

Wallet

Para nuestra primera evaluación se nos entregó una problemática, la cual consiste en dar una solución demostrando habilidades de programación en java. La idea es realizar una pequeña aplicación que funcione a través de consola, la cual deberá mantener activos financieros.

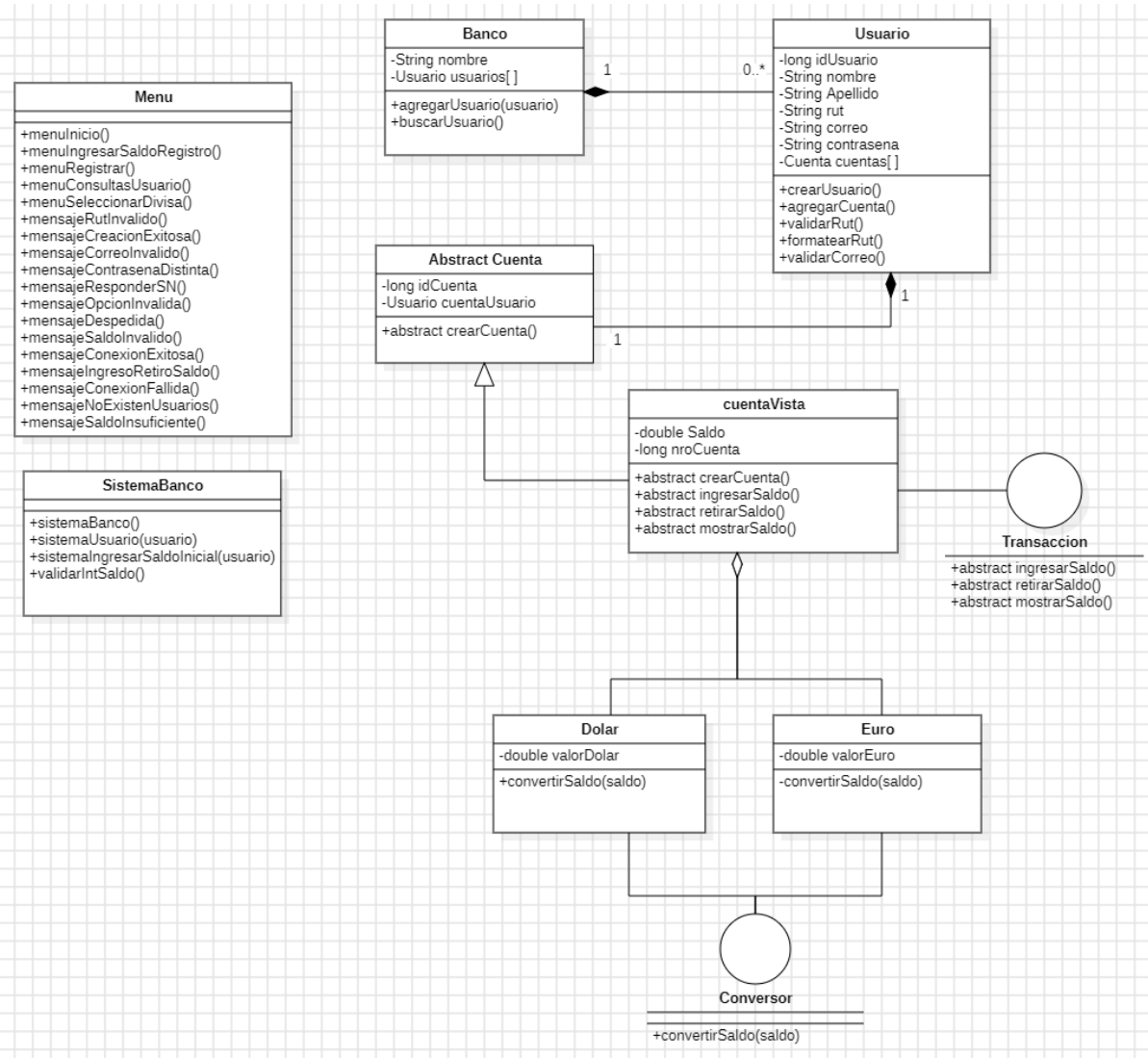
Características y funciones necesarias:

- Solución Segura.
- Fácil de usar.
- Administrar activos financieros.
- Transacciones
- Administración de fondos
- Conversión de moneda
- Java implementando POO
- Diagrama de clase
- Pruebas Unitarias

En cuanto a los aspectos técnicos se tomará en cuenta lo siguiente.

1. **Legibilidad del código:** Es de suma importancia que el código esté bien organizado y legible, incorporando buenas prácticas en el desarrollo.
2. **Documentación:** Es importante documentar el código, explicando las funcionalidades de este. Realizando comentarios, utilizando JavaDoc entre otros.

Diagrama de Clases



Muestra proyecto y código

A continuación se muestra el menú inicial del banco, el cual te permitirá realizar las siguientes funcionalidades mostradas en la imagen.

```
=====
| BIENVENIDO A WALLET BANK |
=====
| Opciones:                |
| 1. Conectar              |
| 2. Registrar             |
| 3. Salir                 |
|                           |
=====
Opción: _
```

Si presionamos la opción número 2 nos comenzará a solicitar los datos para realizar el registro. Además de cumplir con distintas validaciones en caso de proporcionar datos incorrectos. Una vez finalizado, volveremos al menú inicial.

```
=====
| REGISTRO                 |
=====
| PROPORCIONE LOS DATOS   |
| SOLICITADOS              |
|                           |
Ingrese Nombre: Javier
Ingrese Apellido: Alcántara
Ingrese Rut: 18298640-2
Ingrese Correo: javier@gmail.com
Ingrese Contraseña: 12345
Ingrese Contraseña nuevamente: 12345
=====
| REGISTRO COMPLETADO      |
| ¿DESEA INGRESAR SALDO?  |
| ( S / N )               |
|                           |
=====
Respuesta: s
Ingrese saldo: 10000
=====
| CUENTA CREADA CON ÉXITO |
=====
Presione enter para continuar...
```

A continuación si vamos a la opción número 1 el programa solicitará el correo y la contraseña para luego desplegar el menú de usuario bancario.

```
Ingrese Correo: javier@gmail.com
Ingrese Contraseña: 12345
=====
|          CONEXIÓN ÉXITOSA          |
=====
|          CONSULTAS AL BANCO        |
=====
|  Opciones:                         |
|      1. Ingresar Saldo             |
|      2. Retirar Saldo              |
|      3. Cambio Divisa              |
|      4. Ver Saldo                  |
|      5. Salir                      |
=====
Opción:
```

Si deseamos ingresar saldo, se deberá colocar la opción 1, donde a continuación solicitará el saldo a ingresar y finalmente mostrará un mensaje de éxito y su nuevo saldo.

```
=====
|          CONSULTAS AL BANCO        |
=====
|  Opciones:                         |
|      1. Ingresar Saldo             |
|      2. Retirar Saldo              |
|      3. Cambio Divisa              |
|      4. Ver Saldo                  |
|      5. Salir                      |
=====
Opción: 1
Ingrese Cantidad a Ingresar: 50000
=====
|          SALDO INGRESADO CON ÉXITO  |
|          SALDO ACTUAL               |
=====
| Pesos                               | $60,000 |
=====
Presione enter para continuar...
```

Para retirar, es el mismo proceso, vamos a opción 2, se solicita saldo a retirar y muestra mensaje de éxito al retiro y su nuevo saldo.

```
=====
| CONSULTAS AL BANCO |
=====
| Opciones: |
| 1. Ingresar Saldo |
| 2. Retirar Saldo |
| 3. Cambio Divisa |
| 4. Ver Saldo |
| 5. Salir |
=====
Opción: 2
Ingrese Cantidad a Retirar: 15000
=====
| RETIRO DE SALDO EXITOSO |
| SALDO ACTUAL |
=====
| Pesos $45,000 |
=====
Presione enter para continuar...
```

La siguiente opción es para poder realizar un cambio de divisa u conversor, lo que hará será mostrar a cuanto equivale nuestra divisa en dólares o euros.

```
=====
| CONVERSION DIVISA |
=====
| Opciones: |
| 1. Dólar |
| 2. Euro |
=====
Opción: 1
=====
| SALDO EN DÓLAR |
| 46.43 |
=====
Presione enter para continuar...
```

Finalmente, la última opción es para ver el saldo disponible en la cuenta bancaria.

```
=====
| CONSULTAS AL BANCO |
=====
| Opciones: |
| 1. Ingresar Saldo |
| 2. Retirar Saldo |
| 3. Cambio Divisa |
| 4. Ver Saldo |
| 5. Salir |
=====
Opción: 4
=====
| SALDO DISPONIBLE |
=====
| Pesos $45,000 |
=====
Presione ENTER para continuar...
```

Pruebas Unitarias

Se mostrarán algunos ejemplos de las pruebas realizadas y se adjuntará una imagen del coverage.

```
@Test
public void buscarUsuario() {
    String correoIngreso = "javier.alcantara.canto@gmail.com";
    String contrasena = "12345";

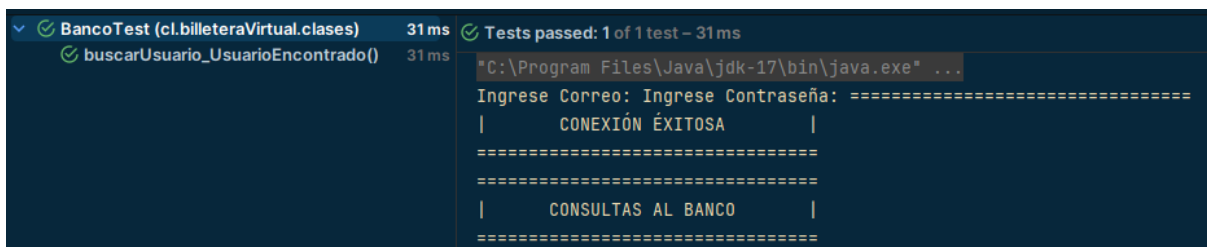
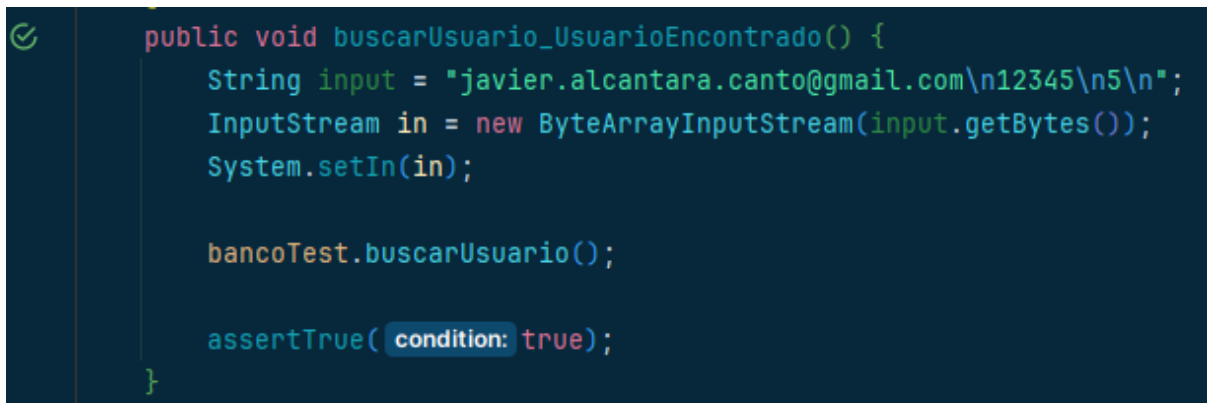
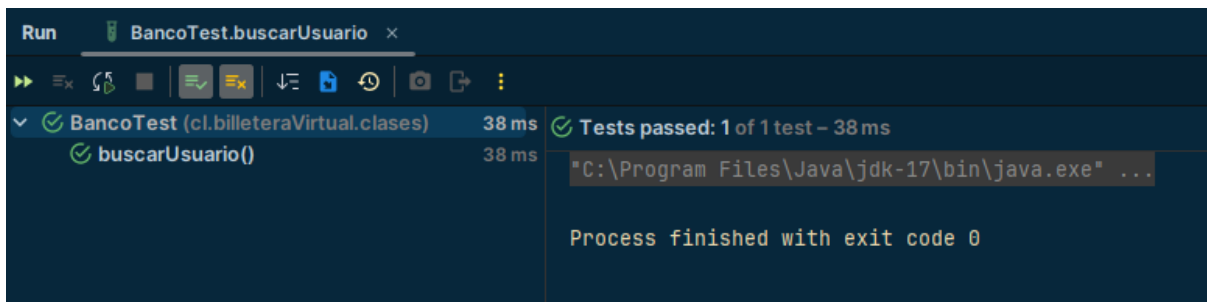
    String input = correoIngreso + "\n" + contrasena + "\n";
    InputStream in = new ByteArrayInputStream(input.getBytes());
    System.setIn(in);
    Scanner scanner = new Scanner(System.in);

    // Rescato de datos simulados
    String correoIngresoComparacion = scanner.nextLine();
