



OBJETIVO

El objetivo de este documento es mostrar cómo se configura el proyecto relativo a la fase T1 del trabajo para poder ejecutar un conjunto de pruebas JUnit que permiten testear el diseño de las interfaces y los enumerados que hay que trabajar en el aula.

CONFIGURACIÓN DEL PROYECTO

Para que se puedan ejecutar las pruebas en el proyecto hay que añadir al mismo cuatro bibliotecas: la relacionada con JUnit, cuyo archivo .jar es proporcionado en el propio entorno Eclipse; la biblioteca *guava-16.0.1.jar*; la biblioteca *helpers-1.0.8.jar* y la biblioteca *tests1516-E0-1.0.0.jar* que contiene las pruebas. Estas tres últimas bibliotecas las puede descargar de la Enseñanza Virtual.

Antes de configurar el proyecto, es conveniente crear una carpeta llamada *lib* y copiar en ella los archivos *guava-16.0.1.jar*, *helpers-1.0.8.jar* y *tests1516-E0-1.0.0.jar*.

A continuación se muestran los pasos para añadir estas bibliotecas:

- Para añadir la biblioteca JUnit.
 1. Marcando el proyecto pulsamos el botón derecho del ratón y escogemos la opción *Build Path* > *Configure Build Path* (Figura 1).

