

# Tutorial de creación de proyectos Java con el API de LeJOS para el desarrollo de robots autónomos

Juan Antonio Breña Moral

## índice

Introducción.....	3
Proceso de creación de un proyecto.....	3
Creacion de un nuevo proyecto en Java.....	4
Añadir las librerías de programación de robots empleando el API de LeJOS.....	8
Creación de la primera clase empleando el API de LeJOS.....	11
Codigo del ejemplo Hello World.....	14

## Índice de las ilustraciones

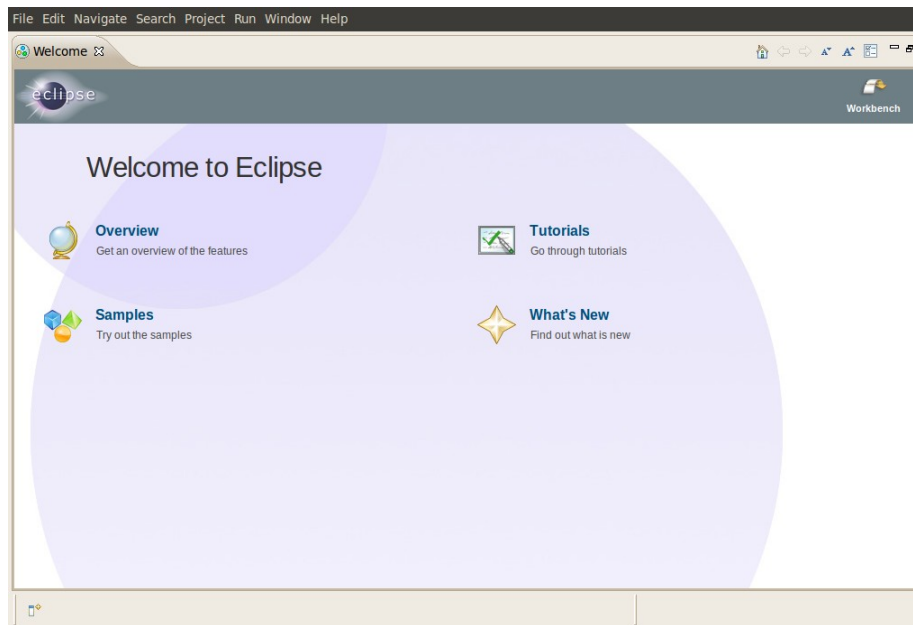
Illustration 1: Imagen inicial de Eclipse.....	3
Illustration 2: Entorno de trabajo en Eclipse.....	4
Illustration 3: Asistente de creacion de proyecto en Java con Eclipse.....	5
Illustration 4: Paso segundo en el asistente.....	6
Illustration 5: Paso tercero en el asistente.....	7
Illustration 6: Detalle del visor de proyectos con un nuevo proyecto vacio.....	8
Illustration 7: Pantalla inicial en la configuracion del Build Path.....	9
Illustration 8: Añade la libreria externa de LeJOS.....	9
Illustration 9: Si tienes instalado LeJOS, classes.jar se situa en /usr/local/lejos/lib/.....	10
Illustration 10: Fijate en el visor que aparece la referencia a Classes.jar.....	11
Illustration 11: El mismo asistente te sirve tambien para crear Clases en Java.....	12
Illustration 12: Paso segundo en la creacion de una clase.....	13
Illustration 13: Ejemplo HelloWorld.....	14

## Introducción

El presente documento, tiene como objeto enseñar como crear un proyecto en eclipse de Java usando el API de programación del proyecto Open Source LeJOS, Java for Lego Mindstorms

