Mejoramiento de la Aplicación BILBO

Docente: Patrick Cuadros Quiroga

Curso: Calidad y Pruebas de Software

Código de curso: SI-784

Facultad: Facultad de Ingeniería

Escuela: Escuela Profesional de Ingeniería de Sistemas (EPIS)

Integrantes:

- Medina López, Marcelo José - 2020066917
- Poma Manchego, Rene Manuel - 2017057491
- Zevallos Purca, Justin Zinedine

TACNA – PERÚ

2023 - I

Resumen

El presente proyecto radica en la implementación de un sistema web llamado “Bilbo” para la empresa con el mismo nombre, enfocado a la venta de libros, ante la necesidad de la empresa de llevar a cabo la venta y compra de libros, luego registrando los procesos de aquellas acciones en una base de datos. Cabe aclarar que este proyecto se encuentra con codificación ya desarrollada casi en su totalidad; sin embargo, el objetivo del proyecto se centra en la mejora de su desarrollo y desempeño, lo cual incluirá procedimientos que aseguran la calidad y control del software, tales como revisión de código, testeo del código, identificación del mayor número de errores posibles, el menor número de revisiones posibles evitando costos, y el aseguramiento de no cometer errores externos, implicando que no se efectúe la satisfacción del cliente.

Abstract

This project is based on the implementation of a web system called "Bilbo" for the company with the same name, focused on the sale of books, given the company's need to carry out the sale and purchase of books, then recording the processes of those actions in a database. It should be clarified that this project is with coding already developed almost in its entirety; however, the objective of the project is focused on improving its development and performance, which will include procedures that ensure the quality and control of the software, such as code review, code testing, identification of the largest possible number of errors, the least possible number of revisions avoiding costs, and ensuring that no external errors are made, implying that customer satisfaction is not realized.

1. Antecedentes o introducción

- Se tiene en cuenta que, aparte de nuestro proyecto, existen más sistemas de bibliotecas, con posibles funciones parecidas, los cuales pueden llegar a servir de inspiración, pero una gran diferencia entre aquellos sistemas y el nuestro es que se puede vender libros en lugar de prestarlos, cambiando un poco esta tradición por así decirlo. Con el presente proyecto se pretende cubrir un área principal de una biblioteca, dónde se desea implementar un sistema el cual agilizará el proceso de la biblioteca, registrando los datos necesarios y realizando las consultas que el usuario desee. El proyecto consiste en buscar optimizar el servicio de una biblioteca, así mismo mantener un orden al momento de realizar el CRUD. Para ello usaremos el Design Thinking, dónde podremos: Empatizar, Definir, Idear, Prototipar y Testear.

2. Título

- El título del presente proyecto es “Mejoramiento de la Aplicación BILBO”.

3. Autores

- Marcelo José Medina López.
- Rene Poma Manchebo.
- Justin Zevallos Purca.

4. Planteamiento del problema

4.1. Problema

- La baja revisión de la codificación del sistema web “Bilbo”.

4.2. Justificación

- La mejora de la experiencia del usuario, si los usuarios no están satisfechos con la biblioteca web, es importante mejorarla para que su experiencia sea más agradable y fácil de usar. Al mejorar la interfaz de usuario y la navegación, se puede hacer que la biblioteca sea más intuitiva para el usuario, además, permitiría la llegada de nuevos clientes interesados en el sistema web.

4.3. Alcance

- El alcance de nuestro será dentro de la empresa y toda Tacna, haciendo que las personas noten la mejoría de la empresa, atrayéndolas como nuevos clientes.

5. Objetivos

5.1. General:

- La implementación y revisión de un sistema capaz de poner al día los procesos o funcionalidades de la biblioteca, mejorando significativamente la atención al cliente.

5.2. Específicos:

- Clasificar los libros por el catálogo que posean.
- Vender los libros de forma eficiente y ágil.
- Ofrecer un espacio agradable para que cada uno se centre solo en la lectura y también para congeniar con los demás, compartiendo opiniones.

6. Referentes teóricos

Diagramas de Casos de Uso:

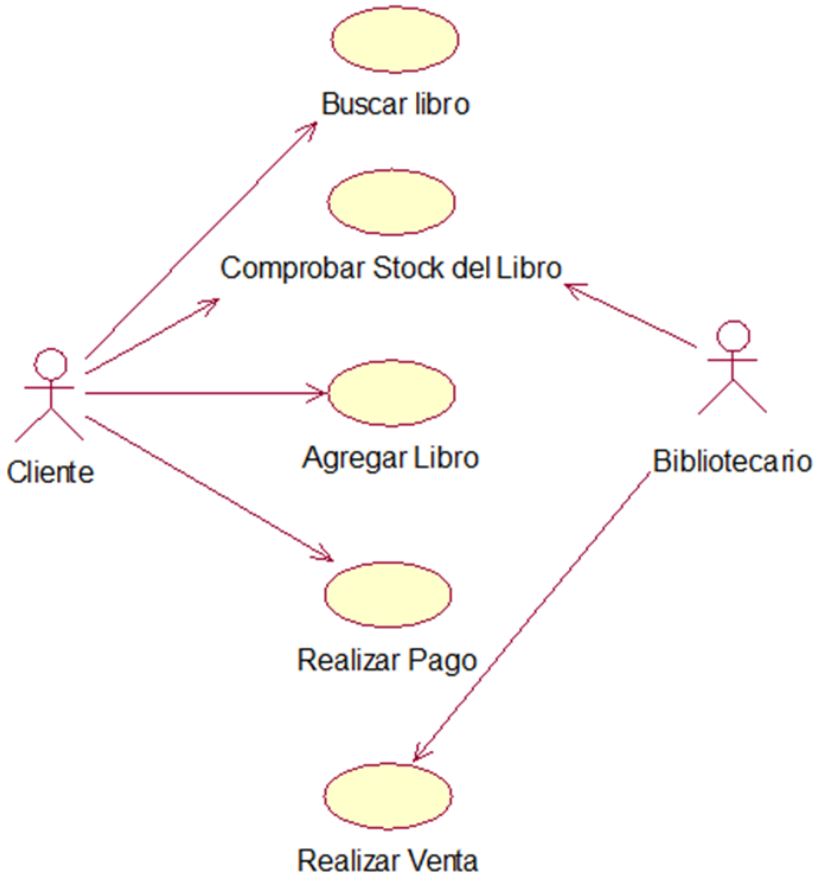


Diagrama de Clases:

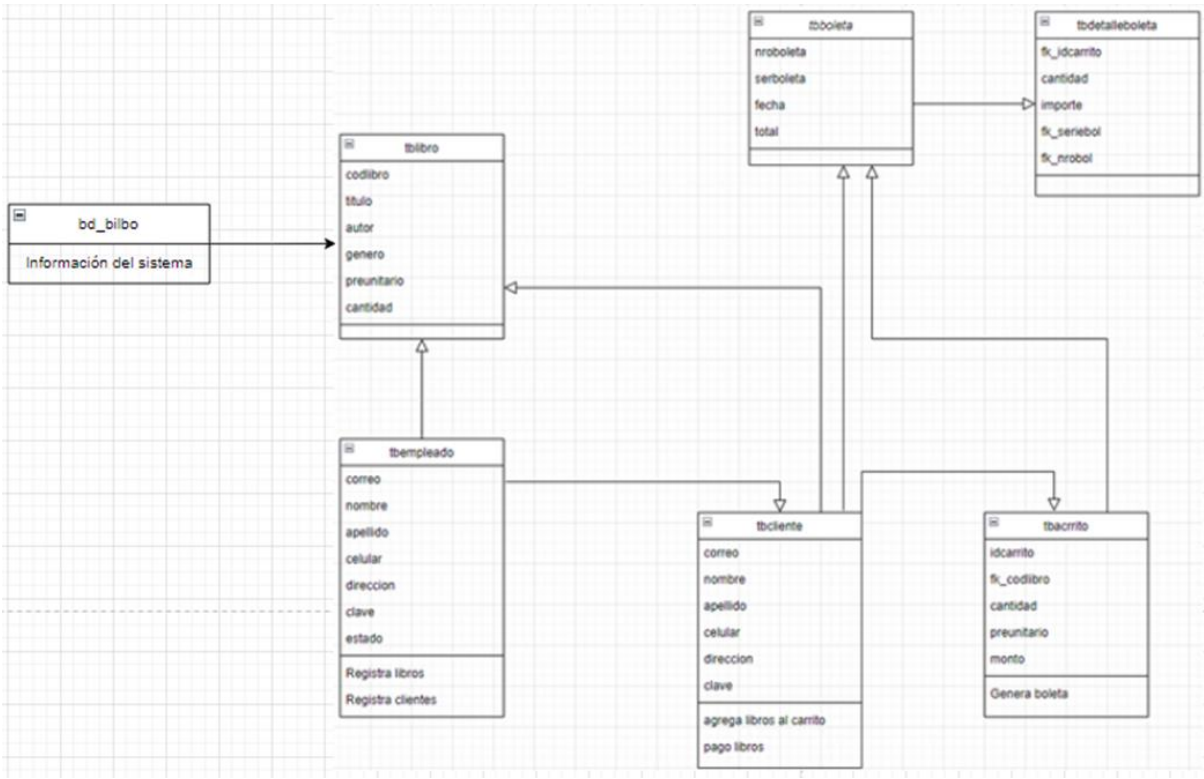
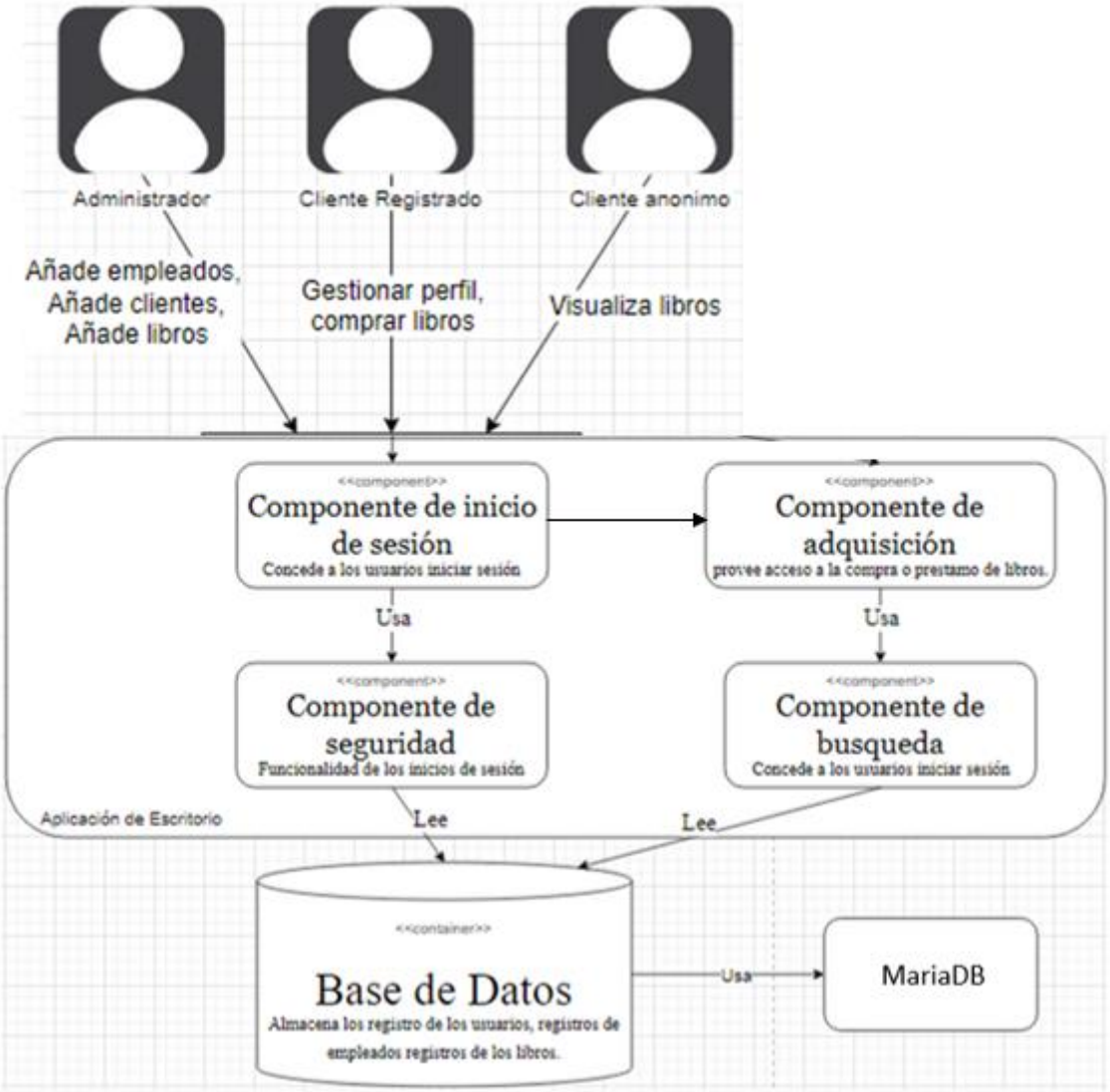


