

Ejercicio 3

Objetivo

Modificar la aplicación desarrollada en el [Ejercicio 2](#) anterior para reflejar los nuevos requisitos que se describen a continuación.

Nuevos requisitos: cotitulares y transacciones

- Las cuentas deben tener, además del CBU, un alias (una combinación de texto y números). El alias debe ser único (no puede haber dos cuentas con el mismo alias). Las cuentas son dadas de alta por los oficiales de cuenta de cada sucursal.
- Las transferencias entre cuentas se deben poder hacer ahora también mediante el alias (además del CBU).
- Interesa saber en qué sucursal del banco está radicada cada cuenta. Por ejemplo, la cuenta con alias "Ing.Soft.FIUBA" está radicada en la sucursal 984, denominada "Suc. Belgrano", que está ubicada en la dirección "Cabildo 3456 CABA". La Gerencia de sucursales es la encargada de dar de alta nuevas sucursales, modificar información de las actuales y dar de baja (lógica) aquellas que no operarán más.
- Cada cuenta pertenece a uno o varios clientes. Uno de los clientes debe ser titular de la cuenta y los restantes son cotitulares. No todas las cuentas tienen cotitulares. Cada cliente se identifica mediante su número de DNI y se necesita conocer su apellido y nombre, fecha de nacimiento y domicilio. Los clientes se dan de alta antes de sus cuentas.
- Es necesario saber si los clientes son cónyuges. En el caso de que lo sean, se necesita saber la fecha de casamiento.
- Por razones legales, es necesario que cada transacción realizada con una cuenta (retiro, depósito) o entre cuentas (transferencia) quede registrada. Cada transacción debe tener un número correlativo que la identifique; se deben conocer fecha y hora de realización, tipo de transacción y monto; y debe estar asociada a la cuenta o a las cuentas involucradas.

Estructura del repositorio

Branch

Crear una nueva rama (*branch*) llamada Ejercicio3.

Docs

Sobre esta carpeta del repositorio deberás agregar (a excepción de las Historias de Usuario que se representan con *Issues*) subcarpetas con los siguientes documentos en formato PDF:

HDD

Historias de dominio (que reflejen la operatoria descrita en los Ejercicios 1, 2 y 3).

MDD

Modelo de dominio con la representación de las entidades, sus atributos y relaciones entre ellas. Recordá en caso de que lo consideres necesario, agregar las notas correspondientes para interpretar el modelo.

DC

Diagrama de contexto (en notación C4 primer nivel).

Reports

Dentro de esta carpeta se deben alojar los reportes generados por el testing (unitario y funcional)

Issues (Historias de Usuario)

Crear un *issue* para cada nueva historia de usuario (basadas en las historias de dominio).

- Cada *issue* debe tener como título el nombre de la historia de usuario
- En el campo descripción se debe incluir la historia propiamente dicha. Ejemplo:
 - “Como cliente, quiero transferir fondos de mi cuenta a la cuenta de otra persona, para así poder cancelar mis obligaciones de pago”.
- Recordá completar el resto de los campos del *issue*:
 - asignado (vos)
 - etiquetas
- Una vez creado el *issue*, se debe asociar a la rama (*branch*) correspondiente al Ejercicio 3.

Desarrollo

Usando *Codespaces* o el *IDE* de tu preferencia, hacé lo siguiente (sobre la rama Ejercicio3):

- Crear un archivo *.feature* por cada historia de usuario. Describir los escenarios siguiendo el formato *Gherkin* ya conocido. Considerar los caminos felices y los no felices. Es importante que hagas esto antes de codificar.
- Modificá el código para reflejar las nuevas funcionalidades y requisitos. Identificá nuevas clases a partir de las entidades del modelo de dominio.
- Definí las pruebas unitarias antes de codificar. Agregá los pasos de prueba para los escenarios *Gherkin* y las pruebas unitarias necesarias.
- Agregá en *Main.java* la creación de cuentas asociadas a titulares, la transferencia de fondos entre dos cuentas por alias y presentá por consola los resultados obtenidos.
- Asegurate de que la aplicación compile y funcionen todas las pruebas.

Gestión de la configuración

- Se espera que, al visualizar los *commits* en el repositorio, el desarrollo de la consigna se refleje de manera iterativa e incremental. Cada *commit* debe mostrar

avances graduales, siguiendo los lineamientos proporcionados (ver sección Consejo) y aplicando buenas prácticas en el desarrollo de software. Esto incluye la organización clara del código, la gestión adecuada de versiones y el uso descriptivo de mensajes de *commit*. Además, deben generar y mantener documentación como las historias de dominio, el modelo de dominio y el diagrama de contexto, asegurando que estas también evolucionen a medida que avanza el proyecto y sean consistentes.

- Una vez que estás seguro del desarrollo (todo desarrollado, funcionando, *test ok*, *commits*, reportes en formato PDF, etc.), deberás cerrar cada *issue* (historia de usuario).
- En Github, creá un *pull-request*.
- Esperá a que corran los chequeos y, una vez listos, hacer la integración con la rama *main* (botón *Merge Pull-Request*)

Consejo

Tené presente los [lineamientos](#) para el desarrollo de

- Historias de dominio
- Modelo de dominio
- Historias de Usuario

y los lineamientos generales: de los requisitos al diseño.

Entrega

El ejercicio se considera entregado al momento de enviar el enlace al repositorio que apunte a la rama *main*, al momento de marcar como entregada la actividad en *google classroom*.