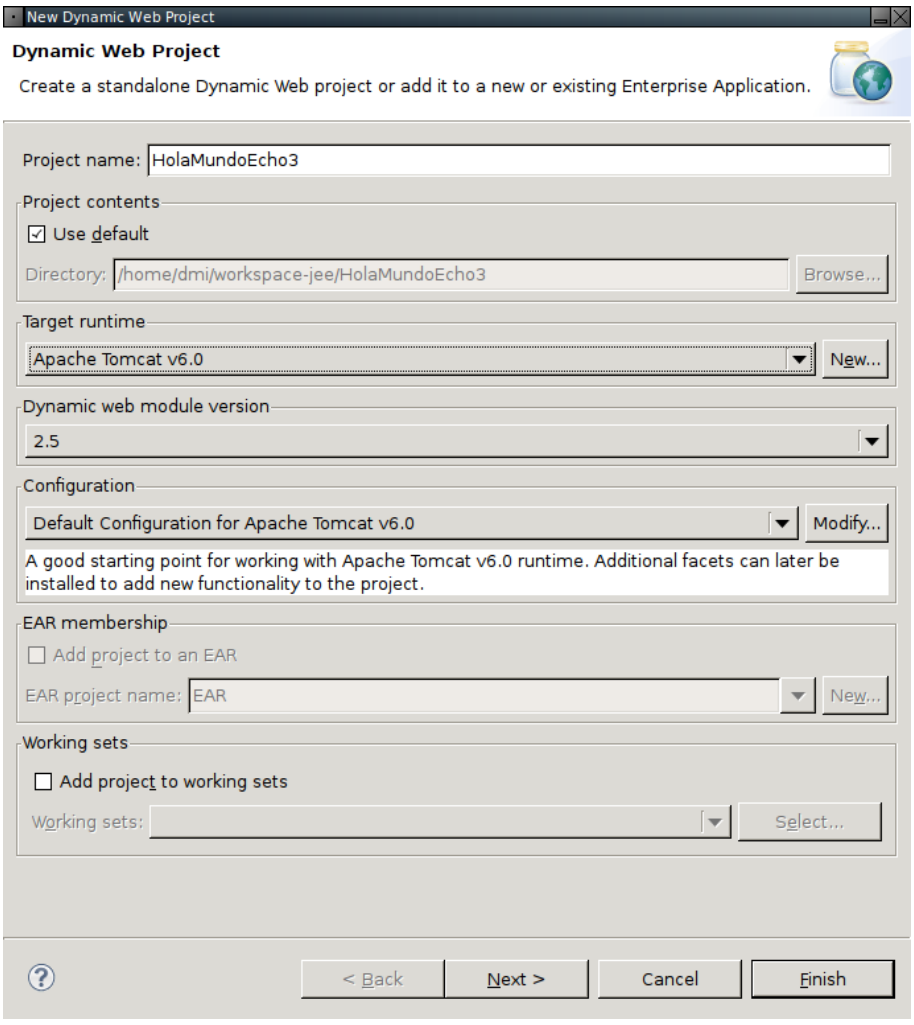
Introducción

El presente documento es una guía rápida de instalación de alguna herramienta particular. De seguro existen otras formas de realizar el proceso de instalación, pero esta es la que mejor le ha funcionado al autor a lo largo de los años. Para utilizar el presente documento se requieren conocimientos mínimos de linux, su estructura básica de directorios, el uso de la consola y las instrucciones básicas de shell. Además, se espera que el lector tenga suficiente sentido común para adaptarse a cualquier diferencia existente entre el procedimiento que se describe en este documento y la distribución o versión de linux utilizada.

PASO 1:

Crear un proyecto WEB. Click en:

File->New->Dynamic Web Project" para que salga el asistente:



Se debe introducir el nombre y es necesario asegurarse que el “target runtime” esté definido correctamente. Si ya está funcionando BloodTime o LCA el “target runtime” debería estar configurado correctamente según los pasos del tutorial “tutorial-desarrollo-lca-bloodtime”.

Una vez creado el proyecto debería tener la siguiente estructura en el “Project Explorer” o el “Package Explorer”:

Es importante tener en mente esta estructura porque nos referiremos a ella en los siguientes pasos.

