

**Laboratorio de Java - 75.079**

**Estructura de la Información / Diseño de Estructuras de Datos**

## **Tipos abstractos de datos (TADs)**



## Índice de contenidos

1.Introducción.....	3
2.Entorno CLI.....	4
2.1.Instalación de Java.....	4
2.2.Variables de entorno.....	4
2.3.Ejecución del EP1 sobre CLI con CLASSPATH.....	6
2.4.Ejecución del EP1 sobre CLI sin CLASSPATH.....	6
3.Entorno IDE: Eclipse.....	7
3.1.Descarga e instalación.....	7
3.2.Inicio.....	7
3.3.Instalación de la biblioteca de TADs.....	8
3.4.Ejecución del EP1 sobre Eclipse.....	11



## 1. Introducción

El documento presente tiene como objetivo explicar de forma clara y concisa como configurar el entorno de trabajo para poder usar la librería de TADs que se utiliza en las asignaturas *Estructura de la información*, y *Diseño de estructuras de datos*.

Existen dos formas posibles para utilizar la librería de TADs: mediante la línea de comandos (CLI) del sistema operativo, o bien usando un entorno de desarrollo integrado (IDE) como es Eclipse. A continuación se expone como configurar ambos entornos, ejemplificado con el EP1 (Ejercicio Práctico 1) entregado en el aula de teoría.



## 2. Entorno CLI

### 2.1. Instalación de Java

Para poder ejecutar la librería de TADs desde la línea de comandos (CLI) es preciso haber instalado previamente Java en nuestro ordenador. La web desde donde se puede descargar la última versión de Java es: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Como se puede ver, en esta web existe la posibilidad de descargar Java Development Kit (JDK) o bien Java Runtime Environment (JRE). En nuestro caso bajaremos JDK ( $\geq$  versión 1.5), ya que permite compilar el código fuente de nuestros programas; JRE en cambio solamente permite ejecutar binarios de Java, previamente compilados en otro sitio.

El proceso de instalación está documentado en la misma página web, según el sistema operativo de que se disponga.

### 2.2. Variables de entorno

Para que sea posible compilar y ejecutar programas Java desde CLI, es importante tener correctamente configuradas las variables de entorno PATH y CLASSPATH:

- La variable PATH se ocupa de informar al sistema operativo sobre las rutas de acceso (paths) que contienen aquellos programas que habitualmente usamos. El objetivo de añadir al PATH la ubicación de los binarios del JDK es permitir ejecutar comandos como java, javac, javadoc, etc. desde cualquier carpeta de nuestro disco. La variable PATH no es exclusiva del JDK, ya que es usada también por el propio sistema operativo y otras aplicaciones.

Ejemplo con Windows: si se ha instalado Java en C:\Java, y PATH=C:\WINDOWS\SYSTEM32, se añade al PATH la ubicación de los ejecutables Java: PATH=C:\WINDOWS\SYSTEM32;C:\Java\bin

- Por otro lado, la variable CLASSPATH indica a Java dónde están ubicadas nuestras clases. Habitualmente añadiremos como valor un . (punto) : este punto indicará a las aplicaciones de JDK que busque dentro del directorio donde estamos ubicados; si no fuera así, para poder referenciar las clases Java tendríamos que utilizar rutas absolutas. Si queremos que nuestros programas Java tengan consciencia de las librerías Java .jar, también las especificaremos dentro del CLASSPATH.

Ejemplo con GNU/Linux: se quiere añadir la ubicación de la librería compilada de TADs, tads.jar, además del punto indicador del directorio actual en el que estamos ubicado: CLASSPATH=/uoc/lib\_cat/tads.jar:.

- En entornos Windows es habitual definir también la variable JAVA\_HOME. Es una variable que no



usaremos para ejecutar ninguna de las pruebas indicadas en este documento, pero que aconsejamos tenerla definida por si la requerís en un futuro. Por ejemplo, se puede definir la variable JAVA\_HOME y condicionar el valor de PATH con JAVA\_HOME de la siguiente forma:

```
JAVA_HOME=C:\Java y PATH=%JAVA_HOME%\bin
```

IMPORTANTE: cuando se modifica el valor de una variable de entorno, se tiene que cerrar y volver a abrir la CLI, para que los cambios queden activados.