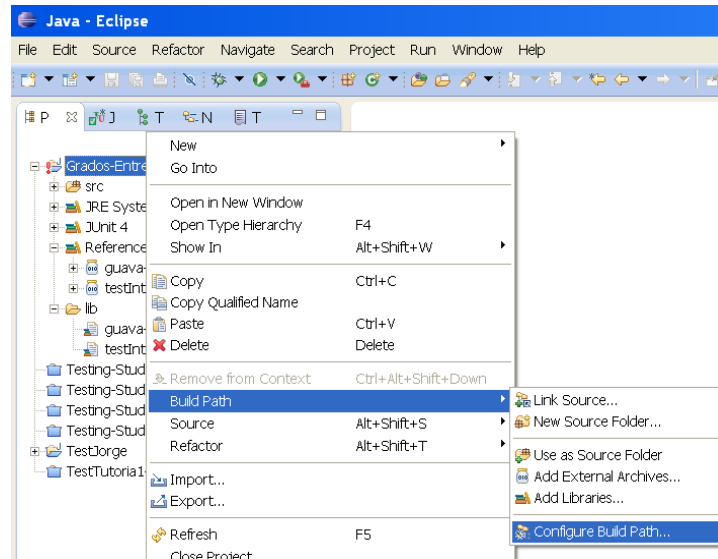


Figura 1. Seleccionar Configure Build Path

2. En el asistente, seleccionamos la pestaña *Libraries* y pulsamos el botón *Add Library* (Figura 2)

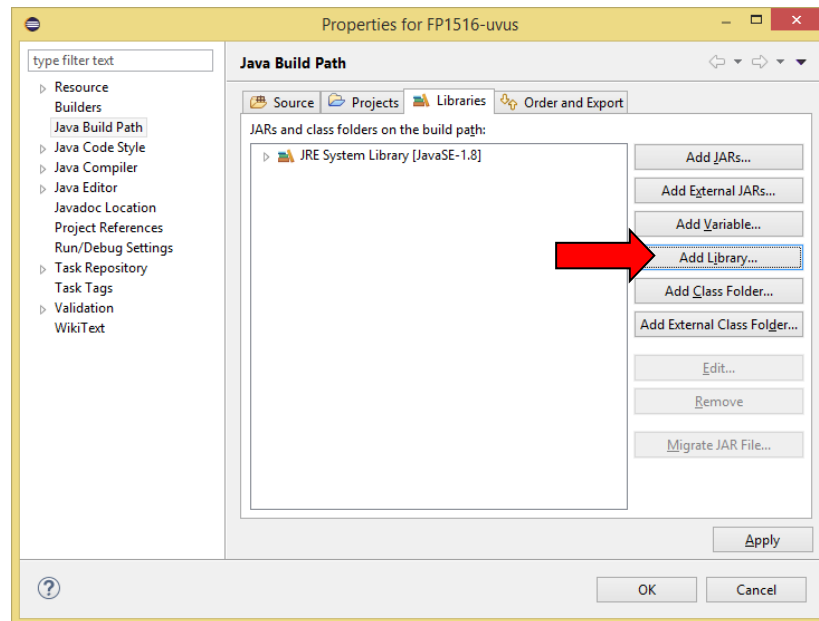


Figura 2. Botón Add Library del asistente

3. Seleccionamos *JUnit* (Figura 3(a)) y pulsamos el botón *Next*. Nos aseguramos de que la versión de la biblioteca JUnit es la 4 (Figura 3(b)) y pulsamos *Finish*.

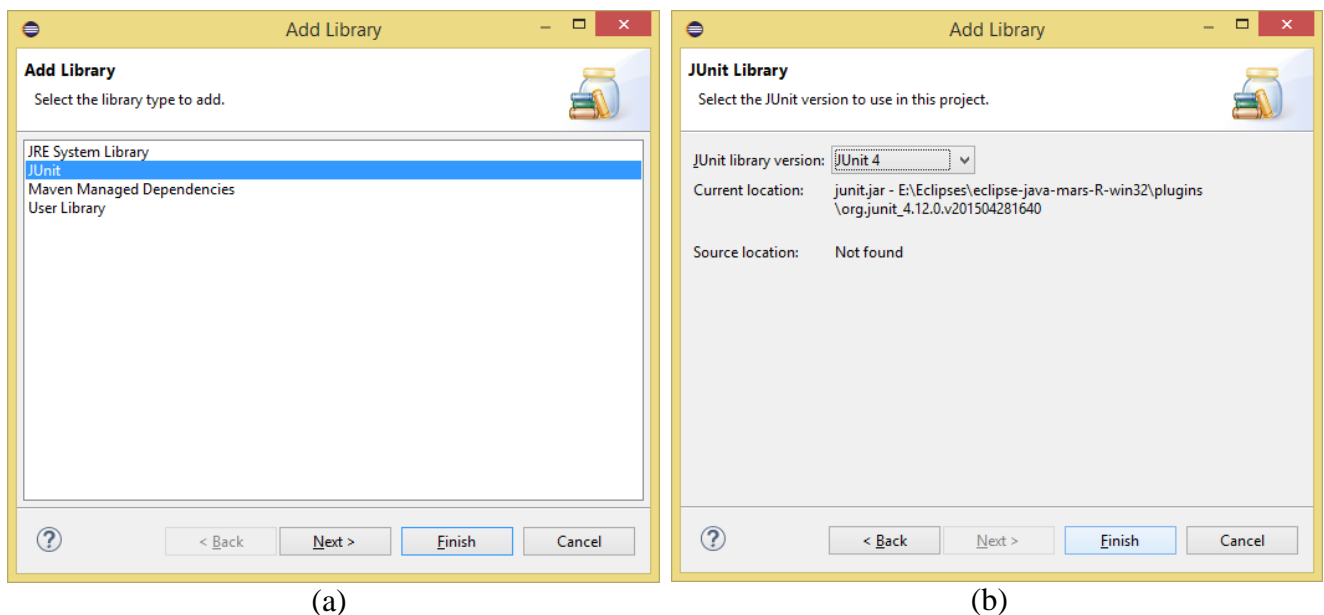


Figura 3. (a) Selección de biblioteca JUnit. (b) Selección de versión JUnit 4

4. En el asistente *Java Build Path* aparece la biblioteca *JUnit 4* (Figura 4). Tras pulsar *OK* la biblioteca queda añadida al proyecto (Figura 5).