PASO 2:

Descargar Echo3 de la siguiente dirección:

http://echo.nextapp.com/site/echo3/download

En general, también se puede bajar de echo2go aquí:

http://download.nextapp.com/downloads/echo3go/

En donde se consiguen “builds” diarios del SVN.

En cualquier caso, las páginas principales de Echo3 y Nextapp (la compañía detrás de Echo3) son: http://echo.nextapp.com/ y http://nextapp.com/ respectivamente.

En general sólo es necesario descargar la biblioteca principal de echo: NextApp_Echo.3.0.b8.tgz, aunque también puede ser útil tener los extras en algunos casos: NextApp_Echo_Extras.3.0.b8.tgz y el “file transfer” en otros casos: NextApp_Echo_FileTransfer.3.0.b7+.zip (para hacer uploads y downloads).

PASO 3:

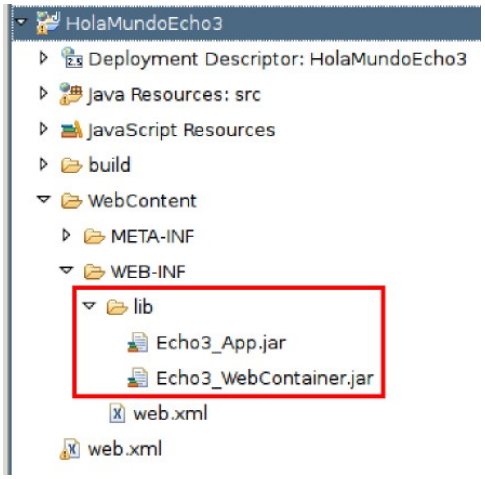
Una vez descargados estos archivos se deben descomprimir usando el comando tar (o zip según sea el caso):

```
tar zxvf NextApp_Echo.3.0.b8.tgz
```

Y se deben localizar los .”jar” de Echo3:

```
./JavaLibraries/Echo3_WebContainer.jar
./JavaLibraries/Echo3_App.jar
```

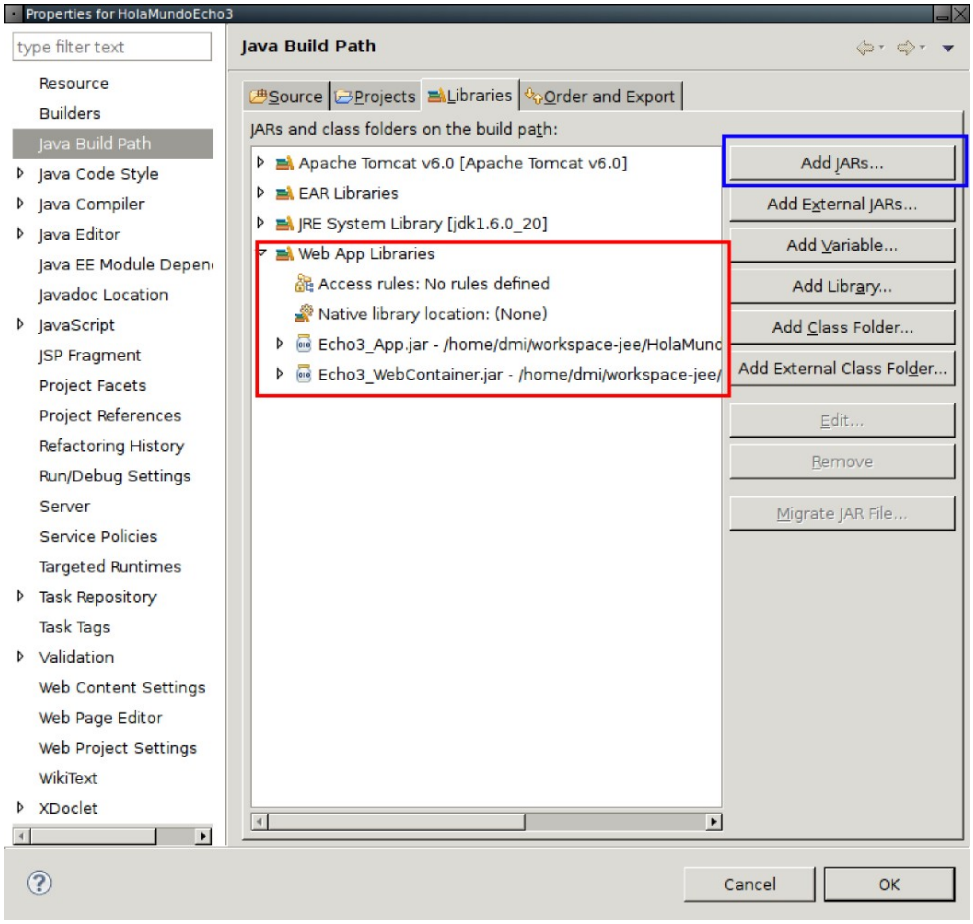
Estos jars se deben copiar o mover al directorio WebContent/WEB-INF/lib del proyecto HolaMundoEcho3 recién creado, de forma que quede:



En caso de problemas:

Eclipse-JEE, en teoría, por defecto añade los jars al build path, en cualquier caso, si hay problemas de compilación o el servidor tomcat no encuentra las clases de Echo3 (ClassNotFoundException) es conveniente revisar el build path de la siguiente forma:

“Seleccionar el Proyecto->Botón Derecho->Properties->Java Build Path->Libraries” y se debería ver algo como esto:



En rojo están marcadas las librerías de Echo3 incluidas por el eclipse-jee (eclipse las incluye automáticamente al estar ubicadas en WebContent/WEB-INF/lib). Sin embargo, en caso de problemas, también se las puede añadir manualmente utilizando el boton “Add JARs” (rectángulo azul) y seleccionando los archivos jar correspondientes (Ojo, esto en casos extremos, tratar de que funcione por defecto como se ha explicado en la sección principal del presente paso.

PASO 4:

Ahora es necesario hacer la aplicación en Echo3, para esto editaremos el archivo web.xml de la siguiente forma:

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
      xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
      id="WebApp_ID" version="2.5">
      <display-name>HolaMundoEcho3</display-name>

      <servlet>
            <servlet-name>HolaMundoServlet</servlet-name>
            <servlet-class>com.tutorial.holamundo.HolaMundoServlet</servlet-class>
      </servlet>

      <servlet-mapping>
            <servlet-name>HolaMundoServlet</servlet-name>
            <url-pattern>/holamundo.echo</url-pattern>
      </servlet-mapping>

</web-app>
```

La etiqueta `servlet` define el nombre “lógico” del servlet dentro del contenedor (etiqueta `servlet-name`) y la clase de Java que implementa el servlet (etiqueta `servlet-class`). La etiqueta `servlet-mapping` relaciona un patrón de URL (etiqueta `url-pattern`) con el nombre lógico del servlet (etiqueta `servlet-name`) que servirá las peticiones cuando el contenedor de servlets detecte el patrón.

En general, lo que se logra, es que cada vez que el servidor tomcat recibe una petición del estilo:

http://el-nombre-del-servidor:8080/holamundo.echo

Entonces tomcat se encarga de invocar al servlet implementado por la clase

`com.tutorial.holamundo.HolaMundoServlet`

Esto en realidad no tiene nada que ver con Echo3 y se puede encontrar en casi cualquier tutorial de servlets existente. Lo importante en Echo3 es que cada aplicación WEB escrita con Echo3 debe tener un servlet asociado para poder funcionar.

PASO 5:

Ahora es necesario crear el servlet y la aplicación en Echo3. El servlet es una clase ubicada en el paquete `com.tutorial.holamundo` y que se llama `HolaMundoServlet`. El código es el

siguiente:

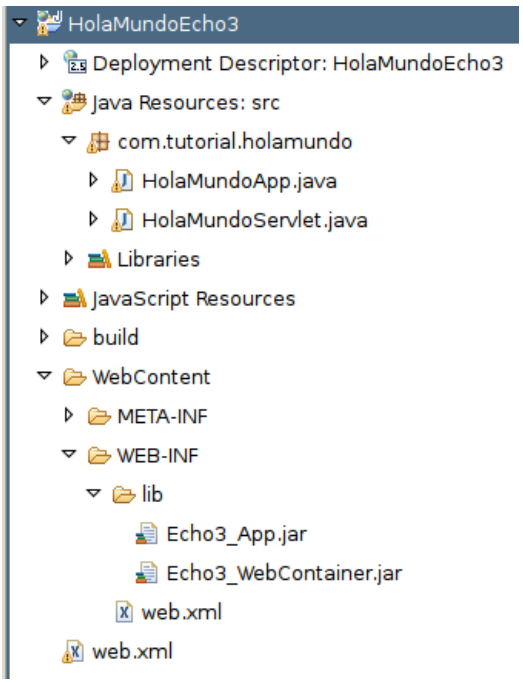
```
package com.tutorial.holamundo;