    String contrasenaComparacion = scanner.nextLine();

    boolean encontrado = false;

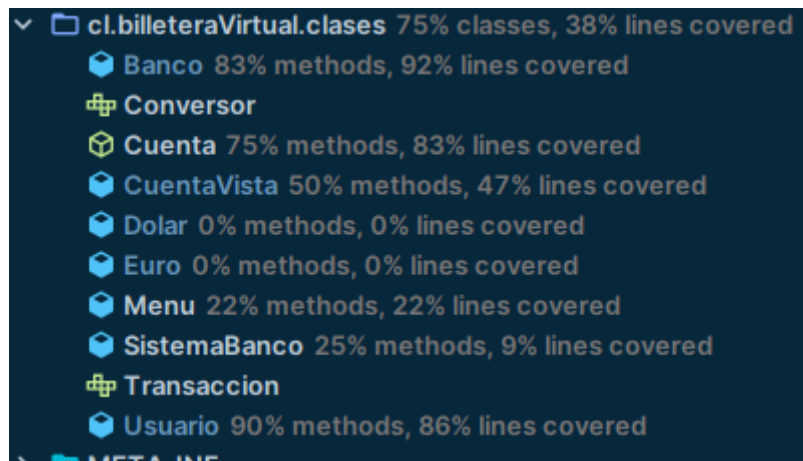
    if (bancoTest.getListaUsuarios().isEmpty()) {
        Menu.mensajeNoExistenUsuarios();
    } else {
        for (Usuario u : bancoTest.getListaUsuarios()) {
            if (u.getCorreo().equalsIgnoreCase(correoIngresoComparacion) && u.getContrasena().equals(contrasenaComparacion)) {
                encontrado = true;
                break;
            }
        }
    }

    assertTrue(encontrado);
}
```



Coverage

Coverage Whole project x			
Element ^	Class, %	Method, %	Line, %
cl.billeteraVirtual.clases	60% (3/5)	69% (18/26)	62% (113/1...
Banco	100% (1/1)	83% (5/6)	92% (25/27)
CuentaVista	100% (1/1)	50% (4/8)	47% (23/48)
Dolar	0% (0/1)	0% (0/1)	0% (0/15)
Euro	0% (0/1)	0% (0/1)	0% (0/15)
Usuario	100% (1/1)	90% (9/10)	86% (65/75)



En conclusión, el desarrollo de esta aplicación en Java ha sido una experiencia beneficiosa, ya que me ha ayudado a comprender mejor lo que es la programación orientada a objetos junto a cómo utilizarla. Respecto al diagrama de clases, se logró hacer una representación clara de la estructura del programa, facilitando su comprensión y posible mantenimiento. Por otro lado, las pruebas unitarias han permitido validar las funciones más importantes del proyecto, garantizando su óptimo funcionamiento.