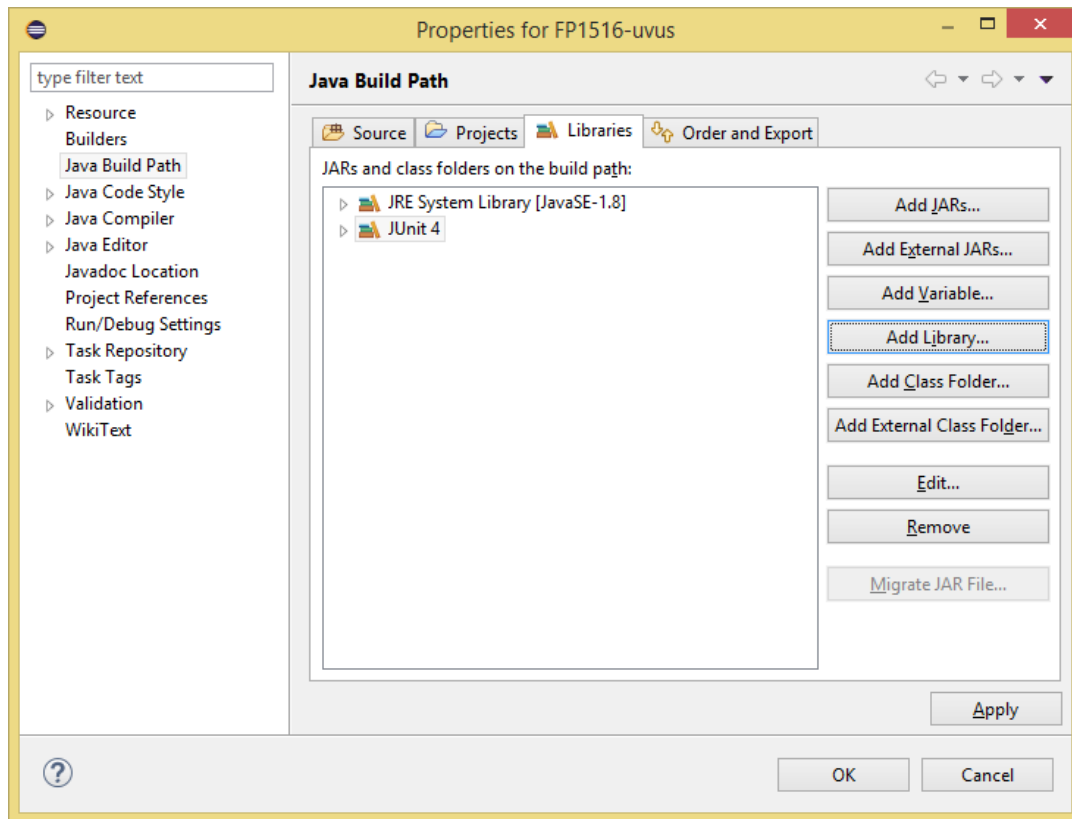


Figura 4. Biblioteca JUnit 4 reflejada en el asistente

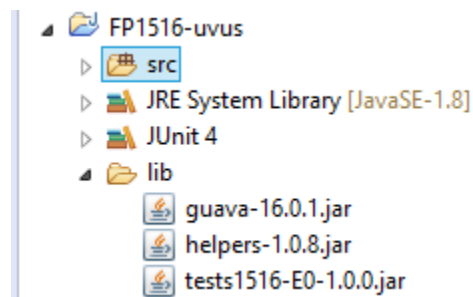


Figura 5. Biblioteca JUnit reflejada en el proyecto

- Para añadir las bibliotecas *guava-16.0.1.jar*, *helpers-1.0.8.jar* y *tests1516-E0-1.0.0.jar* se procederá como se especifica en el apartado IMPORTACIÓN DE BIBLIOTECAS .JAR A PROYECTO ECLIPSE publicado en el *Boletín 0: Entorno de trabajo*. El proyecto debe quedar configurado como se muestra en la Figura 6.

