

## Mentoría No.9 - Proyecto sistema bancario

### Ejercicio autónomo - Calificable

Tiempo estimado: 6 sesiones

#### Conceptos a trabajar

- Programación orientada a objetos
- Abstracción de un problema
- Bases de datos
- Git

#### Objetivos

Construir un proyecto desde cero que involucre todos los conceptos trabajados durante el Ciclo uno.

#### Proyecto

##### Sistema bancario - Mock Nequi

Codum Academy quiere construir un prototipo basado en el servicio creado por Bancolombia para transferir dinero entre personas, cuyo nombre será Mock Nequi.

Todos los usuarios de Mock Nequi contarán con una única cuenta de ahorros en la cual se almacenará su dinero, y estará disponible para ser retirado o transferido a otros usuarios.

Además de su cuenta de ahorros, cada usuario cuenta con un colchón, cero o más bolsillos, y cero o más metas.

Un colchón es un lugar donde el usuario puede guardar dinero que no estará disponible para ser retirado o transferido. Es como guardar la plata debajo del colchón para casos de emergencia.

Un bolsillo, es una forma de dar propósito al dinero disponible. Por ejemplo, Es posible crear un bolsillo para alimentación y transferir plata a dicho bolsillo, proveniente del dinero disponible en la cuenta.

Una meta, es un ahorro a largo plazo con un nombre, una fecha, y un monto total. Por ejemplo, es posible crear una meta con el fin de ahorrar \$1'000.000 de pesos para comprar un celular el 20 de enero del 2019. Con Mock Nequi, será posible transferir dinero a una meta, proveniente del dinero disponible en la cuenta.

El dinero disponible en una cuenta de Mock Nequi está dado por la siguiente fórmula.

**Disponible** = Total en la cuenta - dinero en colchón - dinero en metas - dinero en bolsillos

El objetivo de este proyecto es construir un sistema basado en consola que permita:

1. Registrar nuevos usuarios con su nombre, email y contraseña.
2. Iniciar sesión de un usuario con su email y contraseña. Cada usuario podrá realizar alguna de las siguientes opciones:
  - a. Consultar el saldo de disponible en su cuenta.
  - b. Consultar el saldo total en su cuenta.
  - c. Ingresar una cantidad determinada de dinero a su cuenta.
  - d. Retirar una cantidad determinada de dinero de su cuenta.
  - e. Enviar dinero a otro usuario a través de su email, desde cualquier cuenta de ahorros.
  - f. Consultar sus últimas N transacciones (ingresos, retiros, recepciones y envíos).
  - g. Entrar al menú del colchón, donde podrá:
    - i. Consultar el dinero guardado en el colchón.
    - ii. Agregar dinero disponible al colchón.
    - iii. Retirar dinero del colchón para que vuelva a estar disponible en la cuenta.
    - iv. Regresar al menú anterior.
  - h. Entrar al menú de bolsillos, donde podrá:
    - i. Listar la información de todos sus bolsillos (nombre y saldo).
    - ii. Crear un bolsillo nuevo. El bolsillo se crea con un nombre y un saldo de 0.
    - iii. Eliminar un bolsillo. Esto implica que el dinero que esta en dicho bolsillo, vuelve a estar disponible en la cuenta del usuario.
    - iv. Agregar dinero a un bolsillo.
    - v. Retirar dinero de un bolsillo.
    - vi. Enviar dinero de un bolsillo a otro usuario a través de su email.
    - vii. Regresar al menú anterior.
  - i. Entrar al menú de metas, donde podrá:
    - i. Listar la información de todas sus metas: nombre, monto total, dinero ahorrado, dinero restante para cumplir la meta, el estado actual (cumplida o vencida) y la fecha límite.
    - ii. Crear un nueva meta. Nombre, monto total y fecha límite.
    - iii. Cerrar una meta. Esto implica que el dinero que esta en dicha meta, vuelve a estar disponible en la cuenta del usuario.
    - iv. Agregar dinero a una meta.
    - v. Regresar al menú anterior.
  - j. Cerrar sesión (volver al menú anterior).
3. Salir del programa.

Las restricciones el problema son las siguientes:

1. La contraseña no debe ser almacenada en texto plano. Se debe usar una codificación SHA1 para guardarlas y validarlas.

2. Un usuario solo puede enviar o retirar el dinero disponible en su cuenta.
3. Si un usuario desea retirar o enviar dinero de uno de sus bolsillos, solo podrá hacerlo con el dinero almacenado en dicho bolsillo.
4. El dinero que puede ser enviado al colchón, un bolsillo o una meta, debe provenir del dinero disponible en la cuenta.
5. Si alguna operación no cuenta con el dinero disponible para ser exitosa, un mensaje claro de error debe ser mostrado al usuario.

### **Criterios de evaluación**

- El proyecto debe ser entregado en equipo de 2 personas. Máximo 3 si el grupo es impar.
- La entrega debe realizarse a través de un repositorio en Github, y la colaboración entre los integrantes del equipo debe ser mediante Pull Requests.
- El programa debe funcionar correctamente.
- La solución del problema debe evidenciar una abstracción clara y coherente.
- El código de la solución debe ser claro y fácilmente entendible para un programador externo.
- Se evaluará que tan fácil es agregar nuevas opciones al menú del usuario.

**Recomendación:** Separe la lógica de interfaz de usuario (UI), de la lógica de negocio.

### **Bibliografía**

- <https://www.nequi.com.co>

### **Anotaciones post mentoría**

Listado de conceptos adicionales o preguntas adicionales realizadas a los estudiantes después de haber finalizado la mentoría, junto con un pequeño resumen de cada cosa.