## Configurar PATH

Los pasos a seguir para configurar el PATH dependen del sistema operativo que tengamos:

### Windows XP:

1. Start -> Control Panel -> System -> Advanced
2. Click on Environment Variables, under System Variables, find **PATH**, and click on it.
3. In the Edit windows, modify **PATH** by adding the location of the class to the value for **PATH**. If you do not have the item **PATH**, you may select to add a new variable and add **PATH** as the name and the location of the class as the value.
4. Close the window.
5. Reopen Command prompt window, and run your java code.

### Windows Vista:

1. Right click "My Computer" icon
2. Choose "Properties" from context menu
3. Click "Advanced" tab ("Advanced system settings" link in Vista)
4. In the Edit windows, modify **PATH** by adding the location of the class to the value for **PATH**. If you do not have the item **PATH**, you may select to add a new variable and add **PATH** as the name and the location of the class as the value.
5. Reopen Command prompt window, and run your java code.

### GNU/Linux & Unix, bash Shell:

1. Edit the startup file (~/.bashrc)
2. Modify **PATH** variable:  
**PATH=/usr/local/jdk1.6.0/bin**
3. export **PATH**
4. Save and close the file
5. Open new Terminal window
6. Verify the **PATH** is set properly  
**% java -version**

### GNU/Linux & Unix, C Shell (csh):

1. Edit startup file (~/.cshrc)
2. Set Path  
**set path=(/usr/local/jdk1.6.0/bin )**
3. Save and Close the file
4. Open new Terminal window
5. Verify the **PATH** is set properly  
**% java -version**

En sistemas Microsoft, el separador entre valores de variables de entorno es ';' . En cambio para entornos GNU/Linux, Unix y Mac, se utiliza ':' .

## Configurar CLASSPATH

Los pasos a seguir son los mismos que los expuestos en el punto anterior para el PATH, pero esta vez



actuando sobre la variable CLASSPATH, y añadiendo siempre el valor '.'. Además, también añadiremos la ruta donde esté ubicada la librería tads.jar facilitada para realizar el EP1 sobre CLI.

## 2.3. Ejecución del EP1 sobre CLI con CLASSPATH

Una vez instalado el JDK, es tan sencillo como indicar en la variable de entorno PATH donde están los ejecutables de Java (carpeta bin), indicar en la variable de entorno CLASSPATH donde está la librería tads.jar y añadir el punto al final (ejemplo con Windows: CLASSPATH=C:\UOC\lib\_cast\tads.jar.), y realizar la compilación y ejecución del EP1.

Ejemplo de compilación y ejecución en GNU/Linux:

- Se muestra el valor de la variable CLASSPATH:  

```
/uoc/ep1_cast$ echo $CLASSPATH
```

```
/uoc/lib_cast/tads.jar.
```
- Se compila el código fuente, obteniendo el correspondiente binario EjercicioPractico1.class:  

```
/uoc/ep1_cast$ javac EjercicioPractico1.java
```
- Se ejecuta el programa:  

```
/uoc/ep1_cast$ java EjercicioPractico1
```

```
Crea una cola acotada para 9 elementos y encola los digitos del '0' al '8'
```

```
Crea una pila acotada para 9 elementos y apila los digitos del '0' al '8'
```

```
Desencola todos los elementos: 0 1 2 3 4 5 6 7 8
```

```
Desapila todos los elementos: 8 7 6 5 4 3 2 1 0
```

La salida mostrada puede que no coincida exactamente con el EP1 facilitado en el aula de teoría: es solo a modo de ejemplo.

## 2.4. Ejecución del EP1 sobre CLI sin CLASSPATH

A continuación se expone una alternativa al punto anterior 2.3: realizar la compilación y ejecución del código fuente sin tener definida la variable de entorno CLASSPATH. Así, los puntos 2.3 y 2.4 son disjuntos: o bien podéis compilar y ejecutar como se indica en el punto anterior, o bien como se expone a continuación.

