

Este documento contiene la respuesta a muchas de las inquietudes más comunes de los estudiantes respecto al reto final, así como **una** corrección al valor que debe dar el punto **3b**. Si usted ya realizó el reto, y lo hizo de forma correcta, basta con cambiar este número en el assert de la prueba *testPunto1* y todo lo demás funciona exactamente igual.

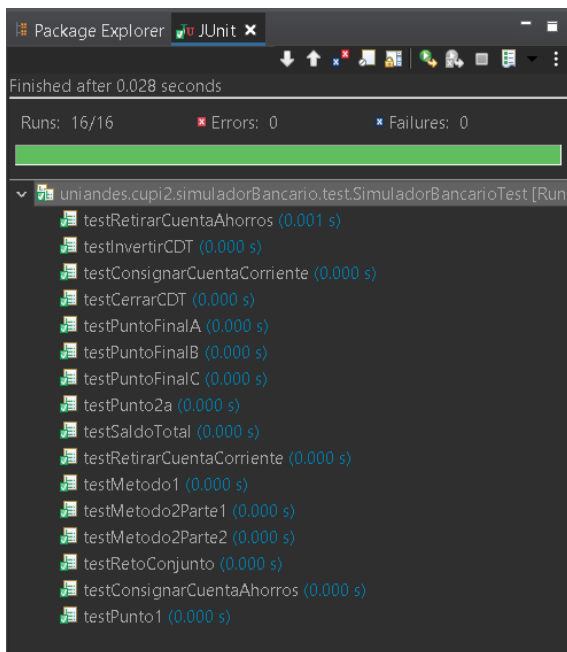
El día Jueves 19 se les subirá a la plataforma un **video demostrativo** para que puedan ver el funcionamiento del programa.

**Corrección:** El valor que debe dar el punto 3b (*testPunto1*) es **\$1,902,800**.

#### Aclaraciones:

- El programa debe superar todas las pruebas automáticas de la clase *SimuladorBancario* (Las que ya están creadas y las que usted debe crear).

Debería verse así:



Nota: En la imagen de ejemplo hay 16 pruebas porque incluye las pruebas ocultas que realizará la plataforma. En su caso, usted tendrá un menor número pero aún así debe superarlas todas.

- Al realizar los puntos 1 y 2 notará que las pruebas automáticas de la clase *CDT* le dan un error de sintáxis. Debe corregirlos (simplemente agregando el bloque de manejo de errores correspondiente que eclipse recomienda por defecto: en caso de excepción imprima el *trace* en consola)
- En el punto 1, si se intenta abrir un nuevo *CDT* cuando ya existe uno abierto, debe lanzar la excepción *solicitud y dejar el cdt original como estaba*. **NO abra el nuevo cdt ni sobrescriba el viejo.**
- Para el punto 2, recuerde que el Invariante solamente alerta al usuario de los errores básicos que suceden en la ejecución del código, no los corrige.
- En consecuencia, con la aclaración anterior, y si hizo bien los puntos 1 y 2. Al ejecutar la prueba *testPunto2a* (Del enunciado: 3c) encontrará aserciones de violaciones de la invariante. **Comente, elimine o corrija.** Según considere, las llamadas a los métodos que causan estos problemas.
- En la prueba *testPunto2a* (Del enunciado: 3c) el saldo que se debe validar es el **saldo total**
- El **consecutivo** de las transacciones empieza en 1. Es decir, la primera transacción que se realice será el consecutivo 1, la segunda será el consecutivo 2 y así sucesivamente.
- Considere qué, al **ejecutar el método 2 (Reiniciar simulación)** también se deben reiniciar las transacciones.

- En el bono A puede hacerse agregando **una línea** (sin contar el lanzamiento de la excepción ni el bloque try-catch)
- Las (primeras) transacciones detalladas que deberían verse en el bono B se muestran en la siguiente tabla:

consecutivo	tipoTransaccion	tipoCuenta	valor	Descripción
1	ENTRADA	AHORROS	\$1,000,000.00	Consignación cuenta de ahorros
2	ENTRADA	CDT	\$50,000.00	Apertura del CDT
3	ENTRADA	AHORROS	\$6,000.00	Intereses cuenta de ahorros mes 1
4	ENTRADA	CDT	\$500.00	Intereses CDT mes 1
5	ENTRADA	AHORROS	\$6,036.00	Intereses cuenta de ahorros mes 2
6	ENTRADA	CDT	\$500.00	Intereses CDT mes 2
7	ENTRADA	AHORROS	\$6,072.22	Intereses cuenta de ahorros mes 3
8	ENTRADA	CDT	\$500.00	Intereses CDT mes 3
9	ENTRADA	AHORROS	\$6,108.65	Intereses cuenta de ahorros mes 4
10	ENTRADA	CDT	\$500.00	Intereses CDT mes 4
11	SALIDA	CDT	\$52,000.00	Retiro CDT (Cierre)
12	ENTRADA	CORRIENTE	\$52,000.00	Al cerrar el CDT, el saldo entra a cuenta corriente
13	ENTRADA	AHORROS	\$6,145.30	Intereses cuenta de ahorros mes 5
14	SALIDA	AHORROS	\$500,000.00	Retiro cuenta ahorros
...				

- Siempre y cuándo logre la funcionalidad deseada, usted es libre de crear la nueva clase (Transaccion) cómo considere: Puede agregar, cambiar o incluso quitar los métodos y atributos sugeridos por el enunciado en el diagrama de clases.
- Note qué, **cada ingreso de interés mensual es una transacción distinta**. Independientemente de si este se genera con el método avanzarMesSimulacion() o con el método 1 del reto 1 (Acelerar simulación)
- No importa el orden en el que cree los intereses. Por ejemplo, **el consecutivo de las transacciones 3 y 4 podrían ser intercambiados**. Sin embargo, **al cerrar el CDT primero debe salir el dinero de la inversión para posteriormente poder consignarla a la cuenta de corriente**. Es decir, los consecutivos de las transacciones 11 y 12 **NO** son intercambiables.
- Finalmente, **los puntos de las transacciones y los puntos del manejo de errores son independientes. No se evalúa manejo de errores para las transacciones**. Es decir, **NO** necesita considerar casos tales cómo: “No crear una nueva transacción de inversión si se intenta abrir un nuevo CDT pero ya hay uno abierto”. Si lo desea, por aprendizaje propio puede pensar en cómo trataría estos casos.