

Oswaldo Cabrera Pérez  
Práctica Final

1. Breve reporte en una cuartilla que explique las decisiones de diseño tomadas para la implementación de los repositorios y las consultas.

Para realizar esta práctica final he decidido tomar las clases Alumno, Cuenta y Transacción las cuales son tres de las más significativas en mi proyecto.

Las tres clases extenderán a la interfaz CRUD repository ya que es necesario que las tres tengan al menos las cuatro operaciones básicas del CRUD: create (save), read (find), delete y update (save). Las tres son candidatas a tener namedQueries.

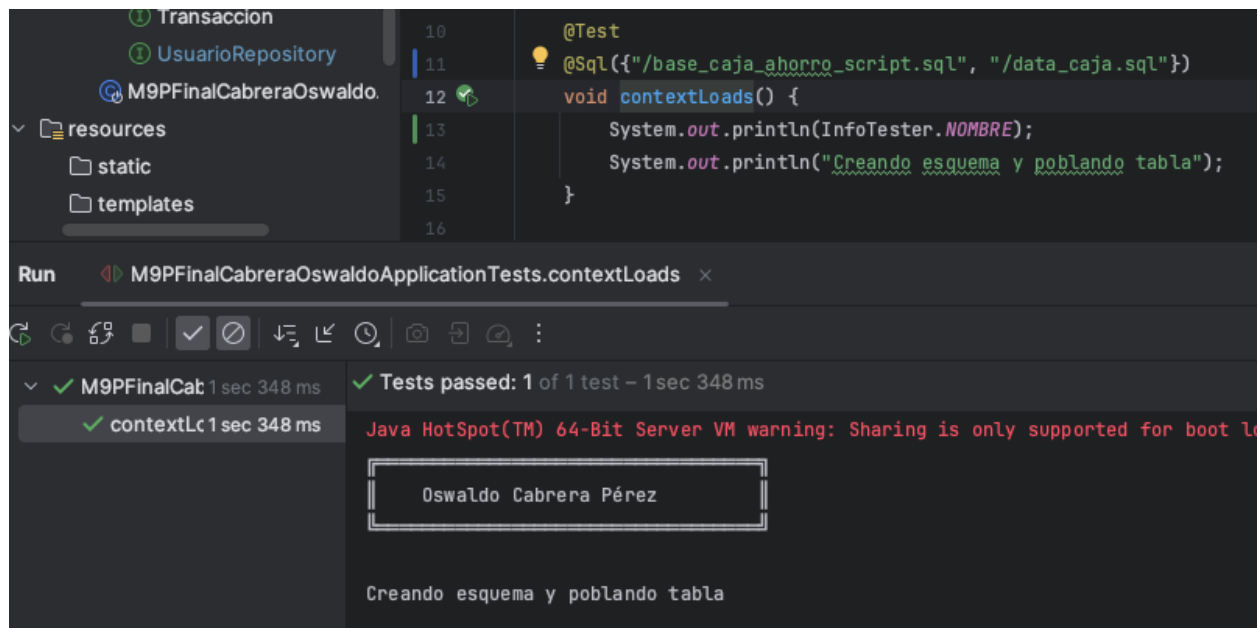
He decidido que la entidad Transacción es la única que implementará a la interfaz PagingAndSortingRepository ya que se necesita registrar todas las transacciones que se generen en el sistema, entonces se espera que sea una tabla que crezca mucho respecto a las demás. Si nos trajéramos todos los datos al mismo tiempo significaría una gran carga de trabajo para nuestra base de datos ocasionando problemas de rendimiento.

Es por lo anterior que Transacción es una gran candidata para utilizar la interfaz PagingAndSortingRepository para así poderme traer los registros necesarios y en el orden que necesite. Otra opción para mejorar el rendimiento serían ocupar queries nativas pero debido a que sería más complejo implementar el paginado no opté por este camino.

