

Prácticas MES (4º ETSinf) Pruebas

- Si trabajáis en **PoliLabs** acordaros que C: se borra todas las noches y si salís de la sesión puede que a la reentrada estéis en una máquina virtual distinta (con C: diferente)
- Si trabajas en los laboratorios del departamento, arranca eclipse con:
eclipseMES
- Usar el acceso https de los repositorios:
<https://gitlab2.dsic.upv.es/pcarsi/XX.git> (ojo con el .git)
- Los proyectos para la práctica 2 y siguientes están en:
M:\ETSINF\mes\Proyectos
- Ojo con las versiones del JDK.

<https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html>

Proyecto y
repositorio Git de
CORRECTOR

```
Microsoft Windows [Versión 10.0.17763.2114]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

W:\>java -version
java version "11.0.10" 2021-01-19 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.10+8-LTS-162)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.10+8-LTS-162, mixed mode)
```

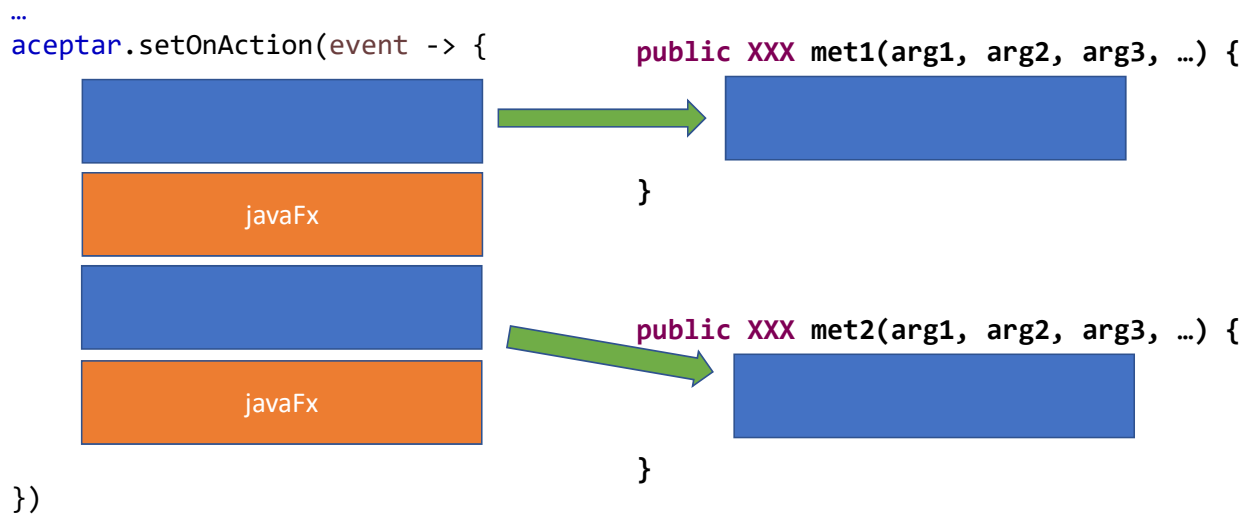
Práctica 4: Pruebas del Código del Caso de Estudio

1. Probar clases de objetos individuales para las clases de la capa de la lógica de negocio. **(NO el controlador)**
 1. Pruebas aisladas de todas la operaciones
 - Técnica del camino básico.
 2. La selección e interrogación de todos los atributos
 3. La ejecución de los objetos en todos los estados posibles
 - En el caso de que el objeto pueda pasar por diversos estados
2. Probar clústers de objetos.
 - Pruebas basadas en los diagramas de secuencia de la práctica 3 de ingeniería inversa.
3. Probar los bugs identificados
 - Hacer pruebas que demuestren que los bugs no existirán cuando sean arreglados.
 - Al principio deberían fallar.

JavaFX y JUnit

- Complicado arrancar el hilo de JavaFX para que funcione en las pruebas.
- SOLUCION: modificar el código para separar el código de JavaFX del resto creando métodos.
 - En las pruebas se invocarán los métodos creados.

JavaFX y JUnit



JavaFX y JUnit


```
...
acceptar.setOnAction(event -> {
    res1 = met1(a, b, c, ...);


    javaFx

    res2 = met2(e, f, g, ...);

    javaFx

})
```

```
public XXX met1(arg1, arg2, arg3, ...) {
    
}


public XXX met2(arg1, arg2, arg3, ...) {
    
}
```


JavaFX y JUnit

```
public class XXTest {

@Test
public void testA() {
    met1(...)
    met2(...)
}

}
```

```
public XXX met1(arg1, arg2, arg3, ...) {
    
}

public XXX met2(arg1, arg2, arg3, ...) {
    
}
```