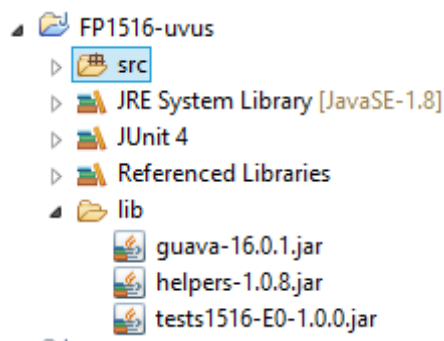


Figura 6. Configuración del proyecto

EJECUCIÓN DE PRUEBAS

Una vez configurado el proyecto para ejecutar el test, en el explorador de proyectos, abra los ítems *Referenced Libraries* y *tests1516-E0-1.0.0.jar*. Busque el paquete *test.main*, y seleccione *TestsE0.class* (Figura 7).

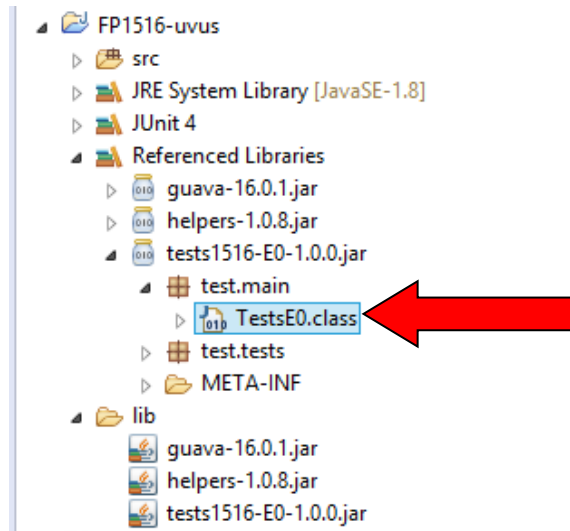


Figura 7. Localización del archivo TestsE0.class en el proyecto

Pulse el botón derecho del ratón, y en el menú desplegable escoja *Run As > JUnit Test* (Figura 8).

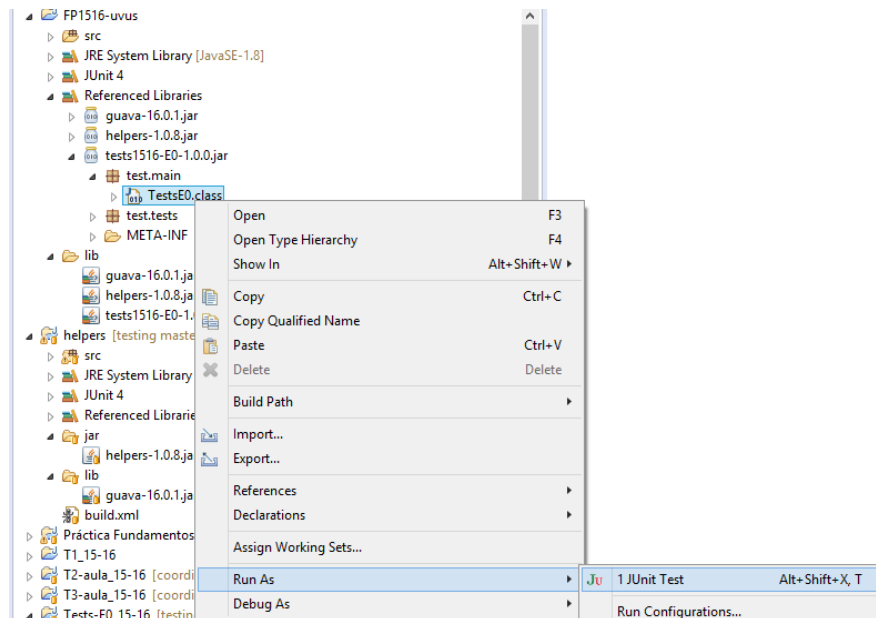


Figura 8. Ejecución de TestsE0.class como test de JUnit a través del menú desplegable

Aparecerá una nueva vista, y si sus interfaces y enumerados son correctos, todos los elementos deben estar marcados con un icono verde, y no debe aparecer ningún error ni ningún fallo. Fíjese que en el cuadro de texto *Errors* de la parte superior de la vista aparece un 0, y que en el cuadro *Failures* aparece también un 0. En la Figura 9 se muestra un ejemplo de ejecución del test que no tiene errores ni fallos.