Evidencias:

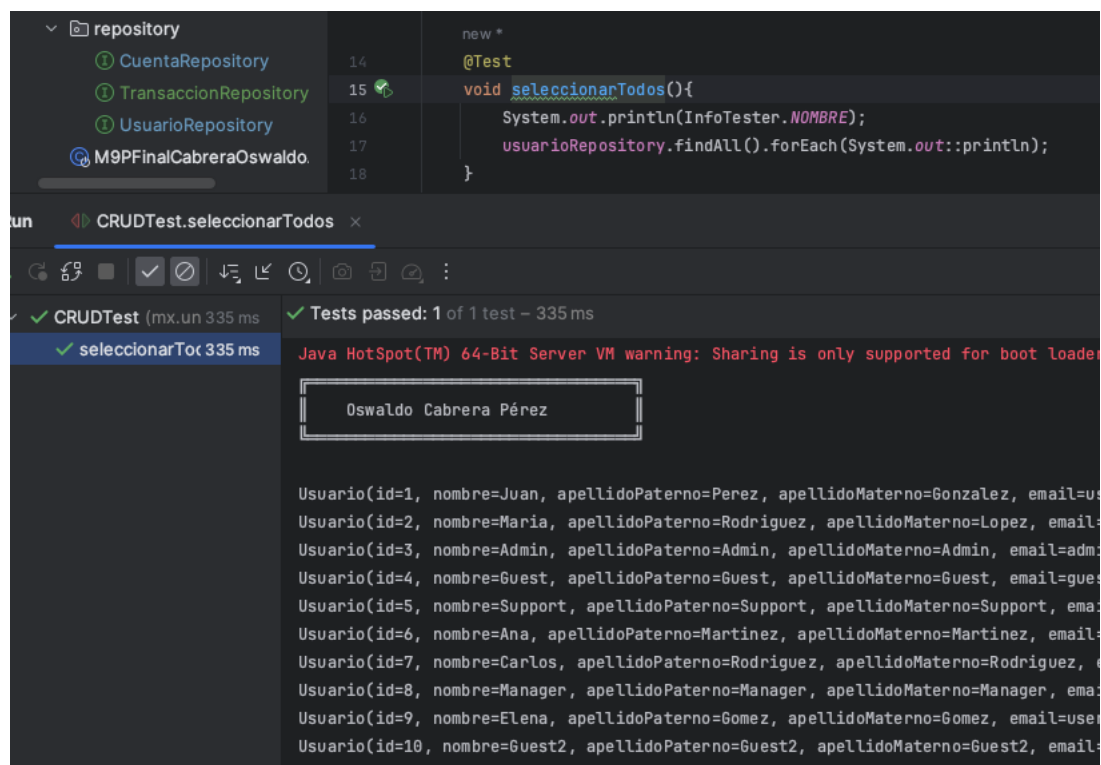
1. Una clase con una prueba unitaria donde implemente la creación del esquema y la inserción de registros de prueba en la base de datos.

1. Debe incluir SQL para la creación de la base de datos, con un nombre específico para tu proyecto.
2. SQL para seleccionar la base de datos.
3. SQL para la creación e inserción de datos de prueba, al menos 10 registros por tabla.



- Una clase de pruebas unitarias donde implemente operaciones(CRUD) básicas de CrudRepository de una tabla, una operación en un método diferente de prueba @Test. Se espera al menos un método de prueba por operación CRUD, es decir 4 métodos de prueba mínimo.

findAll()



save(Entidad entidad)

The screenshot displays an IDE interface with the following components:

- Project Explorer (Left):** Shows a project structure with folders `repository`, `resources`, and `test`. The `test` folder contains a `java` sub-folder, which in turn contains a package `mx.unam.dgtic.M9_PFinal_Ca` with a class `CRUDTest`.
- Code Editor (Center):** Displays the `guardarUsuario()` method in `CRUDTest.java`. The code is as follows:

```
20  new *  
21  @Test  
22  void guardarUsuario(){  
23      System.out.println(InfoTester.NOMBRE);  
24      Usuario usuario = new Usuario();  
25      usuario.setNombre("Oswaldo");  
26      usuario.setApellidoPaterno("Cabrera");  
27      usuario.setApellidoMaterno("Garcia");  
28      usuario.setEmail("oswaldo@hola.com");  
29      usuario.setContrasena("1234");  
30      usuario.setRol_id(1);  
31      usuario.setNumeroTelefonico("5512345678");  
32      usuario.setZip(12345);  
33      usuario.setEstado("CDMX");  
34      usuario.setCalle("Empedrado");  
35      usuario.setContrasena("1234");  
36      usuarioRepository.save(usuario);  
    }
```
- Run Console (Bottom):** Shows the execution of the `guardarUsuario()` test. The status is "Tests passed: 1 of 1 test - 375 ms". A warning message is visible: "Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only sup". The output of the test is displayed in a box:

```
Oswaldo Cabrera Pérez
```

save(Entidad entidad) para actualizar

The screenshot shows an IDE with a project structure on the left: `java` > `mx.unam.dgtic.M9_PFinal_Ca` > `model` (containing `Cuenta`, `Transaccion`, `Usuario`) > `repository`. The main editor displays the following Java code for the `actualizarUsuario()` method:

```
39  @Test
40  void actualizarUsuario(){
41      System.out.println(InfoTester.NOMBRE);
42      System.out.println(usuarioRepository.findById(1));
43      Usuario user = usuarioRepository.findById(1);
44      user.setNombre("Oswaldo");
45      usuarioRepository.save(user);
46      System.out.println(usuarioRepository.findById(1));
47  }
```

The Run window at the bottom shows the test `CRUDTest.actualizarUsuario` passed successfully in 398 ms. The output console displays the following text:

```
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot
Oswaldo Cabrera Pérez
Usuario(id=1, nombre=Juan, apellidoPaterno=Perez, apellidoMaterno=Gonzalez, em
Usuario(id=1, nombre=Oswaldo, apellidoPaterno=Perez, apellidoMaterno=Gonzalez,
```

delete( Entidad entidad)

The screenshot shows an IDE with a project structure on the left: `application.properties`, `base_caja_ahorro_script.sql`, `data_caja.sql`, `test` > `java` > `mx.unam.dgtic.M9_PFinal_Cabrera_Oswaldo`. The main editor displays the following Java code for the `eliminarUsuario()` method:

```
50  @Test
51  void eliminarUsuario(){
52      System.out.println(InfoTester.NOMBRE);
53      System.out.println(usuarioRepository.findById(1));
54      Usuario usuario = usuarioRepository.findById(1);
55      usuarioRepository.delete(usuario);
56  }
```

The Run window at the bottom shows the test `CRUDTest.eliminarUsuario` passed successfully in 342 ms. The output console displays the following text:

```
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader class
Oswaldo Cabrera Pérez
Usuario(id=1, nombre=Oswaldo, apellidoPaterno=Perez, apellidoMaterno=Gonzalez, email=user1
```

3. Una clase de prueba donde implemente 4 Consultas derivadas de una tabla diferente al punto1.

- findSaldoGreaterThan

The screenshot shows an IDE with a project named 'mx.unam.dgtic.M9\_PFinal\_Cabrera\_Osw'. The file explorer on the left shows a package 'ConsultasDerivadasTest'. The main editor displays the following Java code:

```
@Test
void saldoMayorTest(){
    System.out.println(InfoTester.NOMBRE);
    cuentaRepository.findBySaldoGreaterThan(1000).forEach(System.out::println);
}
```

The Run window shows the test 'saldoMayorTest()' passed in 332 ms. The output console displays the name 'Oswaldo Cabrera Pérez' and a list of account records:

```
Cuenta(id=3, saldo=2000, fecha_apertura=2023-11-15 08:15:00.0, usuario_id=Usuario(id=3, nombre=Admin, apelli
Cuenta(id=6, saldo=1200, fecha_apertura=2023-11-18 11:30:00.0, usuario_id=Usuario(id=3, nombre=Admin, apelli
Cuenta(id=8, saldo=2500, fecha_apertura=2023-11-20 10:00:00.0, usuario_id=Usuario(id=4, nombre=Guest, apelli
Cuenta(id=10, saldo=1200, fecha_apertura=2023-11-22 08:45:00.0, usuario_id=Usuario(id=5, nombre=Support, ape
```

- findByFechaAperturaAfter

The screenshot shows the same IDE with a different test class. The file explorer on the left shows a package 'M9\_PFinalCabreraOsw'. The main editor displays the following Java code:

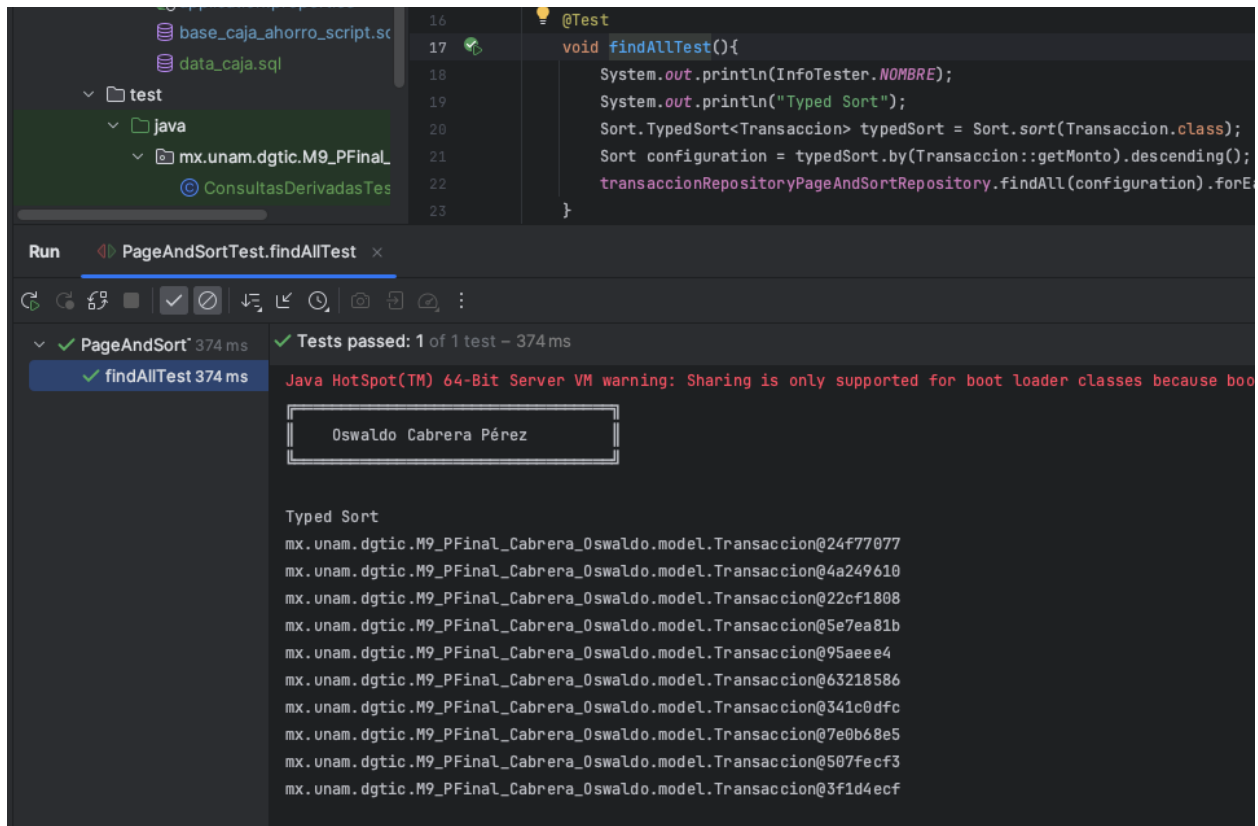
```
@Test
void fechaDespuesTest(){
    System.out.println(InfoTester.NOMBRE);
    cuentaRepository.findByFechaAperturaAfter(java.sql.Date.valueOf("2023-11-20")).forEach(System.out::println);
}
```

The Run window shows the test 'fechaDespuesTest()' passed in 386 ms. The output console displays the name 'Oswaldo Cabrera Pérez' and a list of account records:

```
Cuenta(id=8, saldo=2500, fechaApertura=2023-11-20 10:00:00.0, usuario_id=Usuario(id=4, nombre=Guest, apellidoPaterno=Guest, apelli
Cuenta(id=9, saldo=600, fechaApertura=2023-11-21 16:30:00.0, usuario_id=Usuario(id=2, nombre=Maria, apellidoPaterno=Rodriguez, apelli
Cuenta(id=10, saldo=1200, fechaApertura=2023-11-22 08:45:00.0, usuario_id=Usuario(id=5, nombre=Support, apellidoPaterno=Support, ape
```

- findBySaldoGreaterThanAndFechaAperturaAfterTest

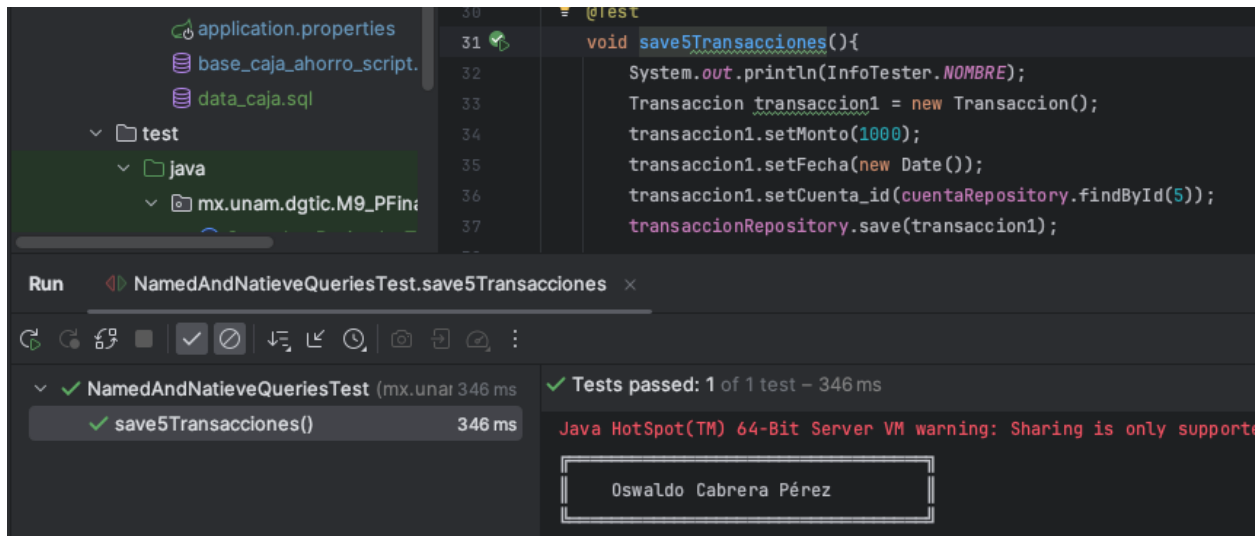




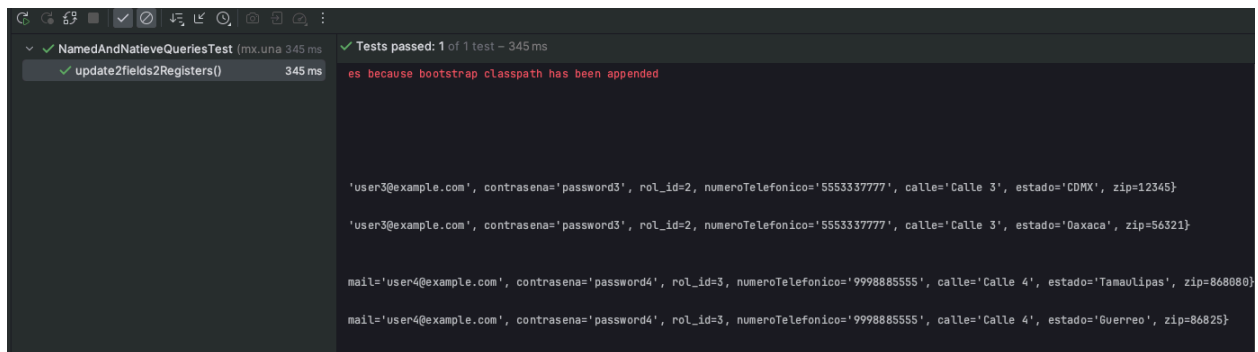
Segunda Prueba:

Crear al menos 5 registros en una tabla





## II. Modificar 2 registros en 2 campos.

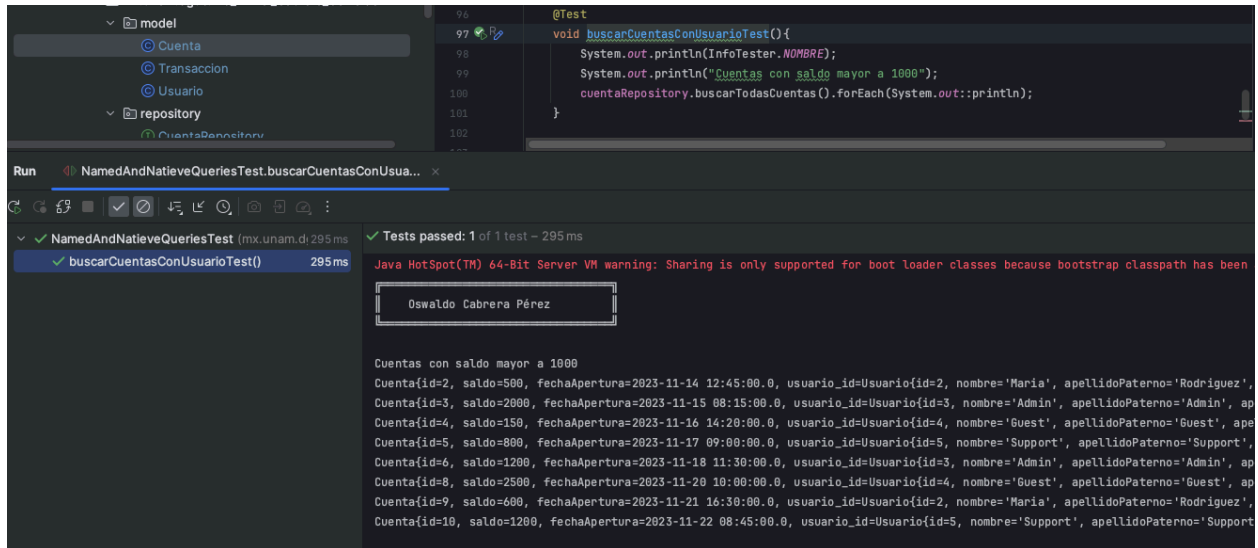


## III. Eliminar 1 registro.



- Al menos 3 Named Queries para consulta en dos tablas diferentes.

## Primera prueba



```
@Test
void buscarCuentasConUsuarioTest(){
    System.out.println(InfoTester.NOMBRE);
    System.out.println("Cuentas con saldo mayor a 1000");
    cuentaRepository.buscarTodasCuentas().forEach(System.out::println);
}
```

Run NamedAndNativeQueriesTest.buscarCuentasConUsua... x

✓ NamedAndNativeQueriesTest (mx.unam.d 295 ms)

✓ buscarCuentasConUsuarioTest() 295ms

Tests passed: 1 of 1 test - 295 ms

Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been

Oswaldo Cabrera Pérez

Cuentas con saldo mayor a 1000

Cuenta{id=2, saldo=500, fechaApertura=2023-11-14 12:45:00.0, usuario\_id=Usuario{id=2, nombre='Maria', apellidoPaterno='Rodriguez',

Cuenta{id=3, saldo=2000, fechaApertura=2023-11-15 08:15:00.0, usuario\_id=Usuario{id=3, nombre='Admin', apellidoPaterno='Admin', ap

Cuenta{id=4, saldo=150, fechaApertura=2023-11-16 14:20:00.0, usuario\_id=Usuario{id=4, nombre='Guest', apellidoPaterno='Guest', ape

Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5, nombre='Support', apellidoPaterno='Support',

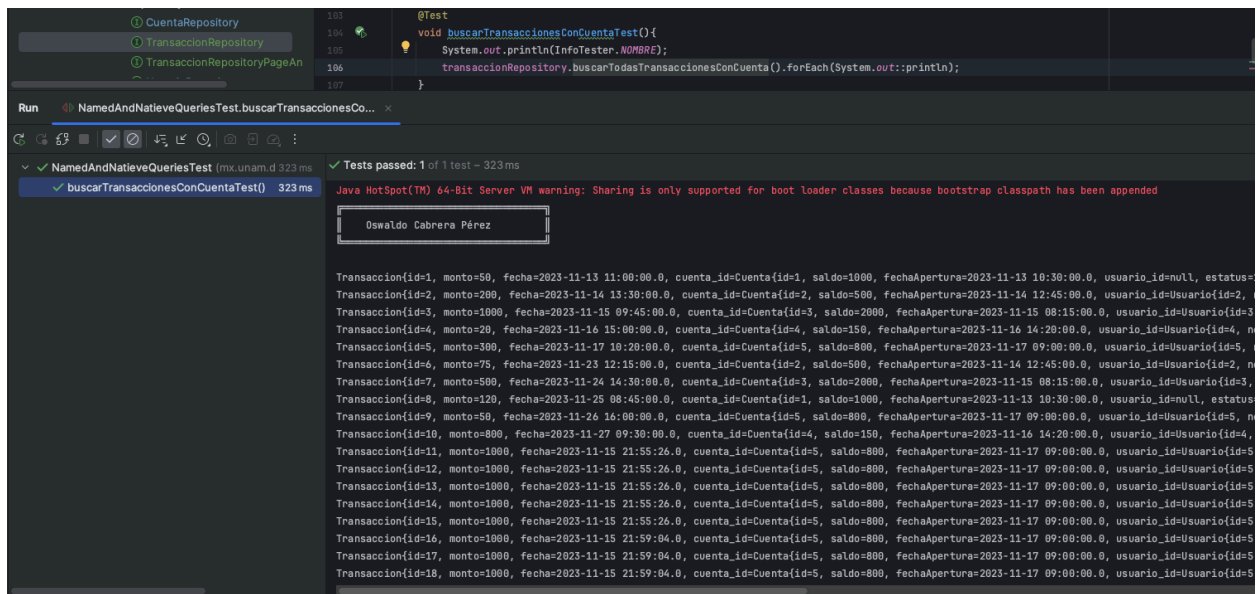
Cuenta{id=6, saldo=1200, fechaApertura=2023-11-18 11:30:00.0, usuario\_id=Usuario{id=3, nombre='Admin', apellidoPaterno='Admin', ap

Cuenta{id=8, saldo=2500, fechaApertura=2023-11-20 10:00:00.0, usuario\_id=Usuario{id=4, nombre='Guest', apellidoPaterno='Guest', ap

Cuenta{id=9, saldo=600, fechaApertura=2023-11-21 16:30:00.0, usuario\_id=Usuario{id=2, nombre='Maria', apellidoPaterno='Rodriguez',

Cuenta{id=10, saldo=1200, fechaApertura=2023-11-22 08:45:00.0, usuario\_id=Usuario{id=5, nombre='Support', apellidoPaterno='Support

## Segunda Prueba



```
@Test
void buscarTransaccionesConCuentaTest(){
    System.out.println(InfoTester.NOMBRE);
    transaccionRepository.buscarTodasTransaccionesConCuenta().forEach(System.out::println);
}
```

Run NamedAndNativeQueriesTest.buscarTransaccionesCo... x

✓ NamedAndNativeQueriesTest (mx.unam.d 323 ms)

✓ buscarTransaccionesConCuentaTest() 323ms

Tests passed: 1 of 1 test - 323 ms

Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended

Oswaldo Cabrera Pérez

Transaccion{id=1, monto=50, fecha=2023-11-13 11:00:00.0, cuenta\_id=Cuenta{id=1, saldo=1000, fechaApertura=2023-11-13 10:30:00.0, usuario\_id=null, estatus=

Transaccion{id=2, monto=200, fecha=2023-11-14 13:30:00.0, cuenta\_id=Cuenta{id=2, saldo=500, fechaApertura=2023-11-14 12:45:00.0, usuario\_id=Usuario{id=2,

Transaccion{id=3, monto=1000, fecha=2023-11-15 09:45:00.0, cuenta\_id=Cuenta{id=3, saldo=2000, fechaApertura=2023-11-15 08:15:00.0, usuario\_id=Usuario{id=3

Transaccion{id=4, monto=20, fecha=2023-11-16 15:00:00.0, cuenta\_id=Cuenta{id=4, saldo=150, fechaApertura=2023-11-16 14:20:00.0, usuario\_id=Usuario{id=4, n

Transaccion{id=5, monto=300, fecha=2023-11-17 10:20:00.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5,

Transaccion{id=6, monto=75, fecha=2023-11-23 12:15:00.0, cuenta\_id=Cuenta{id=2, saldo=500, fechaApertura=2023-11-14 12:45:00.0, usuario\_id=Usuario{id=2, n

Transaccion{id=7, monto=500, fecha=2023-11-24 14:30:00.0, cuenta\_id=Cuenta{id=3, saldo=2000, fechaApertura=2023-11-15 08:15:00.0, usuario\_id=Usuario{id=3,