## Proceso de creación de un proyecto

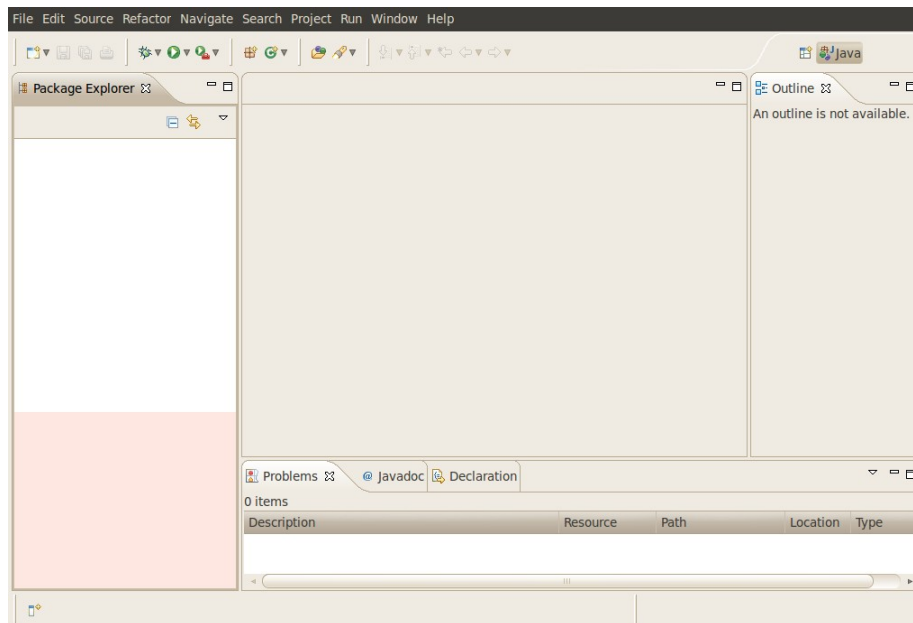
Cuando arrancas eclipse por primera vez, te encontrarás con la siguiente ventana de inicio:



*Illustration 1: Imagen inicial de Eclipse*

Cierra dicha ventana y veras el entorno de trabajo normal.

El entorno de trabajo, se caracteriza por el visor de proyectos y el visor de código fuente. Adicionalmente una ventana muy importante es la consola y la ventana de errores donde te mostrara información útil para depurar tus programas.

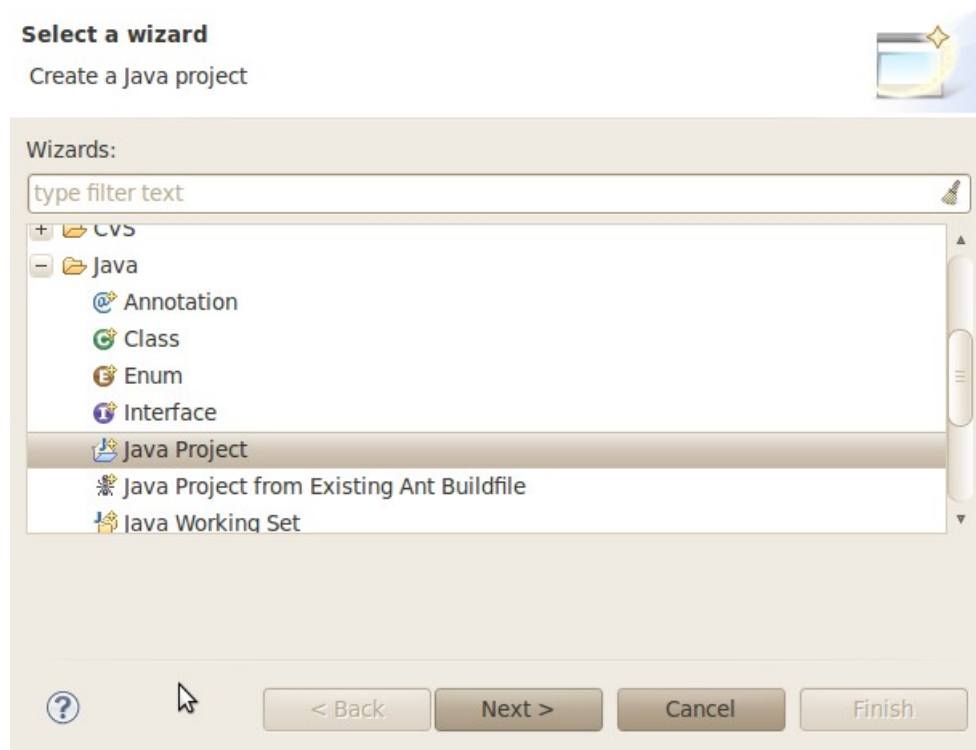


*Illustration 2: Entorno de trabajo en Eclipse*

### ***Creacion de un nuevo proyecto en Java***

Para crear un nuevo proyecto, pulsa Ctrl+n o buscas en la opcion File, New Project entonces tendrás que seguir los pasos del asistente.