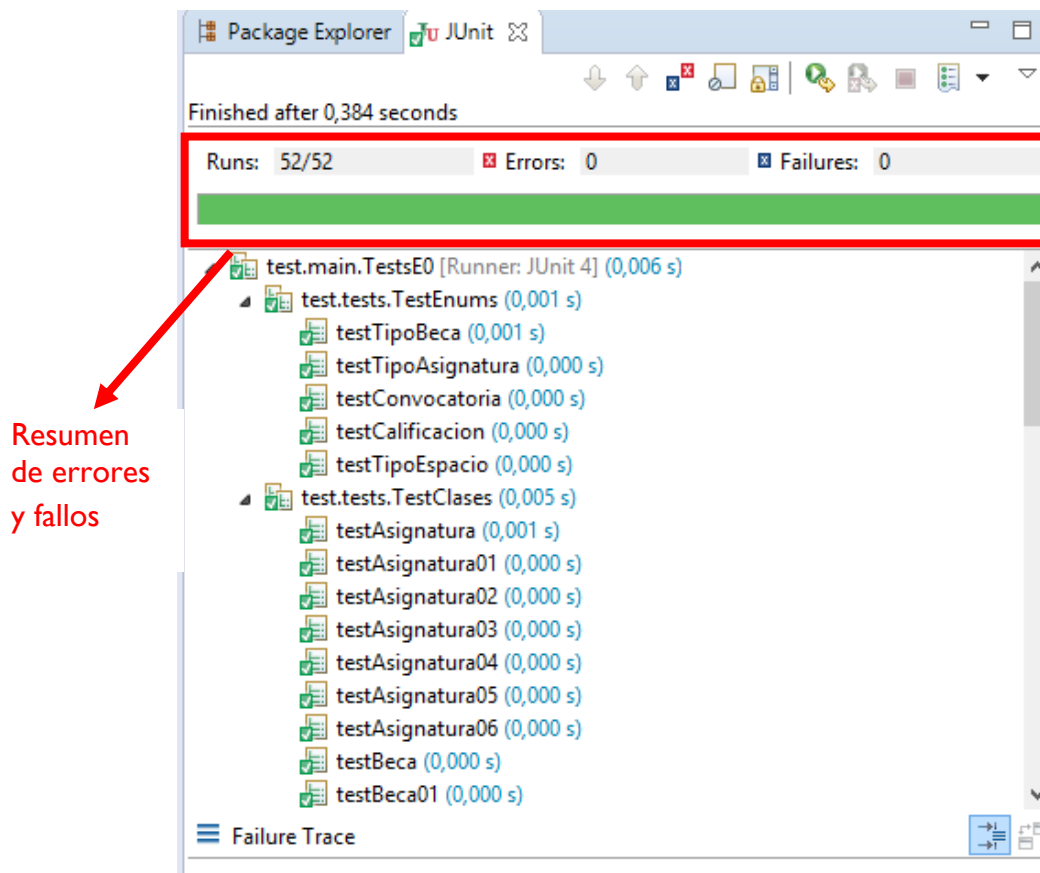


Figura 9. Ejemplo de ejecución de prueba sin errores ni fallos

La Figura 10 muestra una ejecución de un test con un fallo. Fíjese que en el resumen de la ejecución del test el contador del cuadro de texto *Failures* tiene un 1. Además, la barra vertical situada debajo del resumen de errores, que en la Figura 9 era verde, se ha vuelto de color rojo.

Para comprobar el punto concreto en el que se ha producido el error, debe navegar por los distintos tests hasta encontrar el que tiene el icono en azul. En la Figura 10 es el *test17*. Para ver el error concreto que se ha producido, seleccionamos con el ratón el elemento marcado con el icono azul, y nos fijamos en la sección *Failure Trace*. Ahí verá que el problema es el método *getCurso*. En este caso pueden haber ocurrido varias cosas:

- (1) Que el método no haya sido definido en la interfaz.
- (2) Que se haya definido el método pero tenga algún error tipográfico.
- (3) Que el tipo del valor de retorno o el tipo de los parámetros no coincidan con lo esperado.