Ejemplo de compilación y ejecución con Windows:

- Se comprueba que no está definida la variable CLASSPATH:



```
C:\uoc\ep1_cast> set CLASSPATH
```

Variable de entorno CLASSPATH no definida

- Se compila el código fuente con la opción `-classpath`, indicando donde está la librería de TADs y añadiendo el `.` :

```
C:\uoc\ep1_cast>javac -classpath C:\uoc\ep1_cast\tads.jar;. EjercicioPractico1.java
```

- Se ejecuta el programa:

```
C:\uoc\ep1_cast> java -classpath C:\uoc\ep1_cast\tads.jar;. EjercicioPractico1
```

El resultado obtenido es idéntico al del punto 2.3.





## 3. Entorno IDE: Eclipse

### 3.1. Descarga e instalación

El primer paso es descargar el IDE de la web del proyecto Eclipse: <http://www.eclipse.org/downloads/> .

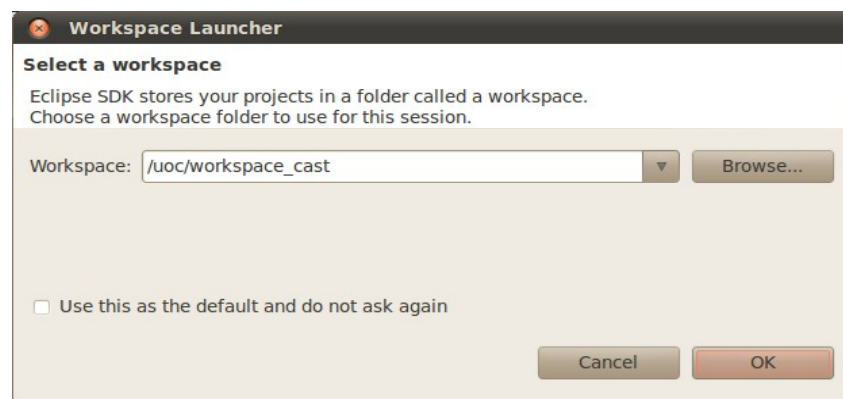
En caso de disponer del software en soporte físico enviado por la UOC, también se puede utilizar; aun así, se recomienda descargar Eclipse desde su web para disponer de una versión actualizada.

Para la realización de esta documentación se ha usado la versión versión Eclipse IDE for Java Developers; otras versiones como Eclipse Classic también son válidas para la asignatura.

Uno de los puntos positivos de Eclipse es que su instalación solo consiste en descomprimir el fichero descargado, y ejecutar el binario eclipse. Esto significa que podemos tener múltiples IDE Eclipse instalados.

### 3.2. Inicio

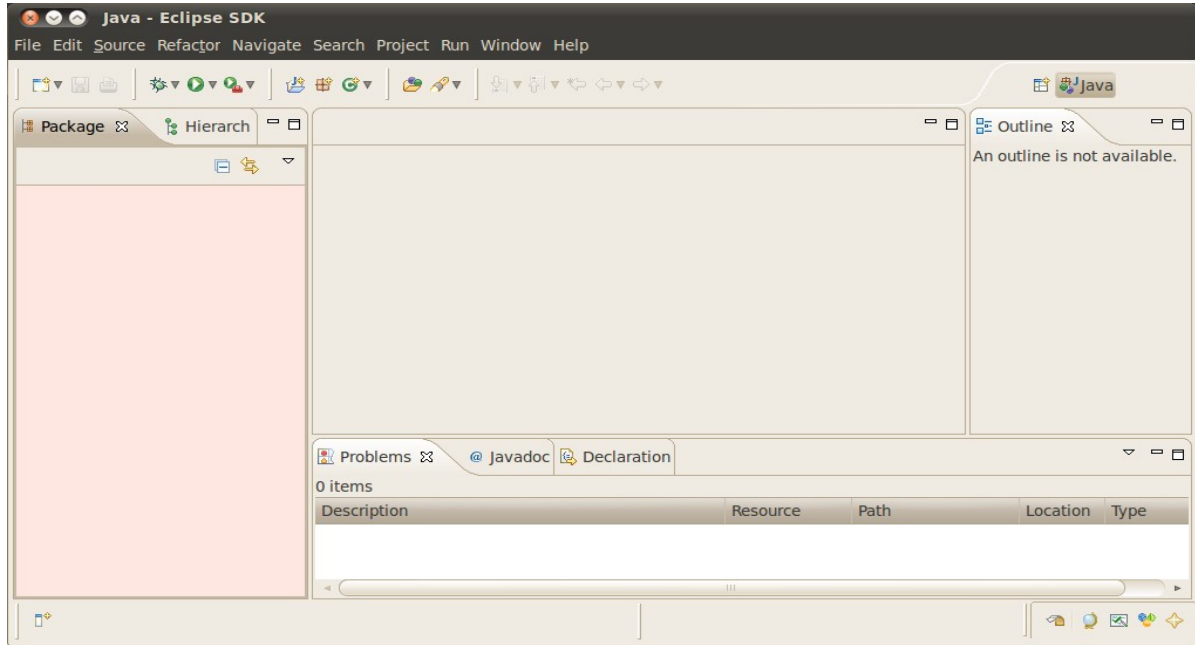
Cuando arrancamos Eclipse, se muestra una pantalla inicial donde nos pide qué workspace utilizar. Un workspace es la carpeta en la cual se guardará uno o más proyectos; se pueden tener múltiples workspace en un mismo ordenador.