import nextapp.echo.app.ApplicationInstance;
import nextapp.echo.webcontainer.WebContainerServlet;

// El servlet de echo siempre debe heredar de WebContainerServlet
public class HolaMundoServlet extends WebContainerServlet {

  // En este método retorna la ApplicationInstance que
  // representa toda la aplicación
  public ApplicationInstance newApplicationInstance() {
    return new HolaMundoApp();
  }
}
```

El servlet está creando una instancia de tipo ApplicationInstance y la está retornando (marcado en negrita), esa clase está en el mismo paquete:

```
package com.tutorial.holamundo;

import nextapp.echo.app.ApplicationInstance;
import nextapp.echo.app.ContentPane;
import nextapp.echo.app.Label;
import nextapp.echo.app.Window;

public class HolaMundoApp extends ApplicationInstance {

  public Window init() {

    // ----------------------------------------
    // Construye una ventana (del navegador)
    // ----------------------------------------

    Window window = new Window();
    window.setTitle("Hola Mundo");
```

```
    // ----------------------------------------
    // Construye un panel raiz y se lo asigna
    // a la ventana
    // ----------------------------------------

    ContentPane contentPane = new ContentPane();
    window.setContent(contentPane);

    // ----------------------------------------
    // Construye la GUI y la añade al panel
    // raiz recien construido
    // ----------------------------------------

    Label label = new Label("¡¡¡Hola Mundo!!!");
    contentPane.add(label);