Diagrama de Componentes:



Arquitectura:

Diagrama Contextual:

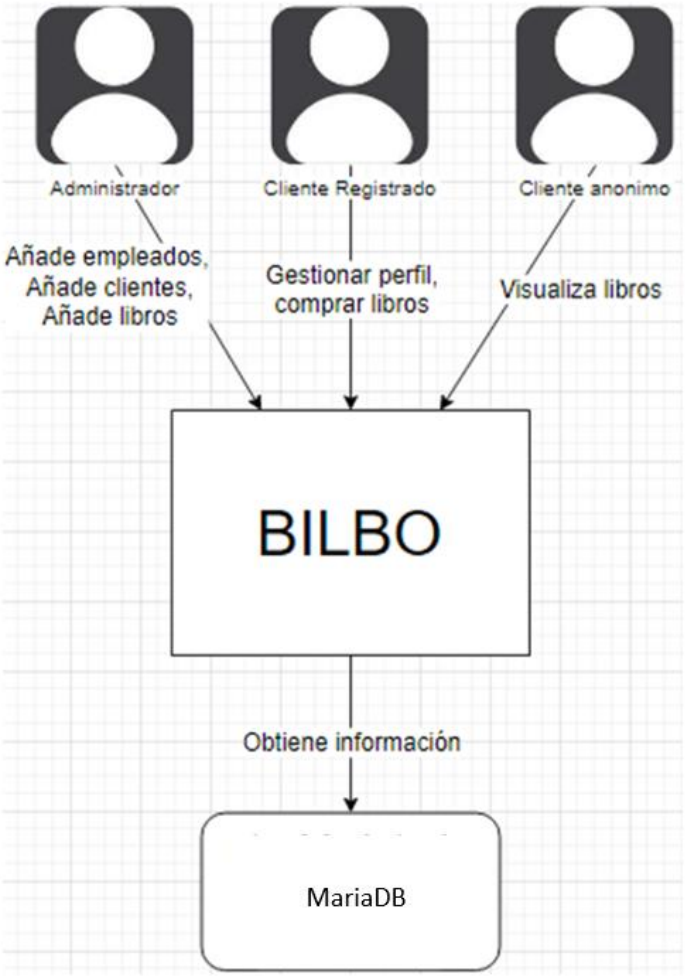
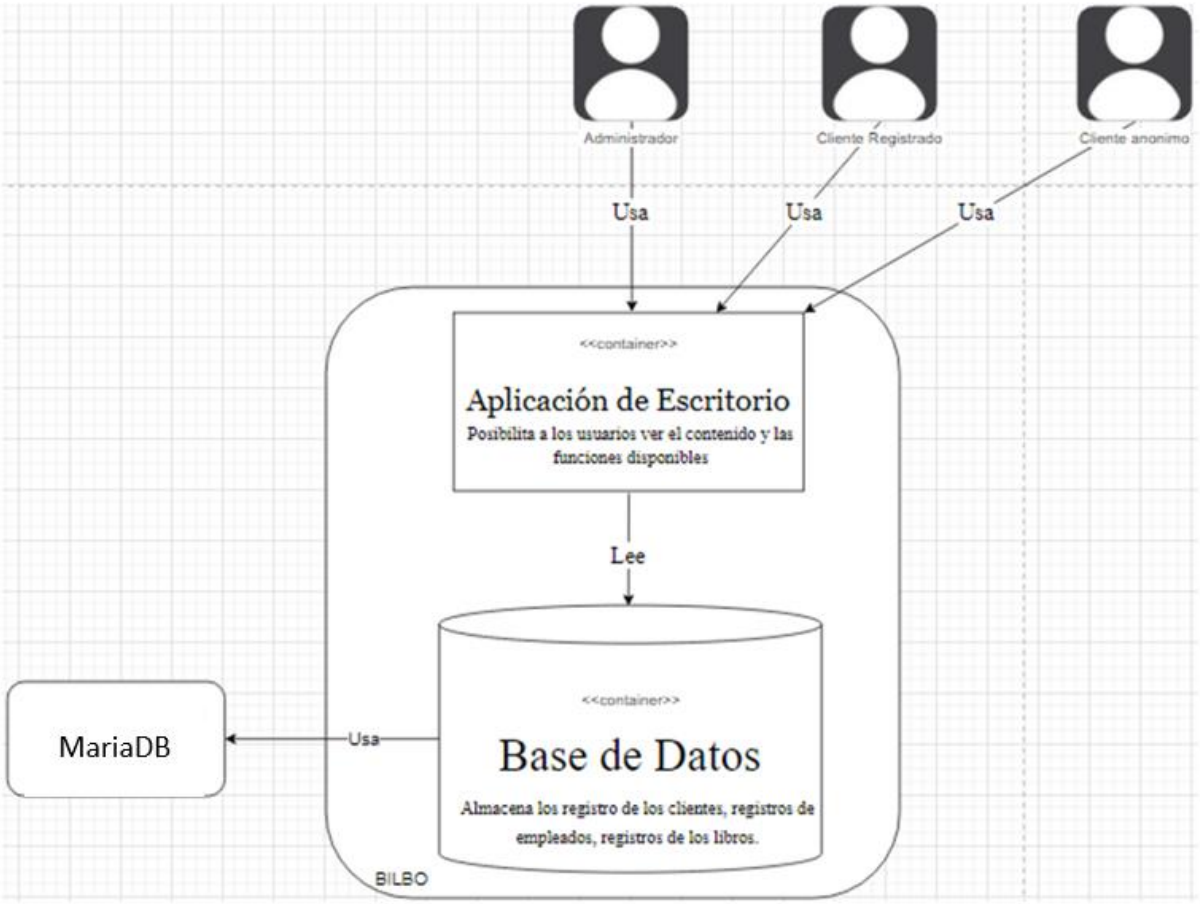


Diagrama de Contenedores:



7. Desarrollo de la propuesta

7.1. Tecnología de información:

- SonarQube: se utilizó para analizar y gestionar la calidad del código fuente del proyecto de software. Se trata de una herramienta que permite detectar problemas en el código, como errores, vulnerabilidades de seguridad, complejidad excesiva, falta de comentarios y otras cuestiones que afectan la calidad y mantenibilidad del software.

7.2. Metodología, técnicas usadas

- Análisis estático de código: proceso automatizado que examina el código fuente sin necesidad de ejecutar el programa. En SonarQube, se utilizan diferentes herramientas y motores de análisis estático para identificar una amplia variedad de problemas en el código, como vulnerabilidades de seguridad, errores de sintaxis, problemas de estilo, complejidad excesiva y otros.

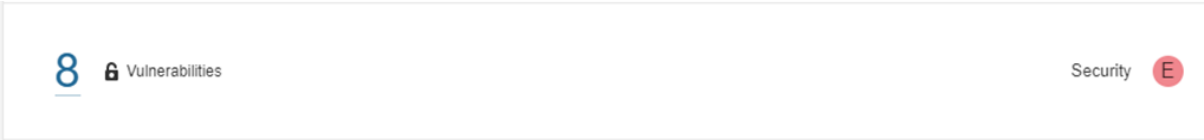
- Pruebas unitarias: pruebas automatizadas que se ejecutan sobre pequeñas secciones de código para verificar su correcto funcionamiento. Estas pruebas permiten detectar errores y problemas en el código, y pueden integrarse con SonarQube para proporcionar información adicional sobre la calidad del software.

8. Cronograma basado en las observaciones que la herramienta SonarQube les informará sobre la aplicación, a fin de reducir la deuda técnica, vulnerabilidades, fallas, etc. a 0.

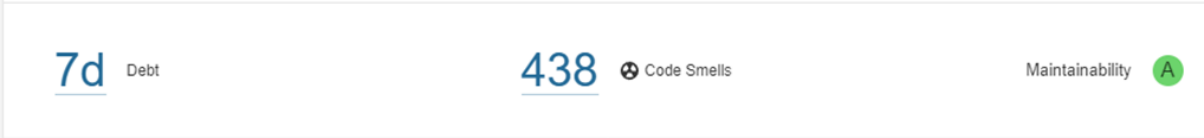
- Las observaciones que nos arrojó el sonarQube son las siguientes:

- 1 bug por lo que nos califica con una “B” en fiabilidad.

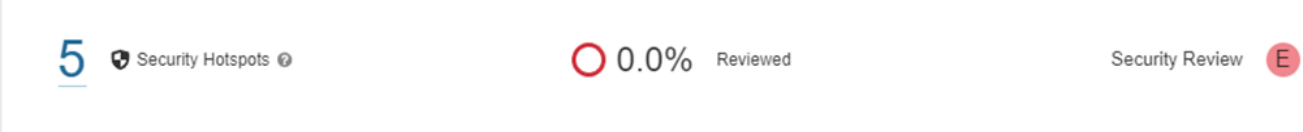
- Nuestro proyecto arrojó 8 vulnerabilidades dándonos una calificación de “E” en seguridad.