Si se desea, se puede marcar la opción Use this as the default and do not ask again, para que no nos pregunte más qué workspace utilizar.



La primera vez se mostrará una pantalla de bienvenida, donde se tiene que seleccionar Go to the workbench. A continuación se visualizará la pantalla principal de Eclipse:



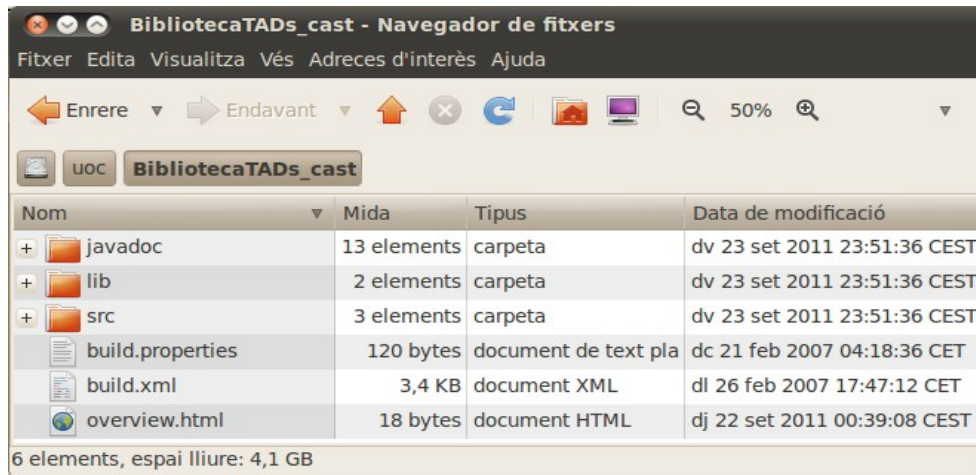
### 3.3. Instalación de la biblioteca de TADs

Eclipse no utiliza la variable CLASSPATH de nuestro SO, por lo que es necesario hacer visible la biblioteca de TADs.