Lo primero que hay que hacer es seleccionar la opcion de creacion de nuevo proyecto en Java:

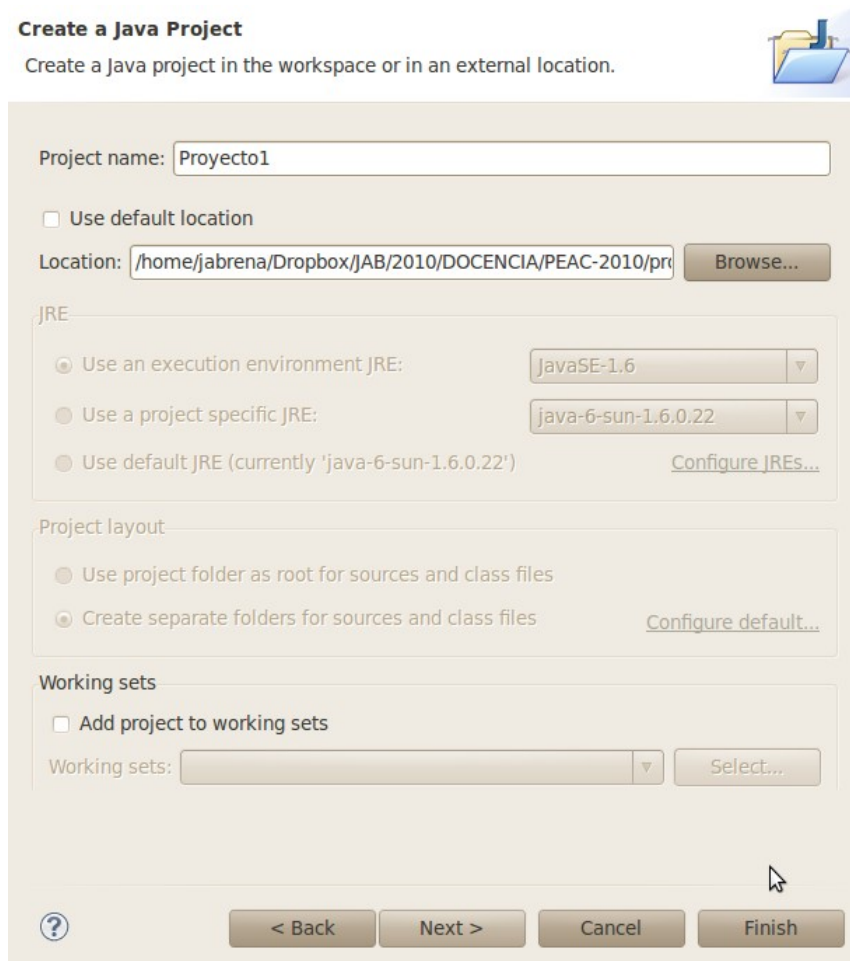


*Illustration 3: Asistente de creacion de proyecto en Java con Eclipse*

Un proyecto en Java tiene los siguientes parámetros:

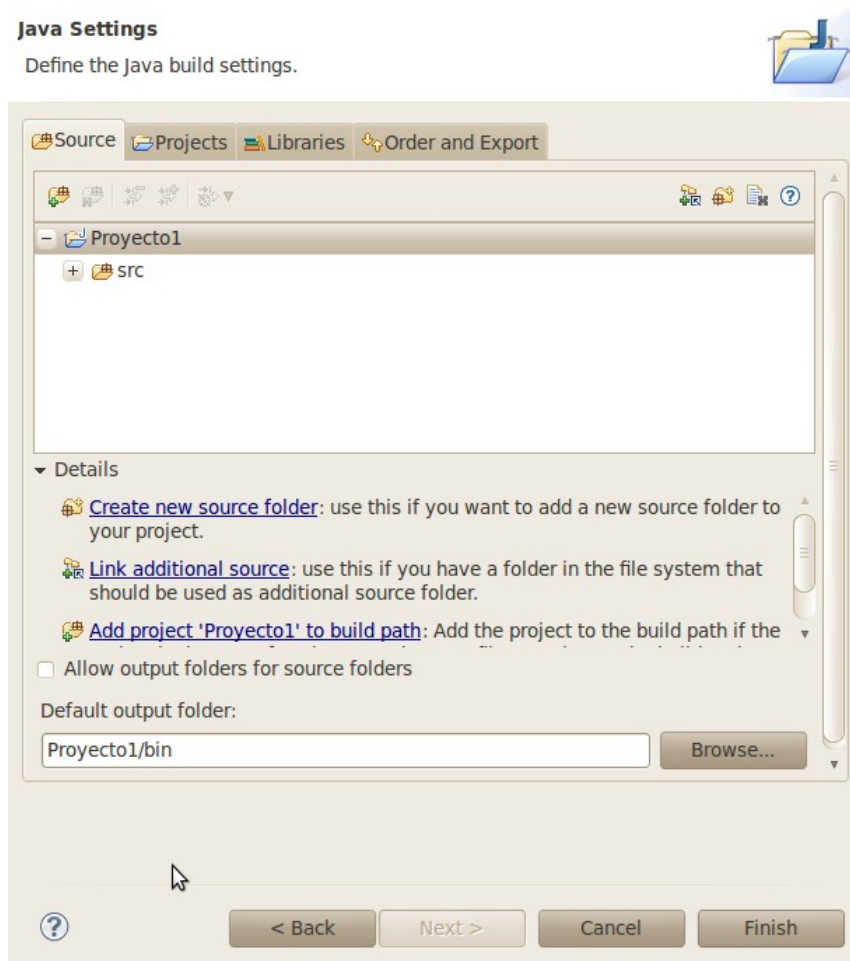
1. Nombre del proyecto
2. Path donde se ubica el código

## Documentación para cursos de introducción a la robótica



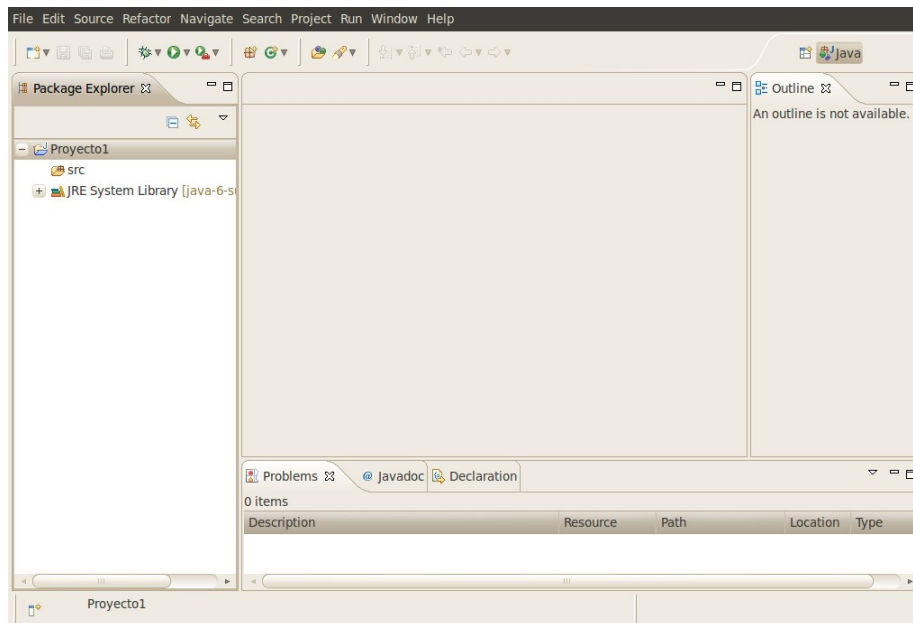
*Illustration 4: Paso segundo en el asistente*

Cuando crees el proyecto, asegurate que se crea una carpeta src para el código fuente y la carpeta bin para los ficheros bytecode (.class), esto lo veras en la siguiente ventana del asistente:



*Illustration 5: Paso tercero en el asistente*

A continuación pulsa Finalizar para terminar con el proceso. En este momento se creará la estructura de un proyecto vacío en el entorno de desarrollo. Si todo ha ido bien, deberías ver la estructura del proyecto de la siguiente forma:



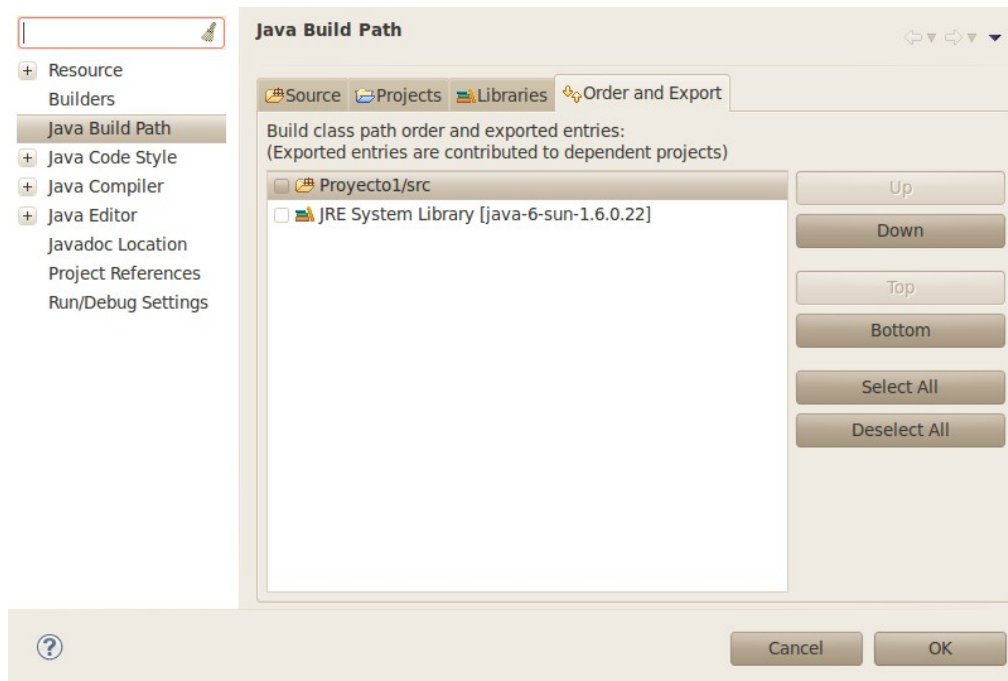
*Illustration 6: Detalle del visor de proyectos con un nuevo proyecto vacío*

Si tienes algún problema, revisa la carpeta del proyecto y repite los pasos con más cuidado.

## **Añadir las librerías de programación de robots empleando el API de LeJOS**

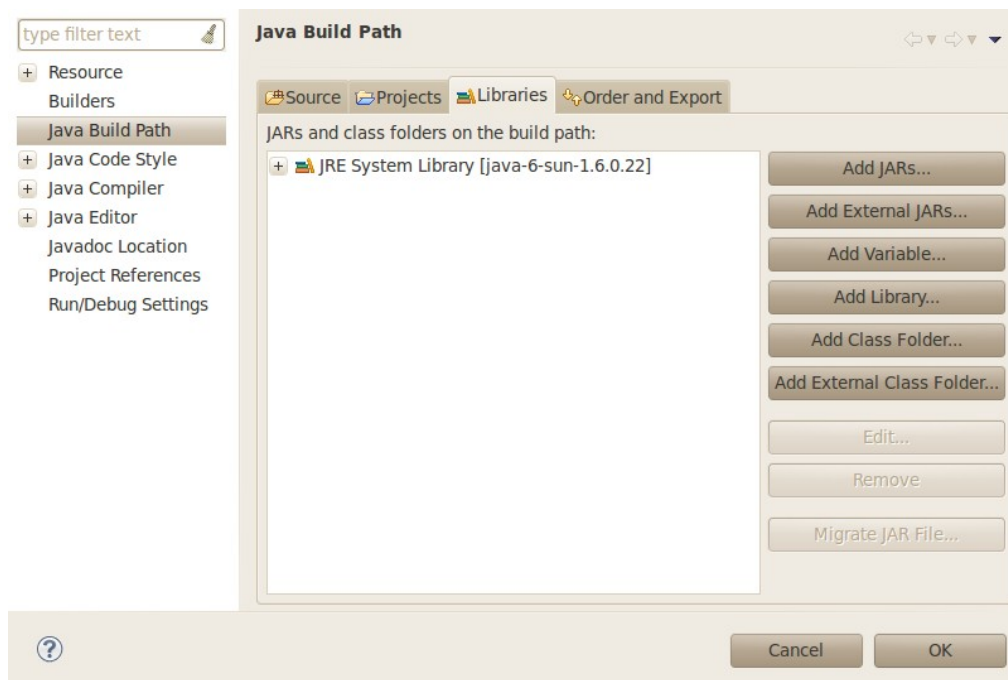
Una vez has creado el proyecto en Java, deberás indicarle que vas a usar las librerías del proyecto LeJOS, Java for Lego Mindstorms. Para conseguir este propósito, deberás añadir la librería empleando la opción de eclipse “Build Path”

## Documentación para cursos de introducción a la robótica



*Illustration 7: Pantalla inicial en la configuración del Build Path*

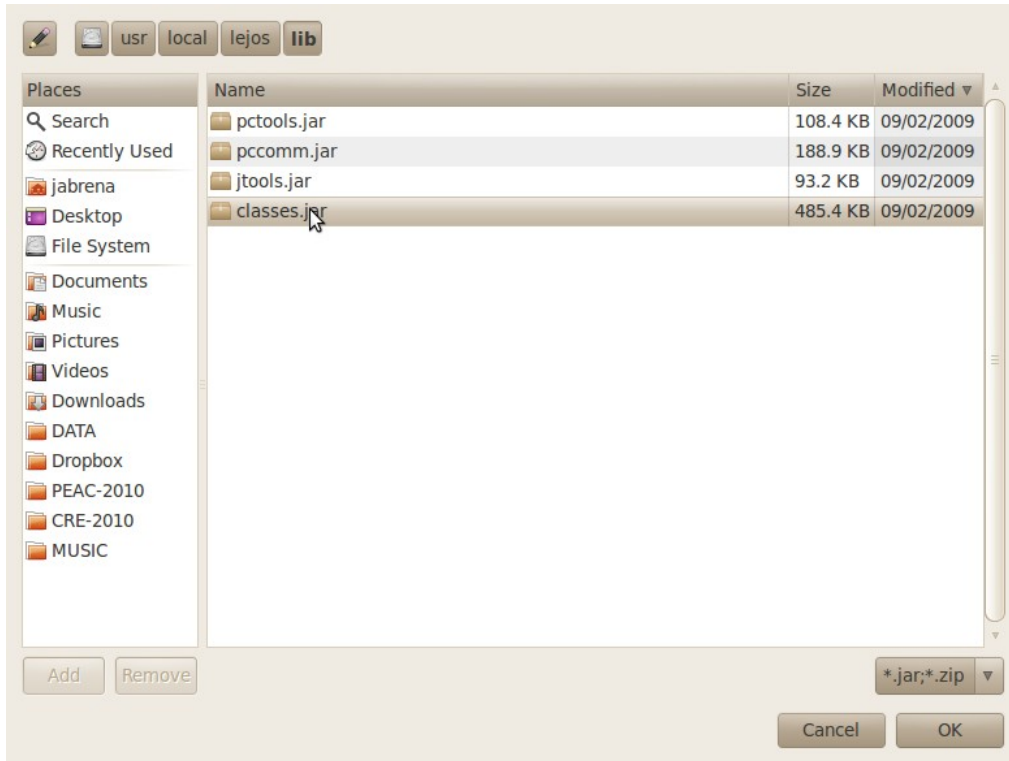
Una vez estas sobre dicha ventana, pulsa la pestaña de Libraries para añadir la librería (.jar) que se encuentra en el path: /usr/local/lejos/lib



*Illustration 8: Añade la librería externa de LeJOS*

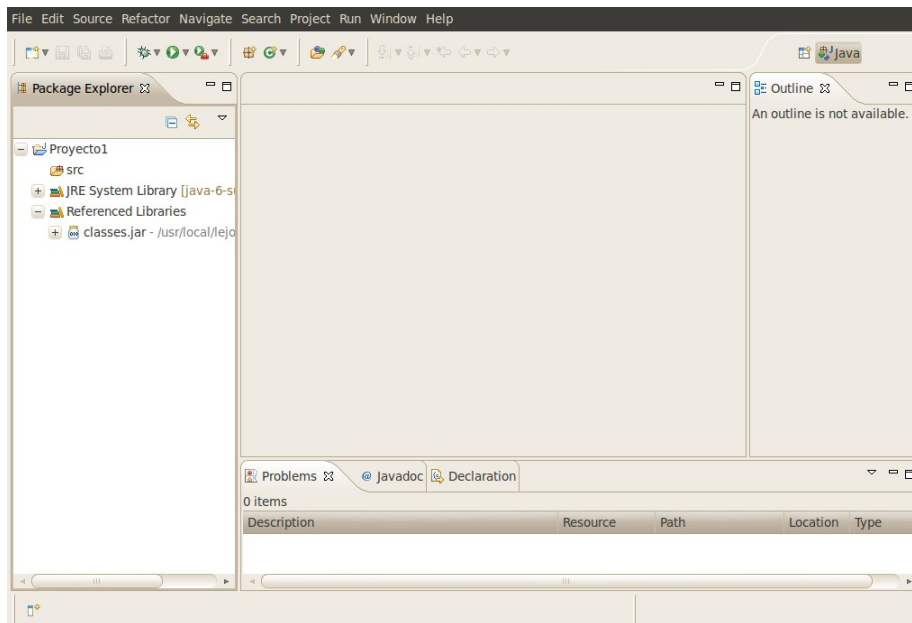
## Documentación para cursos de introducción a la robótica

En dicho path, estan disponibles todas las librerias disponibles, pero para trabajar con el brick, la unica que necesitas es Classes.jar



*Illustration 9: Si tienes instalado LeJOS, classes.jar se sitúa en /usr/local/lejos/lib/*

Al finalizar el proceso, deberías ver tu proyecto en eclipse con la referencia a la librería:

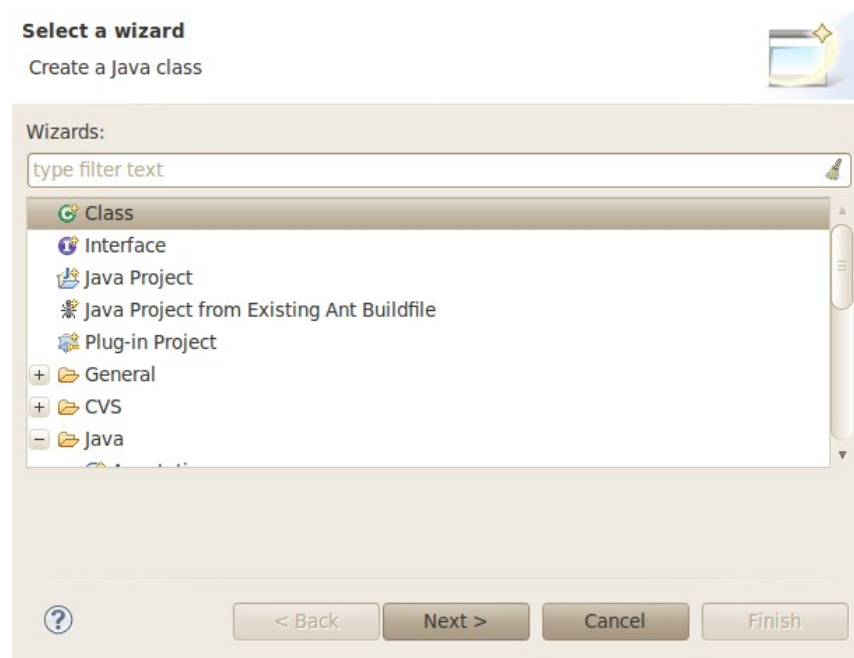


*Illustration 10: Fijate en el visor que aparece la referencia a Classes.jar*

## **Creación de la primera clase empleando el API de LeJOS**

Si el proyecto de Java dispone de la referencia a las librerías de LeJOS, el siguiente paso consiste en la creación de una Clase en Java. En Java, la unidad mínima de programación es la Clase. Para añadir al proyecto una clase, lo que se hace es abrir el asistente a través de `Ctrl + n` o a través del menú `file/new / Class`

## Documentación para cursos de introducción a la robótica



*Illustration 11: El mismo asistente te sirve tambien para crear Clases en Java*

Una vez seleccionado la opcion Class, se escribira el nombre de la Clase empleando notación Camel:

[http://en.wikipedia.org/wiki/Camel\\_notation](http://en.wikipedia.org/wiki/Camel_notation)

<http://www.oracle.com/technetwork/java/codeconventions-135099.html#367>

## Documentación para cursos de introducción a la robótica

**Java Class**

⚠ The use of the default package is discouraged.

Source folder:

Package:

Enclosing type:

Name:

Modifiers:  public  default  private  protected  
 abstract  final  static

Superclass:

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)

Constructors from superclass

Inherited abstract methods

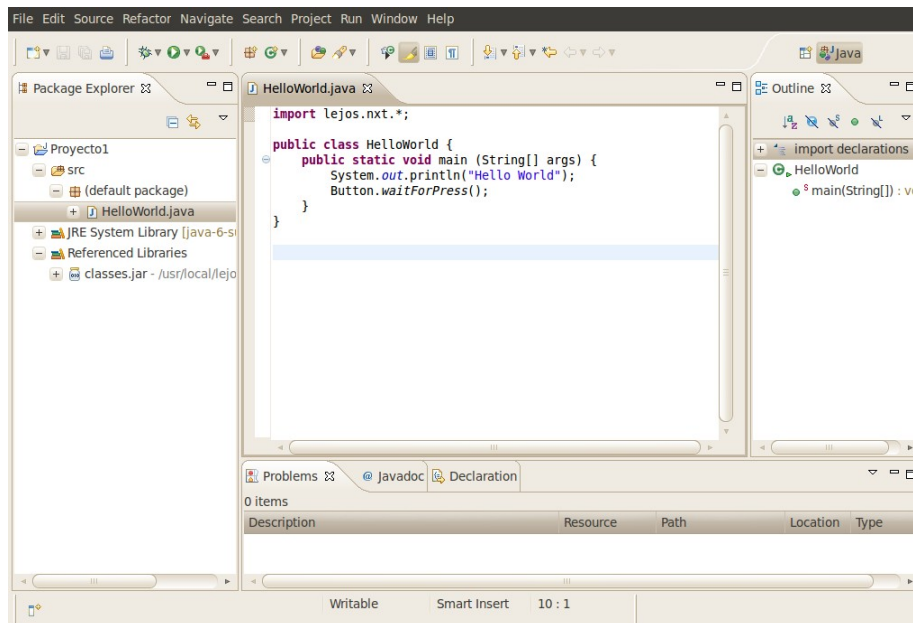
Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

*Illustration 12: Paso segundo en la creación de una clase*

Una vez finalizado el proceso, editarás el fichero y el resultado es el siguiente:

## Documentación para cursos de introducción a la robótica



*Illustration 13: Ejemplo HelloWorld*

### Codigo del ejemplo Hello World

El codigo del ejemplo es el siguiente:

```
import lejos.nxt.*;

public class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello World");
        Button.waitForPress();
    }
}
```