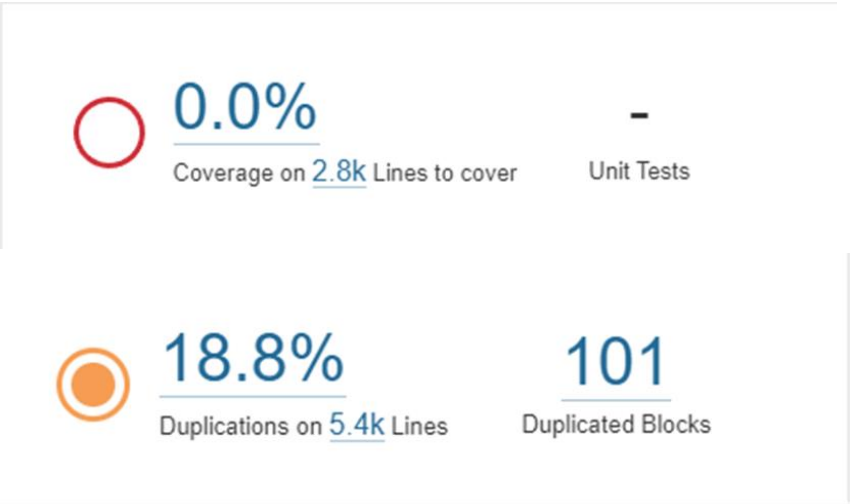
- Sonar nos calificó con 438 de deuda técnica y aproximadamente 7d de reparo, dándonos una calificación de “A” en mantenibilidad.

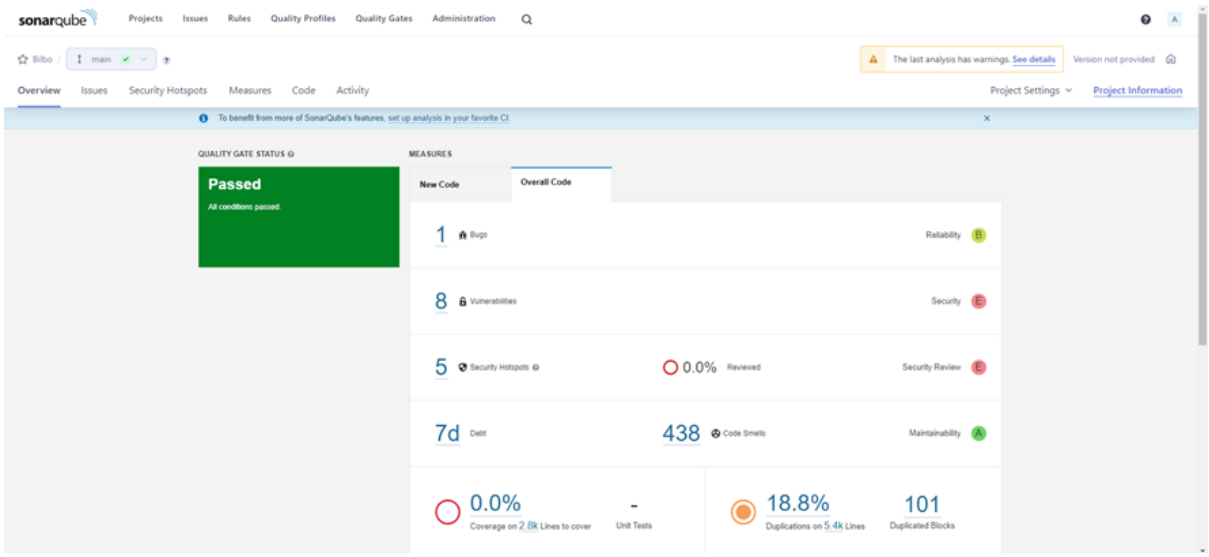


- En los puntos de seguridad tenemos 5 a corregir, sonar nos calificó con “E” en la reseña de seguridad.



- Sonar nos mostró el mensaje de cobertura en 2.8k líneas, calificando con un 0.0% y la duplicación en 5.4k líneas, calificando con un 18.8%.





Cronograma:

	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7	TOTAL
Vulnerabilidad	2	2			2	2		8 bugs corregidos
Hotspots			2	2			1	5 hotspot corregidos

9. Desarrollo de Solución de Mejora

9.1 Diagrama de Arquitectura de la aplicación
Diagrama de Componentes:

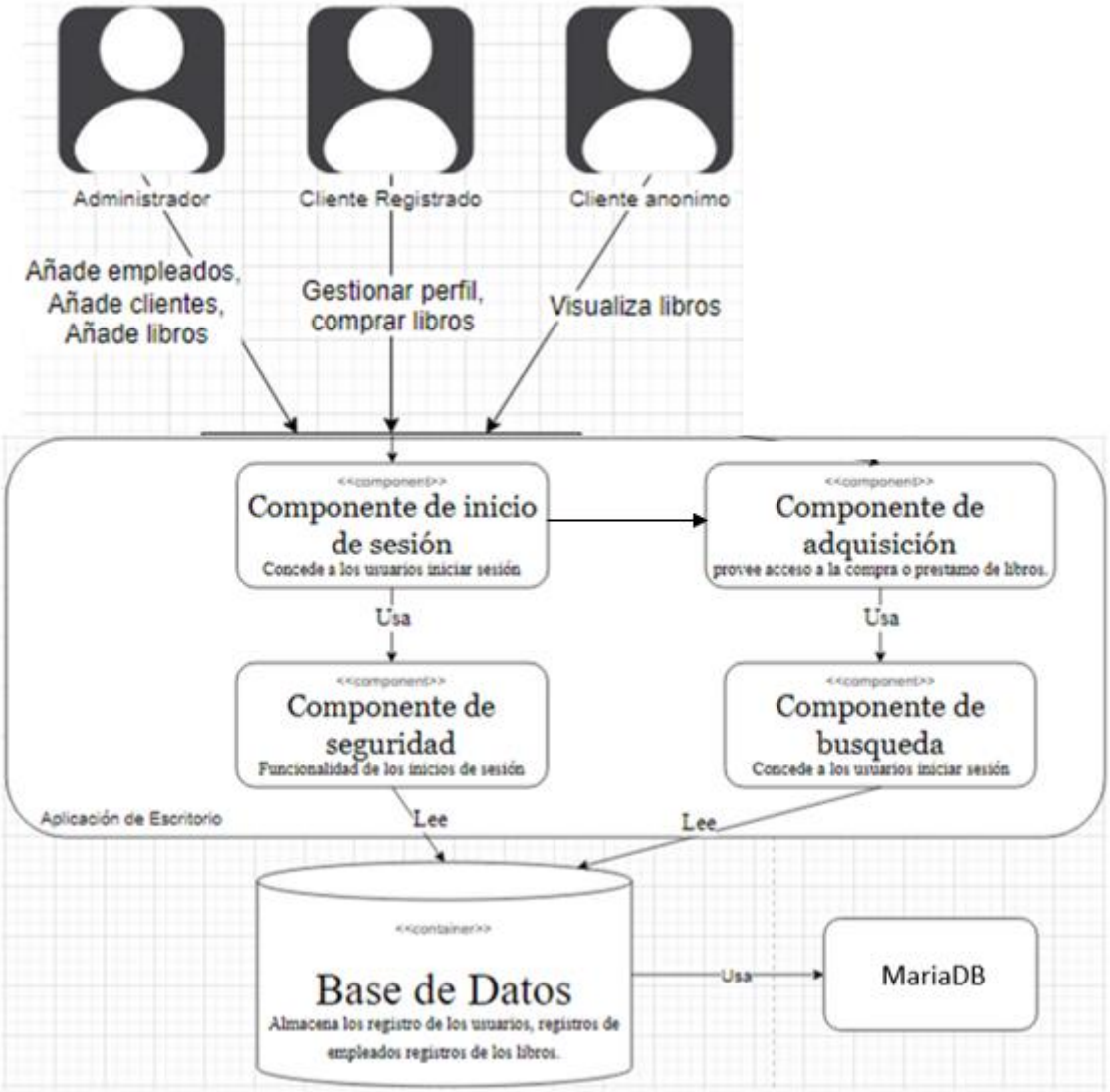


Diagrama Contextual:

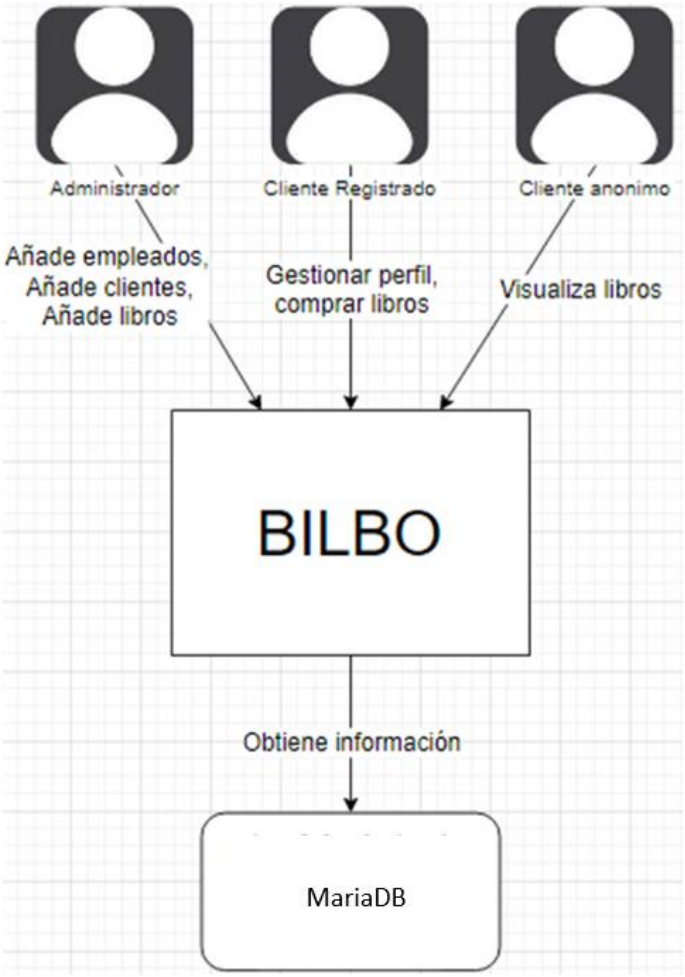
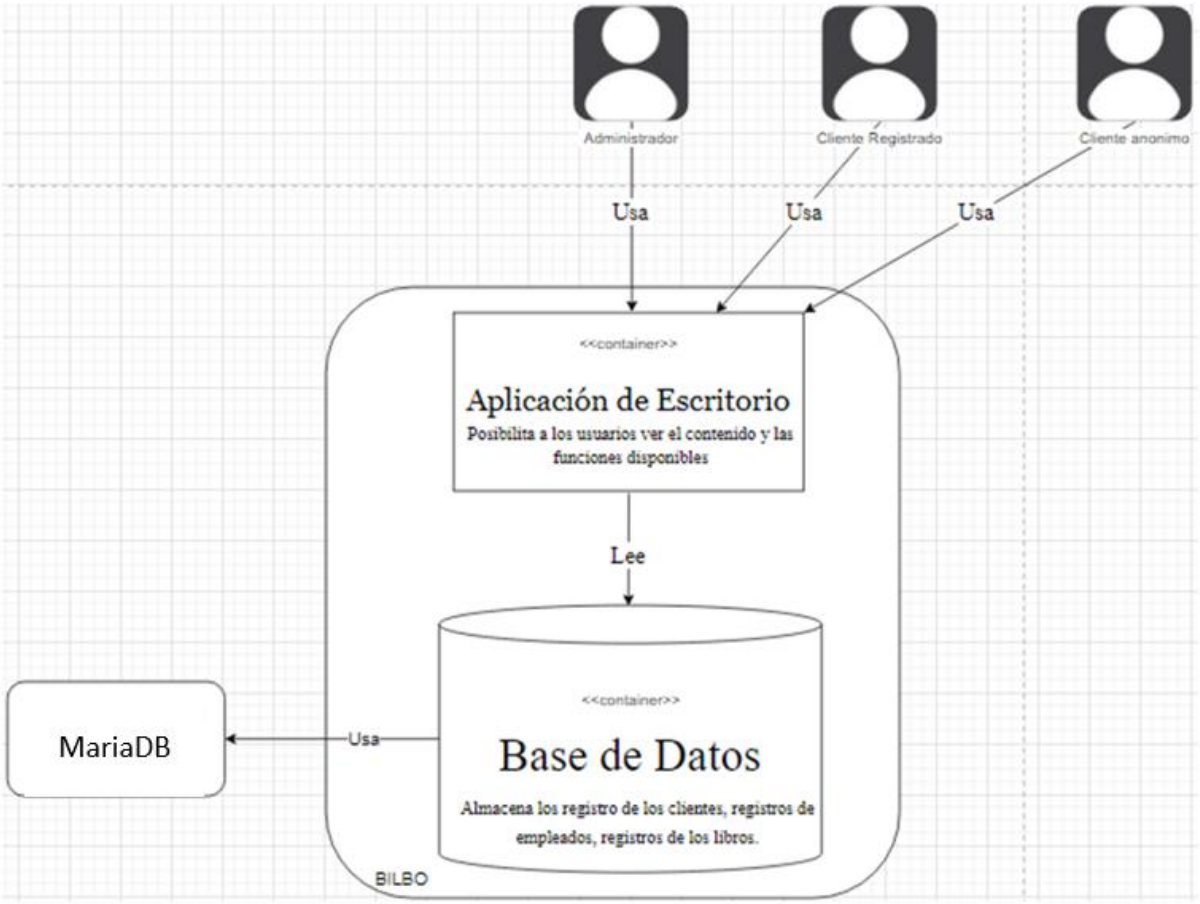
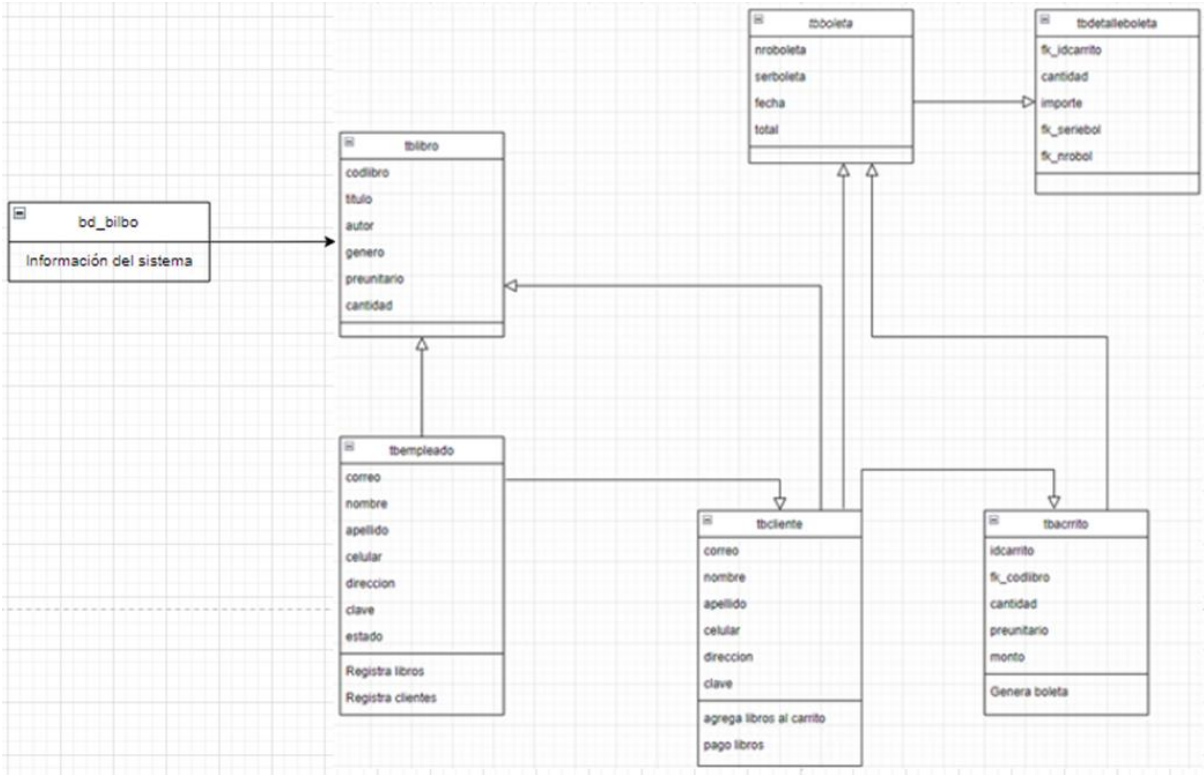


Diagrama de Contenedores:

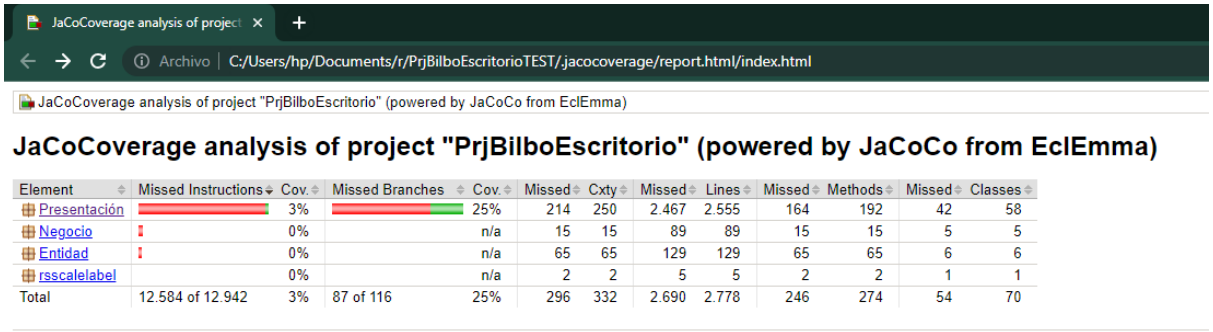


9.2 Diagrama de Clases de la aplicación



9.3 Métodos de pruebas implementados para coberturar la aplicación

- Reporte de cobertura de pruebas



JaCoCoverage analysis of project "PrjBilboEscritorio" (powered by JaCoCo from EclEmma)

JaCoCoverage analysis of project "PrjBilboEscritorio" (powered by JaCoCo from EclEmma)

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Presentación		3%		25%	214	250	2.467	2.555	164	192	42	58
Negocio		0%		n/a	15	15	89	89	15	15	5	5
Entidad		0%		n/a	65	65	129	129	65	65	6	6
rsscalelabel		0%		n/a	2	2	5	5	2	2	1	1
Total	12.584 of 12.942	3%	87 of 116	25%	296	332	2.690	2.778	246	274	54	70

Presentación

Archivo | C:/Users/hp/Documents/r/PrjBilboEscritorioTEST/.jacocoverage/report.html/Presentaci_n/index.html

JaCoCoCoverage analysis of project "PrjBilboEscritorio" (powered by JaCoCo from EclEmma) > Presentación

Presentación

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
FrmInicio		2%		21%	17	20	359	367	11	13	0	1
Frmempleado		3%		20%	19	23	363	376	10	13	0	1
Frmcliente		3%		20%	19	23	332	345	10	13	0	1
Frmlibro		3%		20%	19	23	339	352	10	13	0	1
Frmboleta		3%		29%	15	19	336	349	9	12	0	1
Frmcarrito		2%		19%	17	20	306	314	10	12	0	1
FrmMenu		4%		75%	10	12	163	170	9	10	0	1
Login		6%		50%	6	10	169	182	3	6	0	1
Frmlibro.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmlibro.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmlibro.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmlibro.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmlibro.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmInicio.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmlibro.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmInicio.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmInicio.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmInicio.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmInicio.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmInicio.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmempleado.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmempleado.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmempleado.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmempleado.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmempleado.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmempleado.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Login.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmMenu.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmMenu.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmMenu.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmMenu.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmMenu.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmMenu.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
FrmMenu.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcarrito.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcarrito.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcarrito.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcarrito.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcliente.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcarrito.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcliente.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcliente.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1
Frmcliente.new ActionListener() {...}		0%		n/a	2	2	3	3	2	2	1	1

Negocio

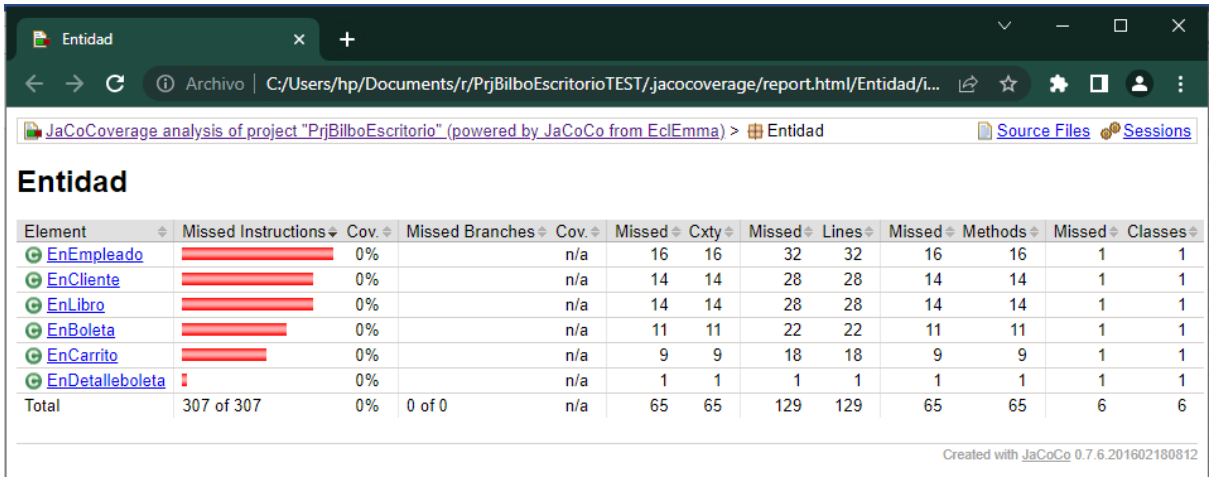
Archivo | C:/Users/hp/Documents/r/PrjBilboEscritorioTEST/.jacocoverage/report.html/Negocio/...