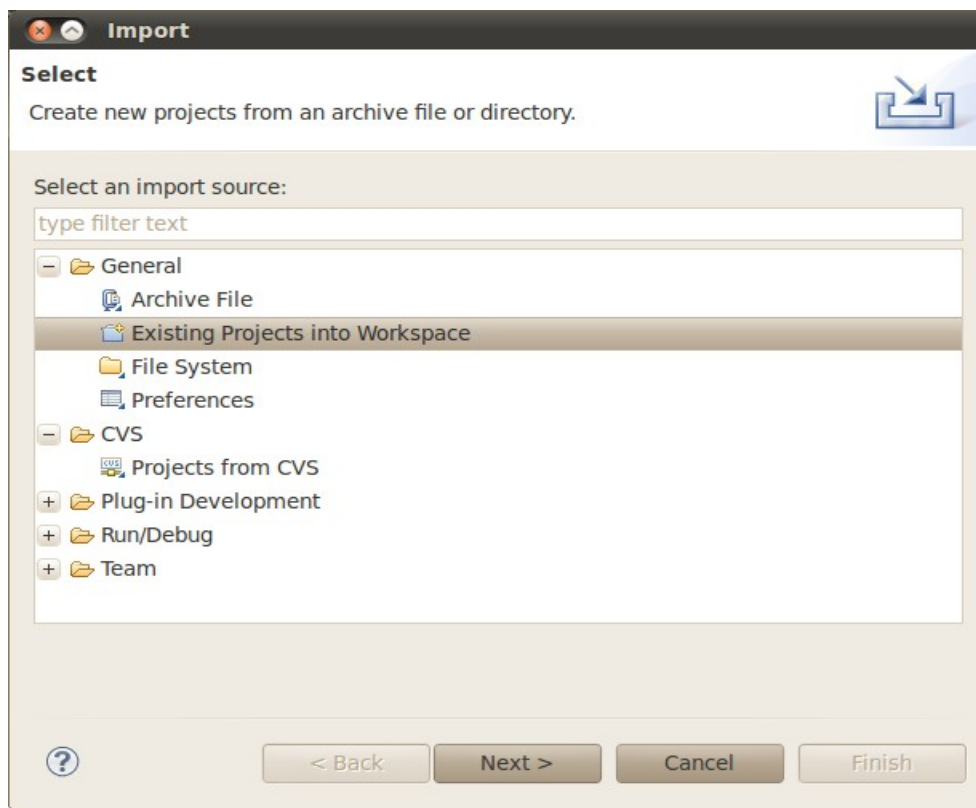
A continuación se propone hacerlo mediante un proyecto propio, BibliotecaTADs\_cast, que solo contendrá la biblioteca de TADs, de manera que cuando cualquier otro proyecto requiera de la biblioteca solo tendrá que referenciar BibliotecaTADs\_cast para tener visibilidad de ella.

En primer lugar, se descarga la biblioteca de TADs BibliotecaTADs\_cast.zip, y se descomprime donde queramos. El contenido de la biblioteca es el siguiente:

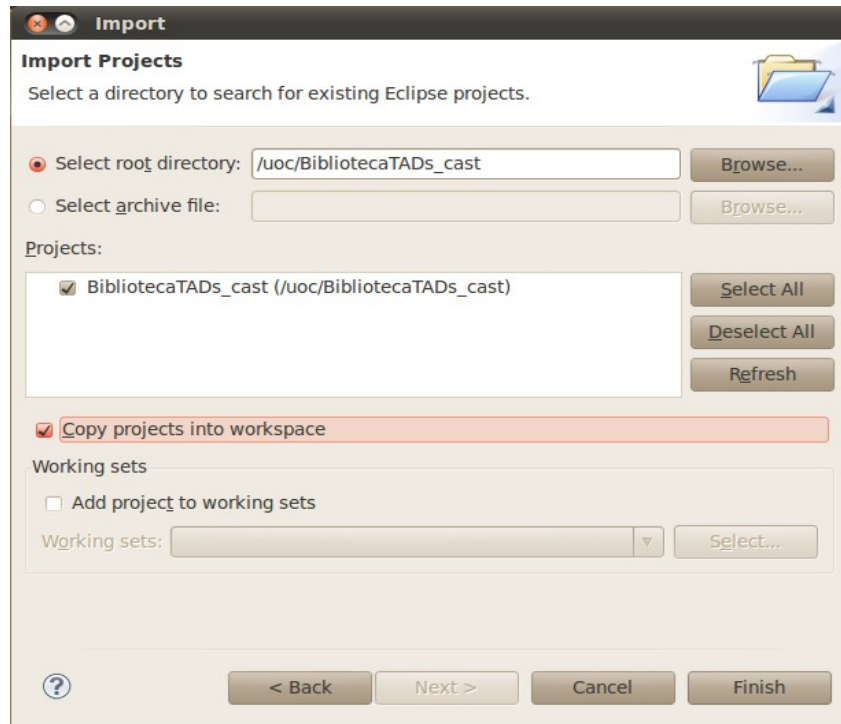




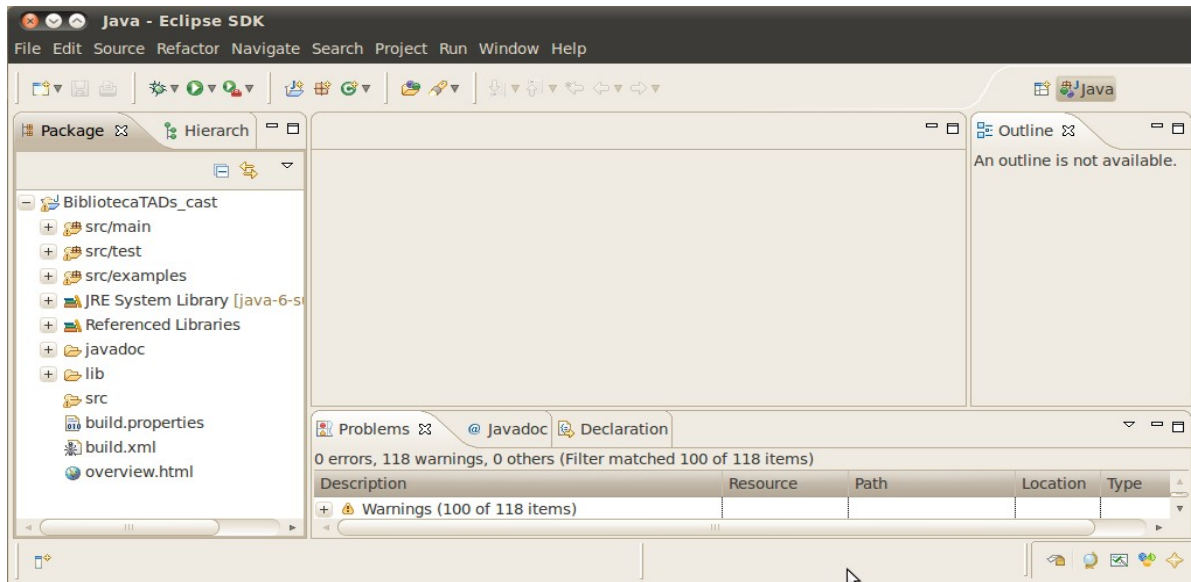
Para crear el proyecto de la biblioteca TADs, dentro de Eclipse se selecciona al menú superior: File → Import... → General: Existing Projects into Workspace:



A continuación se indica donde está ubicada la biblioteca de TADs descomprimida anteriormente , y se indica que copie su contenido dentro de nuestro workspace:



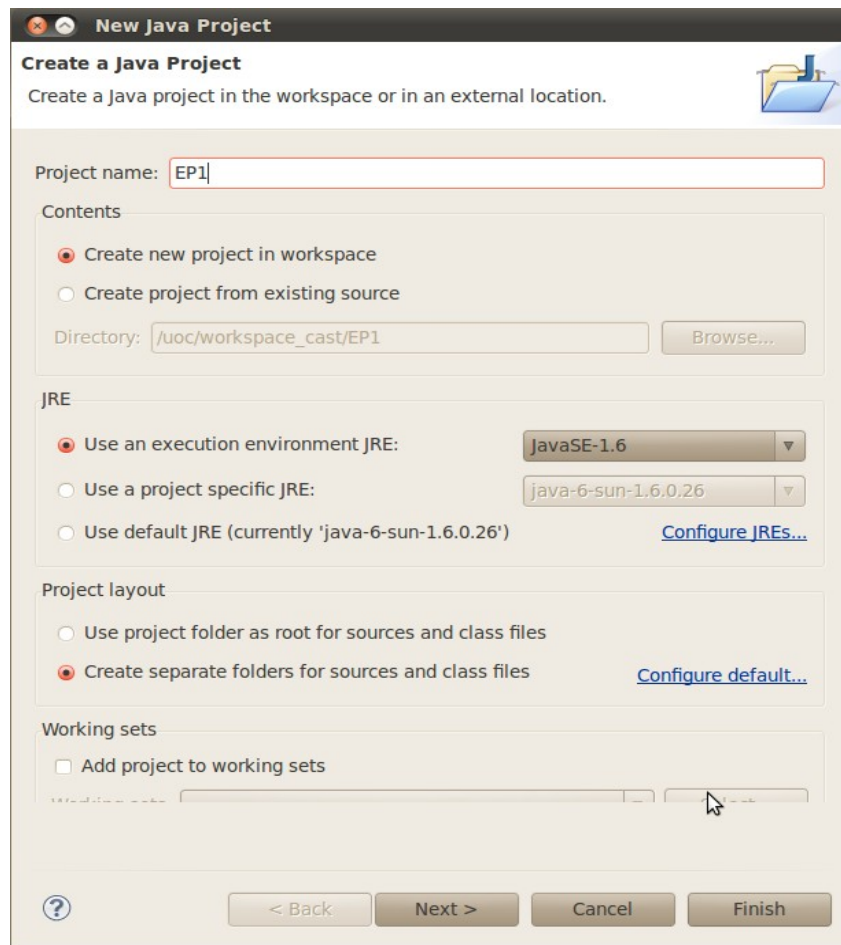
En la pantalla principal, a partir de este momento se muestra el proyecto BibliotecaTADs\_cast:



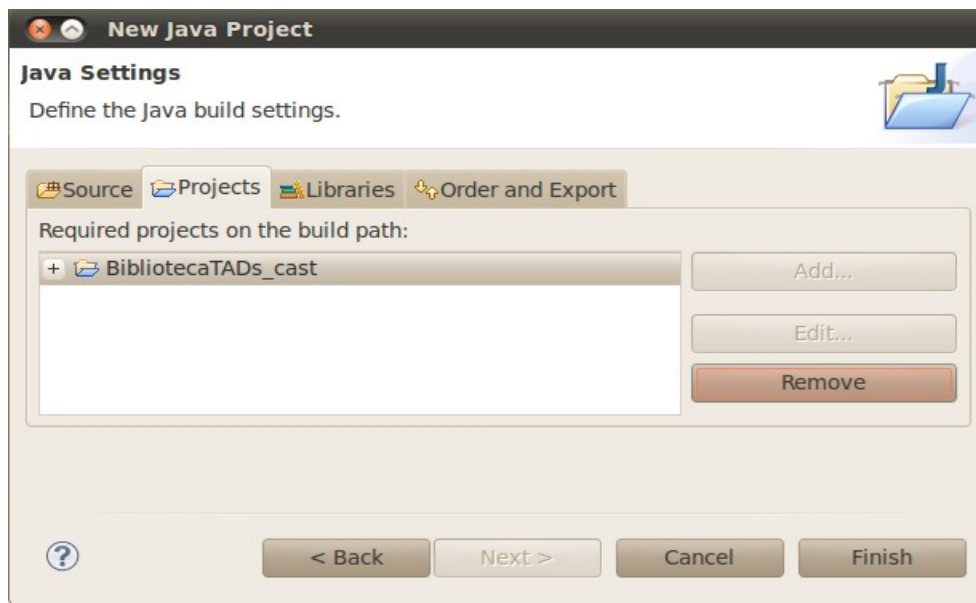
### 3.4. Ejecución del EP1 sobre Eclipse

A continuación se creará un nuevo proyecto llamado EP1, que servirá para validar que la librería de TADs está correctamente instalada.

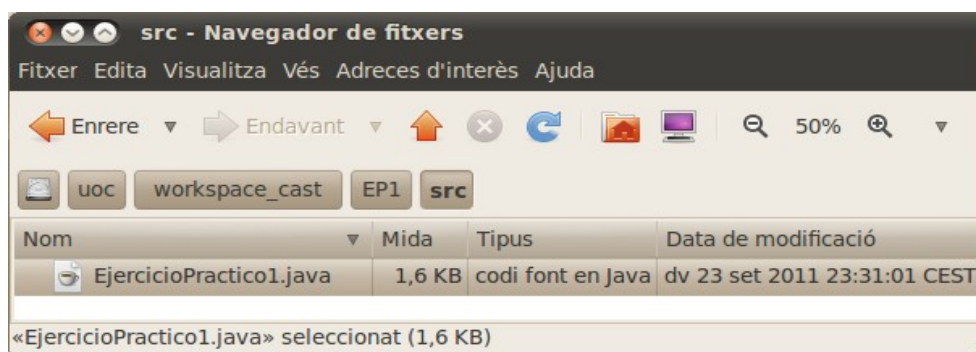
A través del menú superior de Eclipse, se crea un nuevo proyecto a través de la opción: File → New → Java Project:



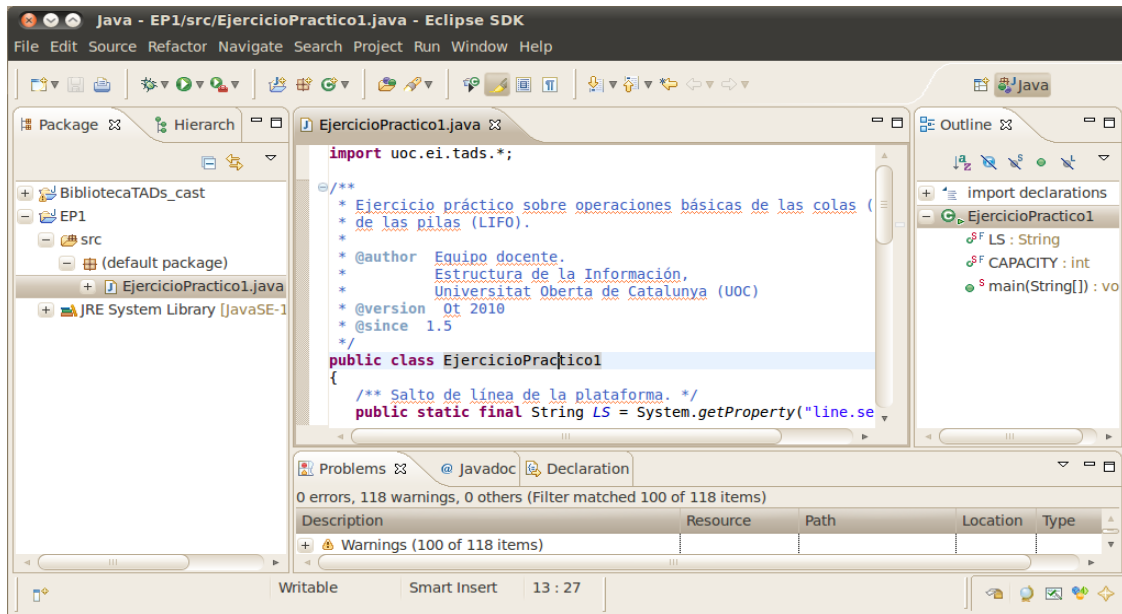
A continuación se selecciona Next >, y en la pestaña Projects se añade el proyecto BibliotecaTADs\_cast, con el objetivo de asociar el proyecto EP1 con BibliotecaTADs\_cast:



En este momento, el proyecto EP1 está vacío, por lo que copiaremos en él el fichero EjercicioPractico1.java facilitado junto con el enunciado del EP1, dentro del proyecto: ...workspace/EP1/src :



Al pulsar F5 en la pantalla principal de Eclipse, se actualiza el contenido del proyecto EP1. Si se hace doble click sobre EjercicioPractico1.java del proyecto EP1, se puede ver su contenido en la pantalla principal:



Antes de ejecutar, con el botón derecho del ratón sobre el proyecto EP1 → Properties → Java Compiler → validar que el valor compiler compliance level sea superior o igual a 1.5.

Para ejecutar la clase, podemos pulsar el botón play; también se puede hacer a través del menú, opción Run → Run. La salida de la ejecución se mostrará en la Console inferior (la que se muestra en la figura inferior es solo un ejemplo):