Transaccion{id=8, monto=120, fecha=2023-11-25 08:45:00.0, cuenta\_id=Cuenta{id=1, saldo=1000, fechaApertura=2023-11-13 10:30:00.0, usuario\_id=null, estatus

Transaccion{id=9, monto=50, fecha=2023-11-26 16:00:00.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5, n

Transaccion{id=10, monto=800, fecha=2023-11-27 09:30:00.0, cuenta\_id=Cuenta{id=4, saldo=150, fechaApertura=2023-11-16 14:20:00.0, usuario\_id=Usuario{id=4,

Transaccion{id=11, monto=1000, fecha=2023-11-15 21:55:26.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5

Transaccion{id=12, monto=1000, fecha=2023-11-15 21:55:26.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5

Transaccion{id=13, monto=1000, fecha=2023-11-15 21:55:26.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5

Transaccion{id=14, monto=1000, fecha=2023-11-15 21:55:26.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5

Transaccion{id=15, monto=1000, fecha=2023-11-15 21:55:26.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5

Transaccion{id=16, monto=1000, fecha=2023-11-15 21:59:04.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5

Transaccion{id=17, monto=1000, fecha=2023-11-15 21:59:04.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5

Transaccion{id=18, monto=1000, fecha=2023-11-15 21:59:04.0, cuenta\_id=Cuenta{id=5, saldo=800, fechaApertura=2023-11-17 09:00:00.0, usuario\_id=Usuario{id=5

c. AL menos 3 Native Named Queries para consulta en dos tablas.

## Primera prueba:

```
@Test
void buscarTodosUsuariosConCuentasTest(){
    System.out.println(InfoTester.NOMBRE);
    List<Object[]> resultados = usuarioRepository.buscarTodosUsuariosConCuentas();
    for (Object[] resultado : resultados) {
        if(resultado[0] instanceof Usuario){
            System.out.println((Usuario)resultado[0]);
        }else{
            System.out.println((Cuenta)resultado[0]);
        }
    }
}
```

NamedAndNativeQueriesTest.buscarTodosUsuariosC... x

NamedAndNativeQueriesTest (mx.unam.dg) 285 ms

✓ Tests passed: 1 of 1 test - 285 ms

Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader c

Oswaldo Cabrera Pérez

Usuario{id=2, nombre='Maria', apellidoPaterno='Rodriguez', apellidoMaterno='Lopez', em  
Usuario{id=3, nombre='Admin', apellidoPaterno='Admin', apellidoMaterno='Admin', email=  
Usuario{id=4, nombre='Guest', apellidoPaterno='Guest', apellidoMaterno='Guest', email=  
Usuario{id=5, nombre='Support', apellidoPaterno='Support', apellidoMaterno='Support',  
Usuario{id=3, nombre='Admin', apellidoPaterno='Admin', apellidoMaterno='Admin', email=  
Usuario{id=4, nombre='Guest', apellidoPaterno='Guest', apellidoMaterno='Guest', email=  
Usuario{id=2, nombre='Maria', apellidoPaterno='Rodriguez', apellidoMaterno='Lopez', em  
Usuario{id=5, nombre='Support', apellidoPaterno='Support', apellidoMaterno='Support',

## Segunda prueba:

java

- mx.unam.dgtic.M9\_PFinal\_Cabrera\_C
  - model
    - Cuenta
    - Transaccion
    - Usuario
  - repository

```
@Test
void buscarCuentaYTransaccionTest(){
    System.out.println(InfoTester.NOMBRE);
    List<Cuenta> resultados = cuentaRepository.buscarCuentaYTransacciones(id: 4);
    resultados.forEach(System.out::println);
}
```

Run NamedAndNativeQueriesTest.buscarCuentaYTransacc... x

NamedAndNativeQueriesTest (mx.unam.dg) 292 ms

✓ Tests passed: 1 of 1 test - 292 ms

Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpa

Oswaldo Cabrera Pérez

Cuenta{id=4, saldo=150, fechaApertura=2023-11-16 14:20:00.0, usuario\_id=Usuario{id=4, nombre='Guest', apellidoPaterno=  
Cuenta{id=4, saldo=150, fechaApertura=2023-11-16 14:20:00.0, usuario\_id=Usuario{id=4, nombre='Guest', apellidoPaterno=

## Tercera Prueba