    return window;
  }
}
```

Si todo está correcto la estructura del proyecto ahora debería ser:

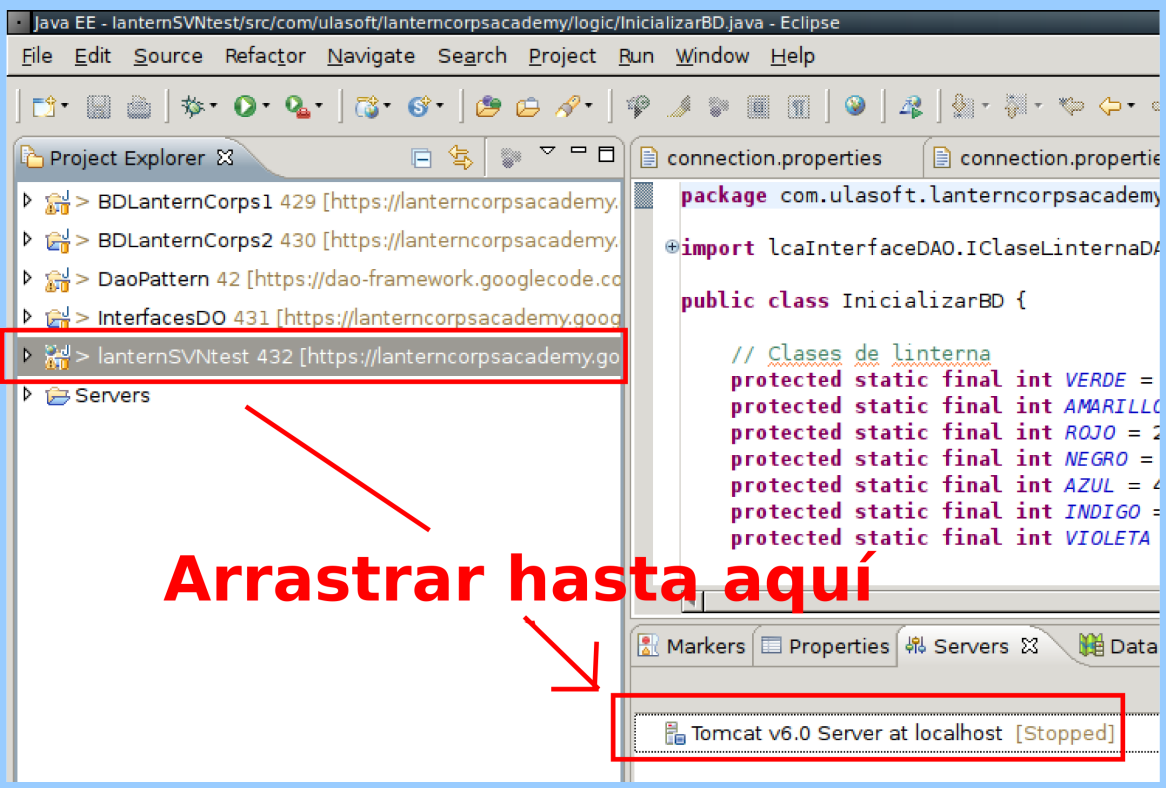


PASO 7:

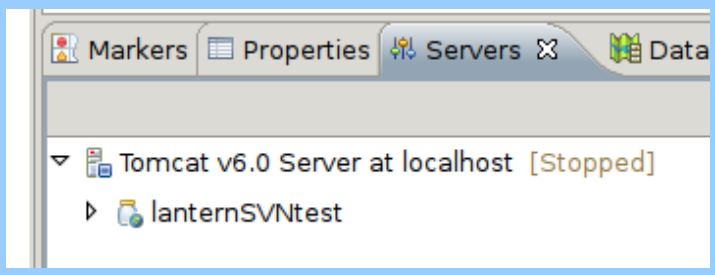
Repetir el Paso 11 y el Paso 12 del tutorial para poner a correr LCA y BloodTime (tutorial-desarrollo-lca-bloodtime.odt) pero para el proyecto Echo3Tutorial en lugar de LCA y BloodTime (Aplicar sentido común aquí). A continuación se reproduce la información de dicho tutorial:

tutorial-desarrollo-lca-bloodtime, PASO 11:

Ahora es necesario asociar el proyecto principal de LCA (lanternSVNtest) con el servidor que acabamos de crear, esto se hace simplemente ***seleccionando y arrastrando*** el proyecto lanternSVNtest desde el Project Explorer hasta el servidor que acabamos de crear:



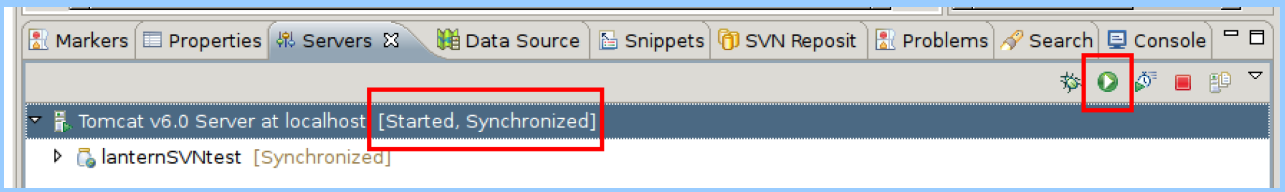
Y la vista Servers debe quedar:



Nota: En el caso de Blood Time es necesario arrastrar el proyecto “BloodTimeProject”

tutorial-desarrollo-lca-bloodtime, PASO 12:

Ahora es necesario arrancar el servidor, para esto hacemos click en el servidor que acabamos de crear en la vista Servers y luego hacemos click en el botón de ejecutar a la derecha:



Si todo sale bien, es posible que se vean algunos mensajes en la consola de eclipse, y luego el estado del servidor pasará de “Stopped” a “Started, Synchronized” tal como se muestra en la figura.

PASO 8:

Finalmente, es posible correr la aplicación. Para esto se abre un navegador y se apunta a la dirección:

http://127.0.0.1:8080/HolaMundoEcho3/holamundo.echo

Debería verse lo siguiente: