

UT3 - DISEÑO Y REALIZACIÓN DE PRUEBAS –

ACTIVIDADES - PARTE III

RESULTADOS DE APRENDIZAJE ASOCIADOS
3.- Verifica el funcionamiento de programas diseñando y realizando pruebas.
CRITERIOS DE EVALUACIÓN
a) Se han identificado los diferentes tipos de pruebas.
b) Se han definido casos de prueba
c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.
d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.
e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.
f) Se han efectuado pruebas unitarias de clases y funciones.
g) Se han implementado pruebas automáticas.
h) Se han documentado las incidencias detectadas.

ACTIVIDAD 3.7

En el proyecto Java que se adjunta, hay definida una clase de nombre **CCuenta**

Esta clase dispone de varios métodos, entre ellos **ingresar** y **retirar**.

1. Realiza una ejecución paso a paso, que verifique el correcto funcionamiento de la aplicación. Indica los valores que marca la inspección de variables tras ejecutar la instrucción **miCuenta.retirar(2300)** en la función **main**
2. Diseña un caso de prueba que permita verificar el método ingresar.
3. Diseña un caso de prueba que permita verificar la función retirar.
4. Diseña un caso de prueba que permita verificar ingresar y retirar al mismo tiempo.
5. Ejecuta las pruebas y comenta el resultado.
6. Documenta las pruebas realizadas.



NOTAS:

- Los apartados 2,3,4 y 5 se deben hacer usando **casos de prueba a mano** y **usando pruebas automatizadas**.
- Los ejercicios se podrán realizar tanto con **Eclipse** como con **Netbeans** (elegid uno de los dos entornos). Si los hacéis con ambos IDEs, lo tendré en cuenta positivamente.



IMPORTANTE: Explica todos los apartados paso a paso cómo los realizaste. No olvides las pruebas de vida. Es decir, añadir un .txt con tu nombre y curso (como ya hicimos en actividades anteriores). **Si esto, la actividad NO será válida.**

Consejos y recomendaciones. Se pretende que, partiendo de una clase con dos métodos ya diseñados, se utilicen herramientas de depuración para corregir errores en el código. A continuación, se deberá diseñar al menos dos casos de prueba, para verificar el comportamiento de los métodos ingresar y retirar