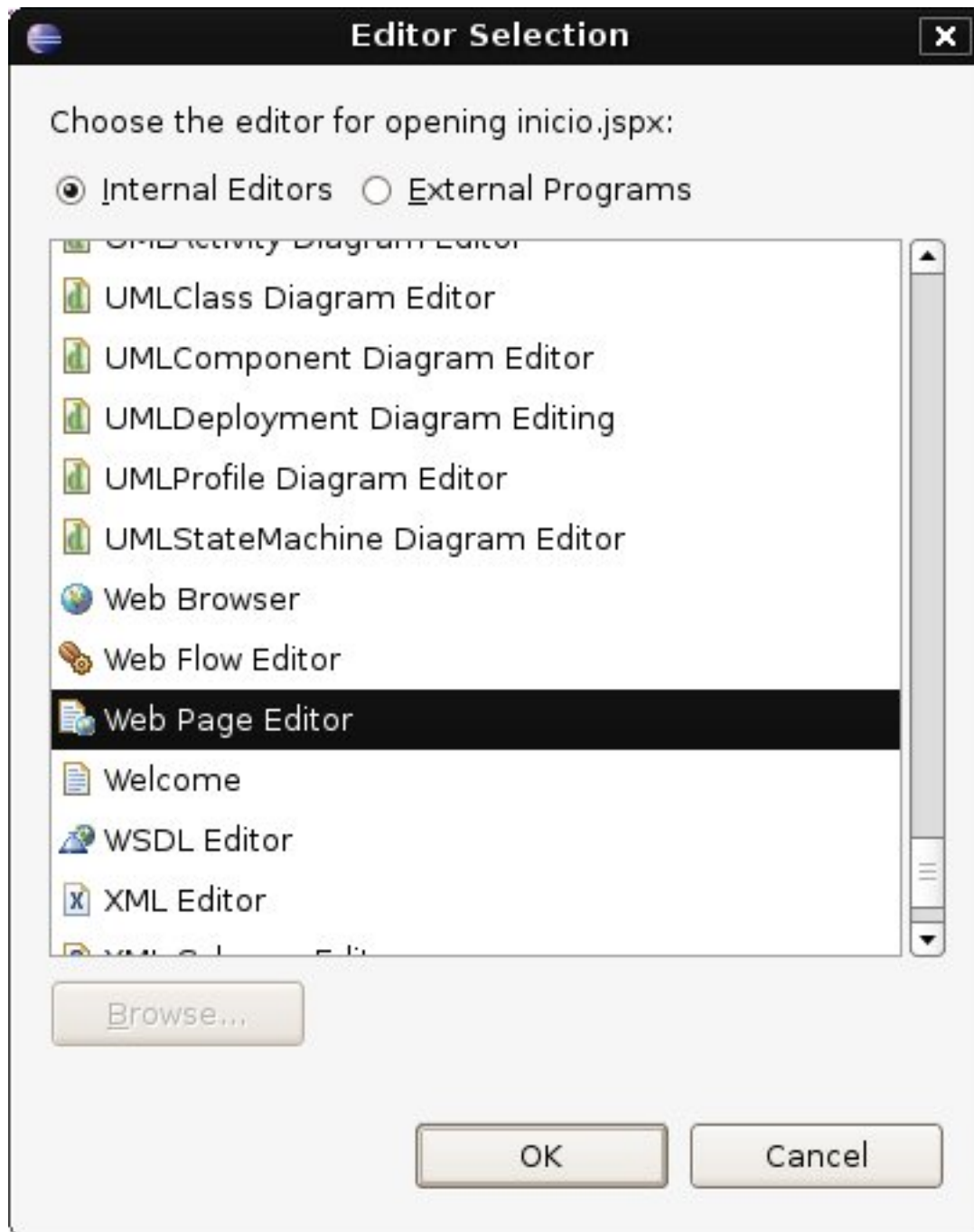


< Back

Tras ésto, el proyecto ya contaría con una página con características ICEfaces. Este proyecto puede ser desplegado en cualquier servidor para ir observando los resultados de las acciones que realizadas. Acto seguido se la da funcionalidad a la página creada.

Una vez que se tenga una página creada, utilizar un editor visual con funcionalidad 'drag and drop' para ir agregando los componentes deseados e ir viendo el resultado de manera visual utilizando un editor WYSIWYG.

Este editor no se encuentra por defecto asociado a las páginas ICEfaces, para habilitarlo habrá que pulsar sobre la página a editar, haciendo clic con el botón derecho y seleccionando ##Open With## / ##Other##. En la ventana que aparece, seleccionar ##Internal Editor## y ##Web Page Editor##.



Tras esto se obtendrá el editor que sigue, en el que se disponen de una paleta con la que podrán pincharse y arrastrarse componentes a su ubicación de una manera visual:

inicio.jspx

Tab1	Tab2	Tab3
Thank you for using ICEfaces.		
<div>Tab1 Container1</div>		
<div>Tab1 Container2</div>		
<div>Tab1 Container3</div>		



Barra de progreso:

0 %

```
<?xml version="1.0" encoding="UTF-8" ?>
<jsp:root version="1.2"
  xmlns:jsp="http://java.sun.com/JSP/Page"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ice="http://www.icesoft.com/icefaces/component">
<jsp:directive.page contentType="text/html; charset=UTF-8" page
<f:view>
  <ice:outputDeclaration doctypeRoot="HTML" doctypePublic="-//
  <html>
    <head>
      <title><ice:outputText value="ICEfaces, Ajax for Java EE"
      <ice:outputStyle href="http://css.xmlhttp.com/xp/css" />
    </head>
    <body>
```