JaCoCoCoverage analysis of project "PrjBilboEscritorio" (powered by JaCoCo from EclEmma) > Negocio Source Files Sessions

Negocio

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
NeCliente		0%		n/a	5	5	35	35	5	5	1	1
NeEmpleado		0%		n/a	3	3	18	18	3	3	1	1
NeLibro		0%		n/a	3	3	18	18	3	3	1	1
NeCarrito		0%		n/a	2	2	10	10	2	2	1	1
Conexion		0%		n/a	2	2	8	8	2	2	1	1
Total	478 of 478	0%	0 of 0	n/a	15	15	89	89	15	15	5	5

Created with JaCoCo 0.7.6.201602180812



Entidad

Archivo | C:/Users/hp/Documents/r/PrjBilboEscritorioTEST/.jacocoverage/report.html/Entidad/i...

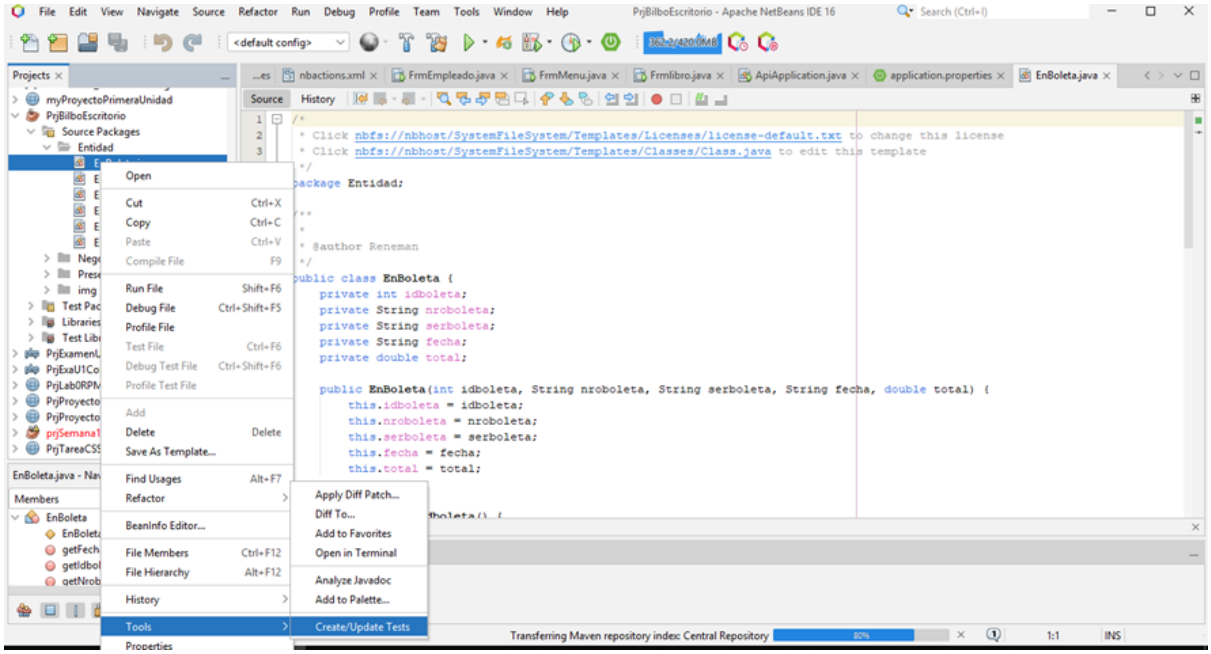
JaCoCoCoverage analysis of project "PrjBilboEscritorio" (powered by JaCoCo from EclEmma) > Entidad Source Files Sessions

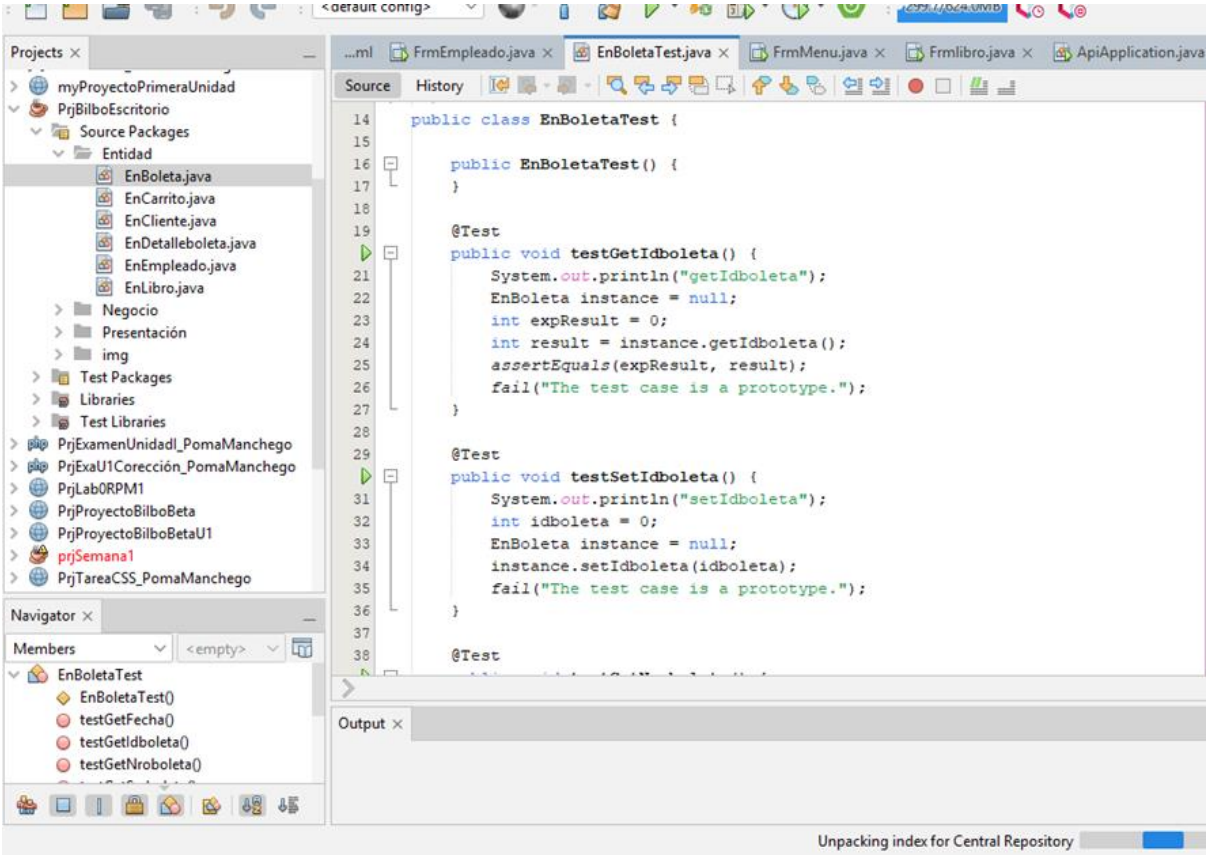
Entidad

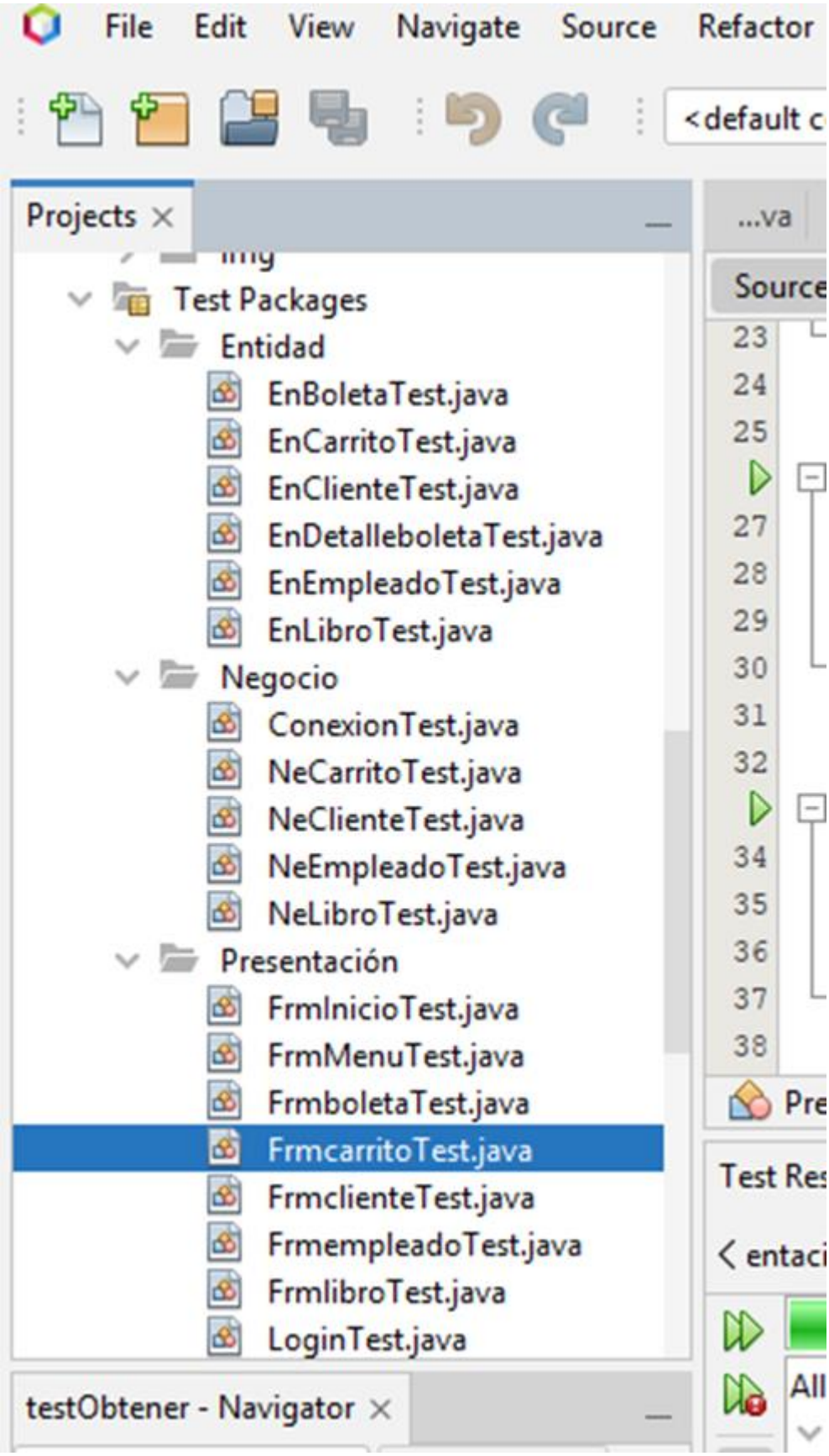
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
EnEmpleado		0%		n/a	16	16	32	32	16	16	1	1
EnCliente		0%		n/a	14	14	28	28	14	14	1	1
EnLibro		0%		n/a	14	14	28	28	14	14	1	1
EnBoleta		0%		n/a	11	11	22	22	11	11	1	1
EnCarrito		0%		n/a	9	9	18	18	9	9	1	1
EnDetalleboleta		0%		n/a	1	1	1	1	1	1	1	1
Total	307 of 307	0%	0 of 0	n/a	65	65	129	129	65	65	6	6

Created with JaCoCo 0.7.6.201602180812

a) Pruebas Unitarias (cobertura de al menos 70% de código - los métodos más importantes)




Projects
myProyectoPrimeraUnidad
PrjBilboEscritorio
Source Packages
Entidad
EnBoleta.java
EnCarrito.java
EnCliente.java
EnDetalleboleta.java
EnEmpleado.java
EnLibro.java
Negocio
Presentación
img
Test Packages
Libraries
Test Libraries
PrjExamenUnidadI_PomaManchego
PrjExaU1Corección_PomaManchego
PrjLab0RPM1
PrjProyectoBilboBeta
PrjProyectoBilboBetaU1
prjSemana1
PrjTareaCSS_PomaManchego
Navigator
Members
EnBoletaTest
EnBoletaTest()
testGetFecha()
testGetIdboleta()
testGetNroboleta()
FrmEmpleado.java
EnBoletaTest.java
FrmMenu.java
Frmlibro.java
ApiApplication.java
Source
History
public class EnBoletaTest {
public EnBoletaTest() {
}
@Test
public void testGetIdboleta() {
System.out.println("getIdboleta");
EnBoleta instance = null;
int expResult = 0;
int result = instance.getIdboleta();
assertEquals(expResult, result);
fail("The test case is a prototype.");
}
@Test
public void testSetIdboleta() {
System.out.println("setIdboleta");
int idboleta = 0;
EnBoleta instance = null;
instance.setIdboleta(idboleta);
fail("The test case is a prototype.");
}
@Test
Output
Unpacking index for Central Repository


File
Edit
View
Navigate
Source
Refactor
<default c
Projects
Test Packages
Entidad
EnBoletaTest.java
EnCarritoTest.java
EnClienteTest.java
EnDetalleboletaTest.java
EnEmpleadoTest.java
EnLibroTest.java
Negocio
ConexionTest.java
NeCarritoTest.java
NeClienteTest.java
NeEmpleadoTest.java
NeLibroTest.java
Presentación
FrmInicioTest.java
FrmMenuTest.java
FrmboletaTest.java
FrmcarritoTest.java
FrmclienteTest.java
FrmempleadoTest.java
FrmlibroTest.java
LoginTest.java
testObtener - Navigator
...va
Source
23
24
25
27
28
29
30
31
32
34
35
36
37
38
Test Res
All

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help PrjBilboEscritorio - Apache NetBeans IDE 16

Projects

- Frmcarrito.java
- Frmcliente.java
- Frmlibro.java
- Login.java
- img
- Test Packages
  - Entidad
  - Negocio
    - ConexionTest.java
    - NeCarritoTest.java
    - NeClienteTest.java
    - NeEmpleadoTest.java
    - NeLibroTest.java
  - Presentación
    - FrmInicioTest.java
    - FrmMenuTest.java
    - FrmboletaTest.java
    - FrmcarritoTest.java
    - FrmclienteTest.java
    - FrmempleadoTest.java
    - FrmlibroTest.java
    - LoginTest.java
- Libraries

FrmcarritoTest - Navigator

Members

- FrmcarritoTest
  - FrmcarritoTest()
  - testLimpiartabla()
  - testMain()
  - testObtener()

EnLibroTest.java | ConexionTest.java | FrmInicioTest.java | FrmMenuTest.java | FrmboletaTest.java

Source History

```
33      }
34
35      @Test
36      public void testMain() {
37          System.out.println("main");
38          String[] args = null;
39          Frmcarrito.main(args);
40      }
41
42  }
43
```

Presentación.FrmcarritoTest

Test Results | Output

Presentación.FrmInicioTest.testObtener | Presentación.FrmMenuTest | Presentación.FrmMenuTest.testMain

Tests passed: 66.67 %

2 tests passed, 1 test caused an error. (1.188 s)

- Presentación.FrmcarritoTest Failed
  - testObtener caused an ERROR: com.mysql.cj.jdbc.exceptions.CommunicationsException: Communicatio
  - testMain passed (0.227 s)
  - testLimpiartabla passed (0.422 s)

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help PrjBilboEscritorio - Apache NetBeans IDE 16

Projects

- Frmcarrito.java
- Frmcliente.java
- Frmlibro.java
- Login.java
- img
- Test Packages
  - Entidad
  - Negocio
    - ConexionTest.java
    - NeCarritoTest.java
    - NeClienteTest.java
    - NeEmpleadoTest.java
    - NeLibroTest.java
  - Presentación
    - FrmInicioTest.java
    - FrmMenuTest.java
    - FrmboletaTest.java
    - FrmcarritoTest.java
    - FrmclienteTest.java
    - FrmempleadoTest.java
    - FrmlibroTest.java
    - LoginTest.java
- Libraries

testMain - Navigator

Members

- FrmMenuTest
  - FrmMenuTest()
  - testMain()

EnEmpleadoTest.java | EnLibroTest.java | ConexionTest.java | FrmInicioTest.java | FrmMenuTest.java

Source History

```
8   import static org.junit.Assert.*;
9
10  /**
11   *
12   * @author Reneman
13   */
14  public class FrmMenuTest {
15
16      public FrmMenuTest() {
17      }
18
19      @Test
20      public void testMain() {
21          System.out.println("main");
22          String[] args = null;
23          FrmMenu.main(args);
```

Presentación.FrmMenuTest > testMain

Test Results | Output

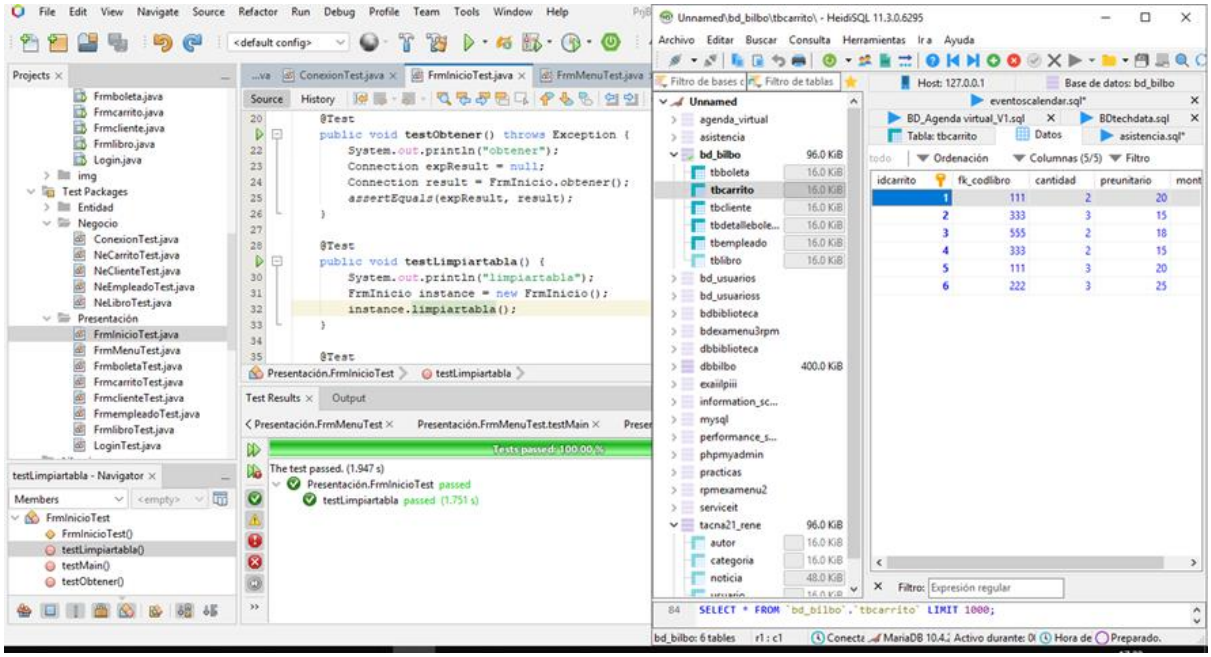
Test | Entidad.EnCarritoTest | Presentación.FrmInicioTest | Presentación.FrmInicioTest.testObtener | Presentación.FrmMenuTest

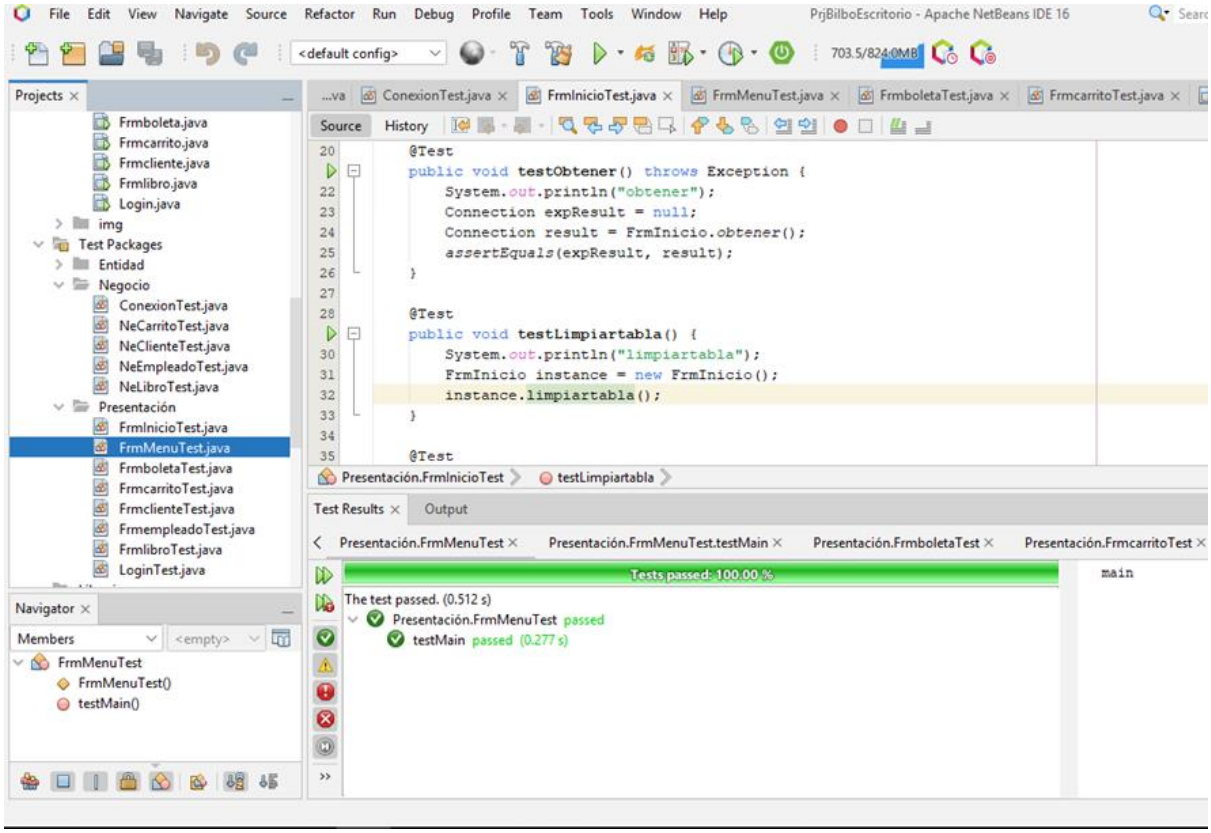
Tests passed: 100.00 %

The test passed. (0.478 s)

- Presentación.FrmMenuTest passed
  - testMain passed (0.285 s)

main


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Frmboleta.java
Frmcarrito.java
Frmcliente.java
Frmlibro.java
Login.java
img
Test Packages
Entidad
Negocio
ConexionTest.java
NeCarritoTest.java
NeClienteTest.java
NeEmpleadoTest.java
NeLibroTest.java
Presentación
FrmInicioTest.java
FrmMenuTest.java
FrmboletaTest.java
FrmcarritoTest.java
FrmclienteTest.java
FrmempleadoTest.java
FrmlibroTest.java
LoginTest.java
@Test
public void testObtener() throws Exception {
System.out.println("obtener");
Connection expResult = null;
Connection result = FrmInicio.obtener();
assertEquals(expResult, result);
}
@Test
public void testLimpiartabla() {
System.out.println("limpiartabla");
FrmInicio instance = new FrmInicio();
instance.limpiartabla();
}
Presentación.FrmInicioTest
testLimpiartabla
Test Results
Output
Tests passed: 100.00 %
The test passed. (1.947 s)
Presentación.FrmInicioTest passed
testLimpiartabla passed (1.751 s)
testLimpiartabla - Navigator
FrmInicioTest
FrmInicioTest()
testLimpiartabla()
testMain()
testObtener()
Unnamed\bd_bilbo\tbcarrito\ - HeidiSQL 11.3.0.6295
Archivo Editar Buscar Consulta Herramientas Ir a Ayuda
Host: 127.0.0.1
Base de datos: bd_bilbo
Unnamed
agenda_virtual
asistencia
bd_bilbo
tbboleta
tbcarrito
tbcliente
tbempleado
tblibro
bd_usuarios
bd_usuarioss
bdbiblioteca
bdexamenu3rpm
dbbiblioteca
dbbilbo
exailpiii
information_sc...
mysql
performance_s...
phpmyadmin
practicas
rpmexamenu2
serviceit
tacna21_rene
autor
categoria
noticia
eventoscalendar.sql*
BD_Agenda virtual_V1.sql
BDtechdata.sql
Tabla: tbcarrito
Datos
asistencia.sql*
Ordenación
Columnas (5/5)
Filtro
idcarrito
fk_codlibro
cantidad
preunitario
SELECT * FROM `bd_bilbo`.`tbcarrito` LIMIT 1000;


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
PrjBilboEscritorio - Apache NetBeans IDE 16
<default config>
703.5/824.0MB
Projects
Frmboleta.java
Frmcarrito.java
Frmcliente.java
Frmlibro.java
Login.java
img
Test Packages
Entidad
Negocio
ConexionTest.java
NeCarritoTest.java
NeClienteTest.java
NeEmpleadoTest.java
NeLibroTest.java
Presentación
FrmInicioTest.java
FrmMenuTest.java
FrmboletaTest.java
FrmcarritoTest.java
FrmclienteTest.java
FrmempleadoTest.java
FrmlibroTest.java
LoginTest.java
Source
History
@Test
public void testObtener() throws Exception {
System.out.println("obtener");
Connection expResult = null;
Connection result = FrmInicio.obtener();
assertEquals(expResult, result);
}
@Test
public void testLimpiartabla() {
System.out.println("limpiartabla");
FrmInicio instance = new FrmInicio();
instance.limpiartabla();
}
@Test
Presentación.FrmInicioTest
testLimpiartabla
Test Results
Output
Presentación.FrmMenuTest
Presentación.FrmMenuTest.testMain
Presentación.FrmboletaTest
Presentación.FrmcarritoTest
Tests passed: 100.00 %
The test passed. (0.512 s)
Presentación.FrmMenuTest passed
testMain passed (0.277 s)
main
Navigator
Members
FrmMenuTest
FrmMenuTest()
testMain()

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help PrjBilboEscritorio - Apache NetBeans IDE 16

Projects

- Frmboleta.java
- Frmcarrito.java
- Frmcliente.java
- Frmlibro.java
- Login.java
- img
- Test Packages
  - Entidad
  - Negocio
    - ConexionTest.java
    - NeCarritoTest.java
    - NeClienteTest.java
    - NeEmpleadoTest.java
    - NeLibroTest.java
  - Presentación
    - FrmInicioTest.java
    - FrmMenuTest.java
    - FrmboletaTest.java
    - FrmcarritoTest.java
    - FrmclienteTest.java
    - FrmempleadoTest.java
    - FrmlibroTest.java
    - LoginTest.java

FrmboletaTest - Navigator

Members

- FrmboletaTest
  - FrmboletaTest()
  - testLimpiartabla()
  - testMain()
  - testObtener()

...va | ConexionTest.java | FrmInicioTest.java | FrmMenuTest.java | FrmboletaTest.java | FrmcarritoTe

Source | History

```
import java.sql.Connection;
import org.junit.Test;
import static org.junit.Assert.*;

/**
 *
 * @author Reneman
 */
public class FrmboletaTest {

    public FrmboletaTest() {
    }

    @Test
    public void testObtener() throws Exception {
        System.out.println("obtener");
```

Presentación.FrmboletaTest

Test Results | Output

.FrmMenuTest.testMain | Presentación.FrmboletaTest | Presentación.FrmcarritoTest | Presentación.FrmInicioTest.testL

Tests passed: 100.00 %

The test passed. (0.189 s)

- Presentación.FrmboletaTest passed
  - testObtener passed (0.008 s)

File Edit View Navigate Source Refactor Run Debug Profile Team Tools

Projects

- NeCarritoTest.java
- NeClienteTest.java
- NeEmpleadoTest.java
- NeLibroTest.java
- Presentación
  - FrmInicioTest.java
  - FrmMenuTest.java
  - FrmboletaTest.java
  - FrmcarritoTest.java
  - FrmclienteTest.java
  - FrmempleadoTest.java
  - FrmlibroTest.java
  - LoginTest.java
- Libraries
  - mysql-connector-java-8.0.26.jar
  - RSScaleLabel.jar
  - JDK 20 (Default)
- Test Libraries
- PrjExamenUnidad1_PomaManchego
- PrjExaU1Corección_PomaManchego
- PrjLab0RPM1
- PrjProyectoBilboBeta
- PrjProyectoBilboBetaU1

testObtener - Navigator

Members

- FrmcarritoTest
  - FrmcarritoTest()
  - testLimpiartabla()
  - testMain()
  - testObtener()

...va | EnLibroTest.java | ConexionTest.jav

Source | History

```
    }

    @Test
    public void testObtener()
        System.out.println("ob
        Connection expResult =
        Connection result = Fr
        assertEquals(expResult
    }

    @Test
    public void testLimpiartab
        System.out.println("li
        Frmcarrito instance =
        instance.limpiartabla(
```

Presentación.FrmcarritoTest > testObtener

Test Results | Output

Presentación.FrmInicioTest.testObtener

2 tests passed, 1 test failed. (1.282 s)

- Presentación.FrmcarritoTest Failed
  - testObtener Failed: expected:<null> b
  - testMain passed (0.228 s)
  - testLimpiartabla passed (0.429 s)

Unnamed\bd_bilbo\tbcarrito\ - HeidiSQL 11.3.0.6295

Archivo Editar Buscar Consulta Herramientas Ir a Ayuda

Filtro de bases de datos | Filtro de tablas

- Unnamed
  - agenda_virtual
  - asistencia
  - bd_bilbo 96.0 KiB
    - tbboleta 16.0 KiB
    - tbcarrito 16.0 KiB
    - tbcliente 16.0 KiB
    - tbdetalleboleta 16.0 KiB
    - tbempleado 16.0 KiB
    - tblibro 16.0 KiB
  - bd_usuarios
  - bd_usuarioss
  - bdbiblioteca
  - bdexamenu3rpm
  - dbbiblioteca
  - dbbilbo 400.0 KiB
  - exaiilpiii
  - information_schema
  - mysql
  - performance_schema
  - phpmyadmin
  - practicas
  - rpmexamenu2
  - serviceit
  - tacna21_rene 96.0 KiB
    - autor 16.0 KiB
    - categoria 16.0 KiB
    - noticia 48.0 KiB

