Diseño general

En esta sección se justifican los principios SOLID aplicados:

## 1. Responsabilidad Única (SRP)

Cada clase se centra en una única responsabilidad. Por ejemplo, `ScreenController` solo gestiona el cambio de escenas.

## 2. Abierto/Cerrado (OCP)

Componentes como el sistema de validaciones pueden extenderse con nuevas reglas sin modificar el código existente, gracias a interfaces en `validations/`.

## 3. Sustitución de Liskov (LSP)

Todas las implementaciones de validadores cumplen el contrato de la interfaz `Validator`, de modo que pueden ser intercambiables.

## 4. Segregación de Interfaces (ISP)

Los controladores de vista exponen solo los métodos que realmente usan las ventanas FXML.

## 5. Inversión de Dependencias (DIP)

La lógica de negocio (`logic/`) depende de abstracciones (interfaces), no de implementaciones concretas.

**Patrones de Diseño Aplicados**

- Strategy: Para las distintas validaciones de formularios.

- Singleton: Para la gestión de la configuración global o conexión a datos.

- Factory Method: Para instanciar controladores de vista basados en un enum de rutas FXML.

- Observer: En caso de eventos de UI que requieran notificar múltiples componentes.

-- Script de creación de tabla Strongbox

CREATE TABLE Strongbox (

id SERIAL PRIMARY KEY,

usuario\_id INTEGER NOT NULL,

nombre VARCHAR(100) NOT NULL,

saldo DECIMAL(15,2) DEFAULT 0.00,

fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Índice para búsquedas por usuario

CREATE INDEX idx\_strongbox\_usuario ON Strongbox(usuario\_id);