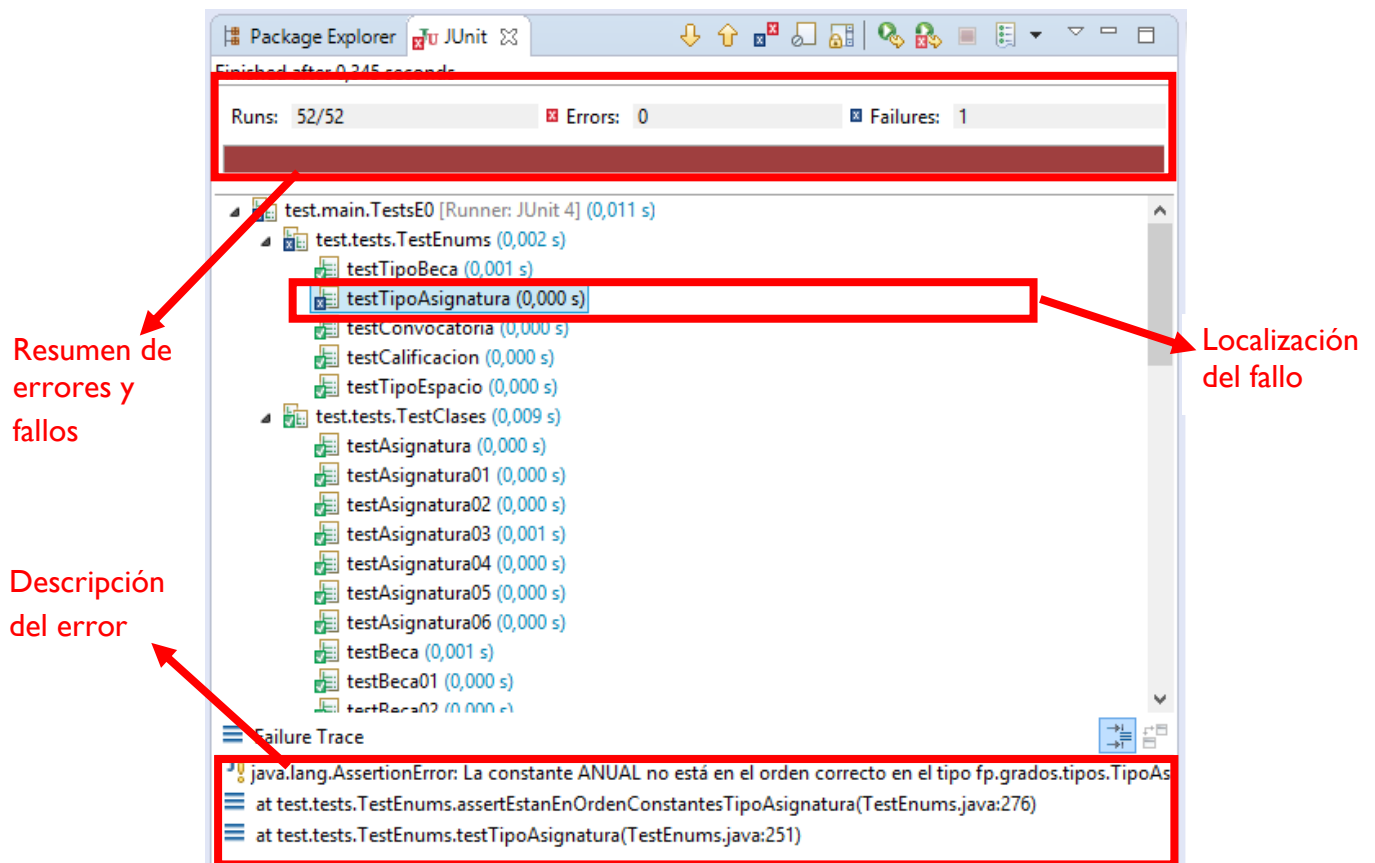


Figura 10. Ejemplo de ejecución del test con un fallo