

# Plan de Pruebas

## Introducción

Este documento refleja cómo se trabaja en **RTSoftware** en lo que se refiere a las pruebas del producto software antes de lanzarlo al mercado. Para ello, elaboramos casos de prueba aplicando el nivel de cobertura: **cobertura de decisiones**.

## Objetivo

El objetivo del plan de pruebas es encontrar fallos antes del paso del producto software a la fase de entrega. Los testers se encargan de dicha tarea analizando el código fuente y elaborando casos de prueba con el propósito de encontrar los máximos errores posibles para futuras correcciones. Con esto, obtenemos una mejor cobertura del funcionamiento del producto software siendo así más fiable en lo que se refiere a los fallos.

## Tareas

Las tareas que se van a realizar en la fase de pruebas son las siguientes:

| Tareas   | Encargado | Herramientas utilizadas |
|--|-----------|-------------------------|
| Hacer testing del Software para encontrar fallos   | Testers   | Eclipse, Junit, Maven   |
| Elaboración de casos de pruebas para los distintos métodos de las distintas clases                         | Testers   | Word, Eclipse           |
| Realizar pruebas del Software usando los casos de prueba y procedimientos de prueba                        | Testers   | Eclipse, Junit, Maven   |
| Documentar los resultados de las pruebas realizadas para una futura corrección en la fase de mantenimiento | Testers   | Word                    |

## Productos

Productos de entrada: código fuente, casos de prueba.

Productos de salida: fallos en el código, documentación de errores.

## Resumen

| Tarea  | Entradas                       | Salida                                    |
|--|--------------------------------|---|
| Hacer testing del Software para encontrar fallos                                   | Código fuente                  | Código fuente revisado                    |
| Elaboración de casos de pruebas para los distintos métodos de las distintas clases | Código fuente                  | Casos de prueba                           |
| Realizar pruebas del Software  | Código fuente, casos de prueba | Pruebas realizadas sobre el código fuente |
| Documentar los resultados de las pruebas realizadas                                | Casos de prueba                | Documento de reporte de pruebas           |

## Procedimiento a seguir para la elaboración de casos de prueba

Con el objetivo de minimizar el esfuerzo y maximizar el resultado obtenido durante la creación de las pruebas, el procedimiento elegido es el siguiente:

1. Contabilizar en líneas de código cada clase de nuestro producto software.
2. Asignar una preferencia de mayor a menor teniendo en cuenta la extensión en líneas de código de cada clase.
3. Contabilizar en líneas de código cada método de cada clase.
4. Asignar una preferencia de mayor a menor teniendo en cuenta las líneas de código de cada método.
5. Una vez realizados los pasos anteriores, empezamos a elaborar los casos de prueba, empezando por los métodos que tienen asignada más prioridad.
6. En cada método definimos los parámetros de entrada, las clases de equivalencia de dichos parámetros y el conjunto de valores de prueba.
7. Dada la información del paso número 6, elaboramos los casos de prueba utilizando el criterio de cobertura: **cobertura de decisiones**.
8. Probamos los casos de prueba en eclipse mediante Junit con el objetivo de encontrar fallos.
9. Reportamos errores.

# Ejecución del Plan de Pruebas

## Ejecución

Para la ejecución del plan hemos creado una rama en el proyecto llamada *Testing*, con el objetivo de introducir allí todas las modificaciones necesarias.

Una vez que las pruebas han sido codificadas, probadas y aprobadas por el equipo, se realiza un *merge* al *origin/master* del proyecto. En el repositorio del proyecto se puede observar esta dinámica de trabajo con más detalle.

Se ha codificado un total de 48 tests siguiendo el procedimiento explicado con anterioridad en el Plan de Pruebas. El objetivo marcado por el cliente era obtener un 75% de cobertura de líneas de código. Con nuestra técnica de asignar prioridad a los métodos que vamos a probar, dicho objetivo se cumplió al realizar 30 pruebas. En vez de parar aquí, el equipo estimó oportuno seguir realizando más pruebas hasta conseguir cubrir cerca del 85% de cobertura en líneas de código y el 95% en cobertura de decisiones.

Por cada cambio realizado y aprobado se ha generado una nueva versión del producto, siguiendo la implantación del Proceso de Gestión de Configuración que ha realizado la empresa **RTSoftware**. Cada versión está marcada y es descargable desde el propio repositorio del proyecto.

## Coste

Las pruebas del proyecto han durado una semana y dado el equipo de pruebas de la empresa **RTSoftware** (un empleado), su sueldo, y el coste de las operaciones de la propia empresa (todo está detallado en la sección de Estructura Empresarial en nuestra Web), el coste de las pruebas es de:

$$\text{Coste}_{\text{TOTAL}} = (\text{Coste}_{\text{NETO}} + \text{BurnRate}_{\text{PERIODO}}) * 1.1$$

$$\text{Coste}_{\text{TOTAL}} = (1600\text{€} + 6490\text{€}) * 1.1$$

$$\text{Coste}_{\text{TOTAL}} = 8.899 \text{ €}$$

## Situación final

Tras haber implementado el Plan de Pruebas en el proyecto Sistema Bancario, con un total de **48** pruebas, hemos obtenido una cobertura del **87%** en líneas de código, y un **98%** en cobertura de decisiones. Tres de las pruebas han detectados errores de lógica de negocio en el proyecto y serán estudiadas y tratadas durante el Mantenimiento.

El resultado se puede observar tanto en el repositorio del proyecto como en el informe generado que está disponible online en la Wiki y en la Web del proyecto.

Cabe destacar que la *release* del proyecto que incluye todas las pruebas y el informe generado es la versión: **0.2.1**, y como se ha mencionado antes, está correctamente etiquetada y descargable desde el repositorio.