

Práctica 3. Pruebas de desarrolladores.

Diseñando las pruebas de unidad

- Comprobar que si el cliente define un trabajo, el atributo Cliente.trabajo no esté nulo.
- Comprobar que si al trabajo se le asigne un técnico, el campo Trabajo.tecnicoAsignado no sea nulo, ni el atributo Tecnico.trabajoAsignado tampoco lo sea.
- Comprobar si al dar un presupuesto un técnico a un trabajo, el campo Trabajo.presupuesto fuera actualizado.
- Comprobar que al dar un parte de técnico, al acceder a Trabajo.parteTecnico, este no sea nulo y al obtener sus atributos, estas contengan los valores que hayamos definido previamente.
- Comprobar que al dar un parte de trabajo, al acceder a Trabajo.parteTrabajo, este no sea nulo y al obtener sus atributos, estas contengan los valores que hayamos definido previamente.
- Comprobar que al efectuar el pago, los filtros modifiquen la cantidad a pagar correctamente.

Diseñando las pruebas de componentes

- Comprobar que al pulsar un botón inactivo, siguiéramos en la misma pantalla.
- Comprobar que al pulsar el botón “Calcular presupuesto” nos lleve a la pantalla “Calculadora de presupuestos”.
- Al pulsar el botón “Calcular” con los campos rellenos, se abriría un popup con el asistente de presupuestos (“Asistencia en presupuesto”)
- Al pulsar el botón “Calcular” con algún campo vacío, se quedaría ahí (no aparece “Asistencia en presupuesto”)

Implementación en Dart

Para ello, se crearon dos archivos en Dart, uno con cada tipo de prueba.

En el archivo de pruebas de unidad, se crearon grupos según la clase más interactuada (Trabajo, Cliente y Tecnico) con los tests descritos como en el primer apartado de este documento.

En el archivo de pruebas de widgets, se hicieron 3 tests. Están orientadas a funcionar con el smartphone.

```
I/flutter (24850): (The following exception is now available via  
WidgetTester.takeException:)  
I/flutter (24850): ══ EXCEPTION CAUGHT BY FLUTTER TEST FRAMEWORK  
══════════════════════════════════════════════════════════════  
I/flutter (24850): The following FormatException was thrown running a test:  
I/flutter (24850): Invalid double  
I/flutter (24850):  
I/flutter (24850): When the exception was thrown, this was the stack:  
I/flutter (24850): #0      double.parse  
(dart:core-patch/double_patch.dart:111:28)  
flutter (24850): #1      CalculadoraState.setValues  
package:ds_practica2/VistaTecnico/Calculadora.dart:33  
I/flutter (24850): #2      CalculadoraState.getPresupuestoFromForm  
package:ds_practica2/VistaTecnico/Calculadora.dart:50  
I/flutter (24850): #3      CalculadoraState.build.<anonymous closure>  
package:ds_practica2/VistaTecnico/Calculadora.dart:163  
I/flutter (24850): #4      main.<anonymous closure>  
test\widgetTest.dart:76  
2  
I/flutter (24850): <asynchronous suspension>  
I/flutter (24850): (elided one frame from package:stack_trace)  
I/flutter (24850):  
I/flutter (24850): The test description was:  
I/flutter (24850):   Calcular sin rellenar campos  
I/flutter (24850):
```

```
I/flutter (24850): (If WidgetTester.takeException is called, the above
exception will be ignored. If it is not, then the above exception will be
dumped when another exception is caught by the framework or when the test
ends, whichever happens first, and then the test will fail due to having not
caught or expected the exception.)
I/flutter (24850): ══ EXCEPTION CAUGHT BY FLUTTER TEST FRAMEWORK
════════════════════════════════════════════════════════════════════════════════
I/flutter (24850): The following FormatException was thrown running a test:
I/flutter (24850): Invalid double
I/flutter (24850):
I/flutter (24850): When the exception was thrown, this was the stack:
I/flutter (24850): #0      double.parse
(dart:core-patch/double_patch.dart:111:28)
I/flutter (24850): #1      CalculadoraState.setValues
package:ds_practica2/VistaTecnico/Calculadora.dart:33
I/flutter (24850): #2      CalculadoraState.getPresupuestoFromForm
package:ds_practica2/VistaTecnico/Calculadora.dart:50
I/flutter (24850): #3      CalculadoraState.build.<anonymous closure>
package:ds_practica2/VistaTecnico/Calculadora.dart:163
I/flutter (24850): #4      main.<anonymous closure>
test/widget_test.dart:76
2
I/flutter (24850): <asynchronous suspension>
I/flutter (24850): (elided one frame from package:stack_trace)
I/flutter (24850):
I/flutter (24850): The test description was:
I/flutter (24850):   Calcular sin rellenar campos
I/flutter (24850):
════════════════════════════════════════════════════════════════════════════════
I/flutter (24850): 00:30 +2: Calcular sin rellenar campos [E]
I/flutter (24850):   Test failed. See exception logs above.
I/flutter (24850):   The test description was: Calcular sin rellenar campos
I/flutter (24850):
I/flutter (24850): 00:30 +2 -1: Some tests failed.
```

Al parecer, los fallos en Android Studio provenían de intentar ejecutarse en Chrome, si bien en smartphone pone “Errores de shader”.