Host: 127.0.0.1 | Base de datos: bd_bilbo | Tabla: tbcarrito

BD_Agenda virtual_V1.sql | BDtechdata.sql

Datos | asistencia.sql* | eventoscalendar.sql*

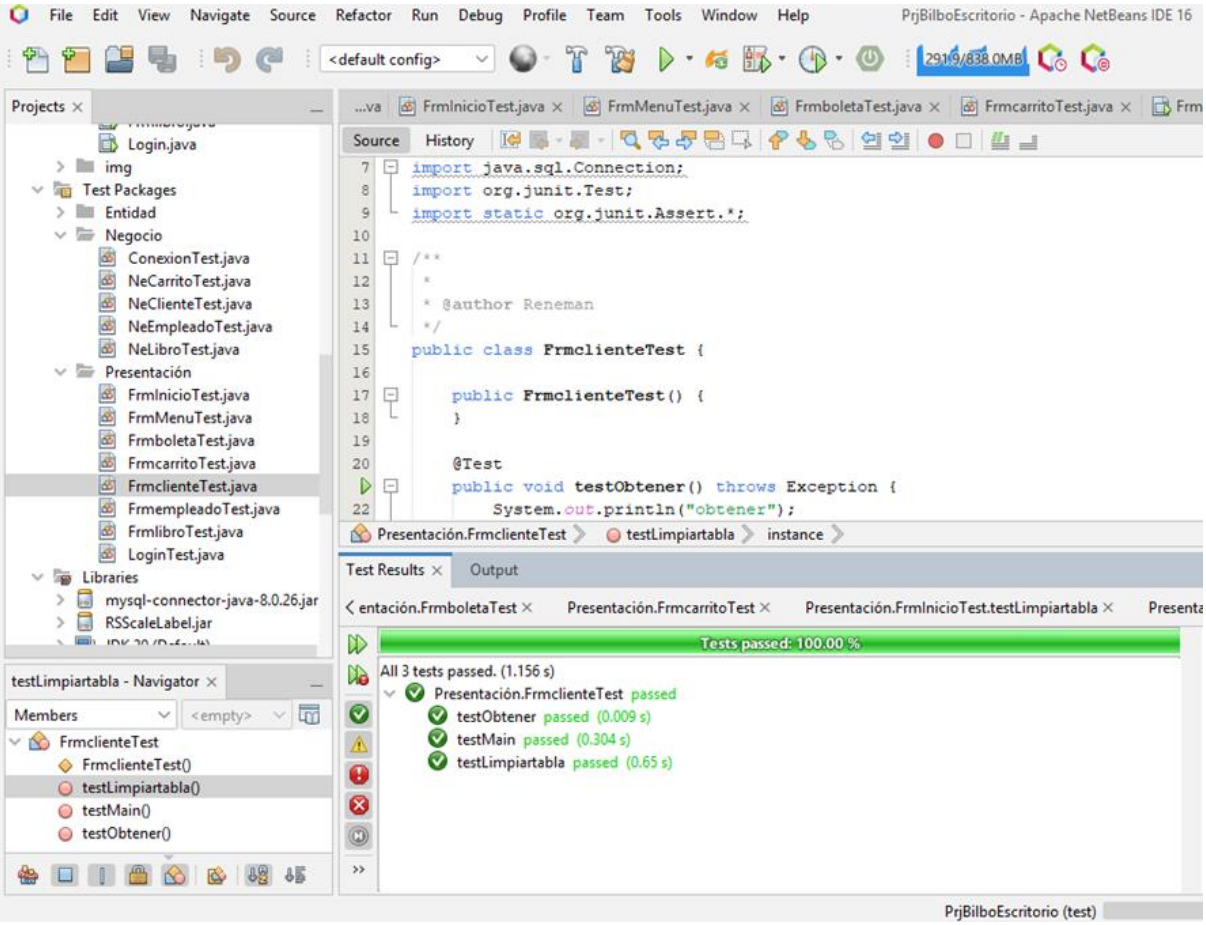
Ordenación | Columnas (5/5) | Filtro

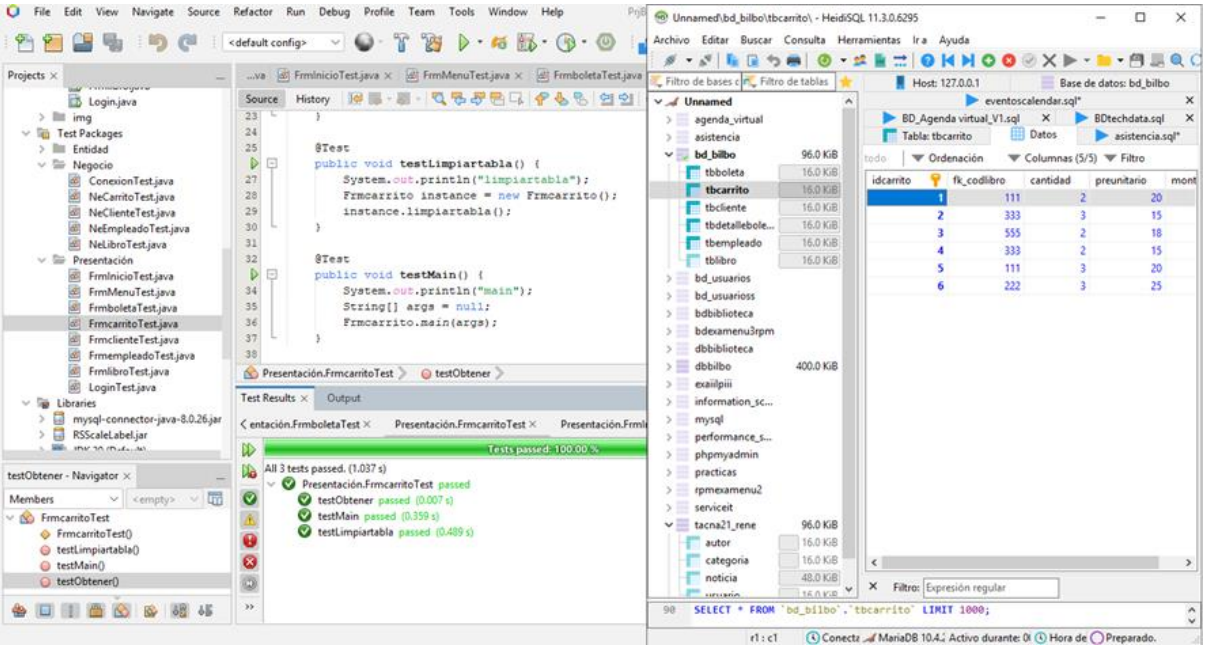
idcarrito	fk_codlibro	cantidad	preunitario	monto
1	111	2	20	40
2	333	3	15	45
3	555	2	18	36
4	333	2	15	30
5	111	3	20	60
6	222	3	25	75

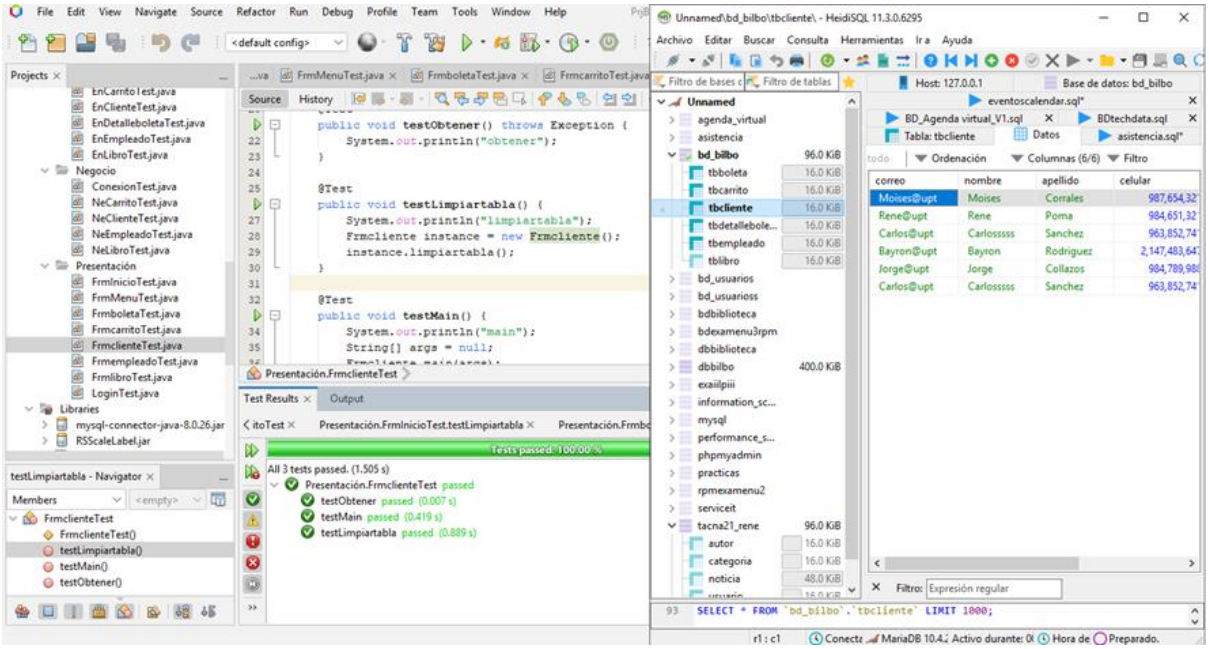
Filtro: Expresión regular

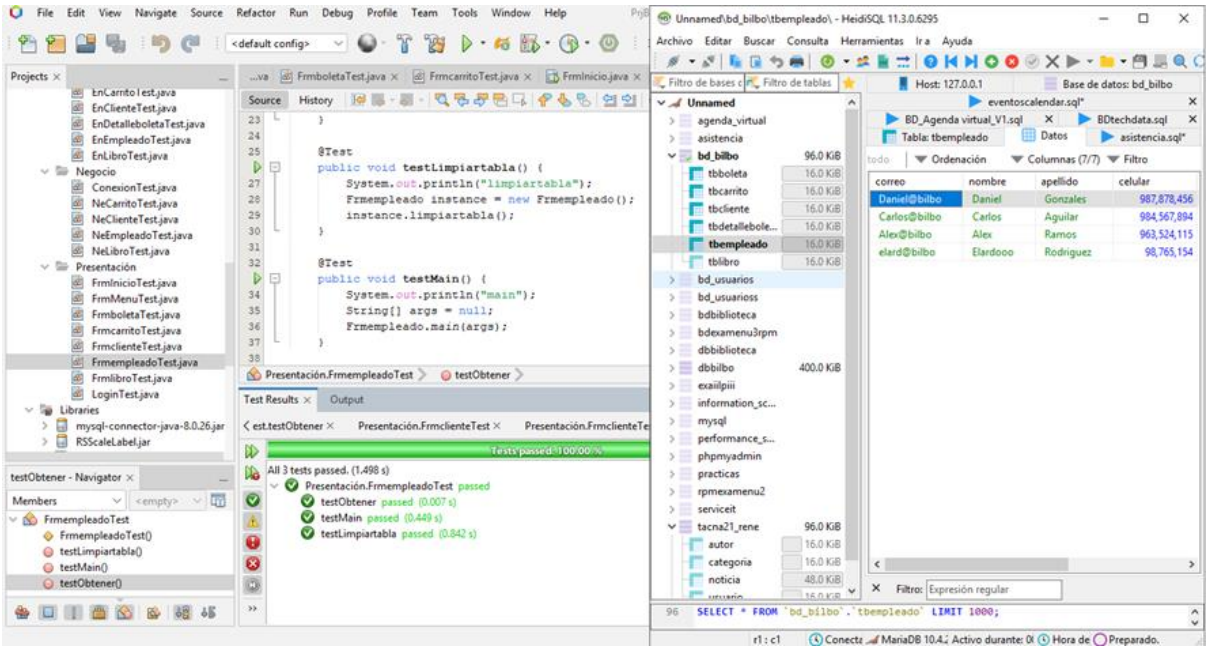
```
SELECT * FROM `bd_bilbo`.`tbcarrito` LIMIT 1000;
```

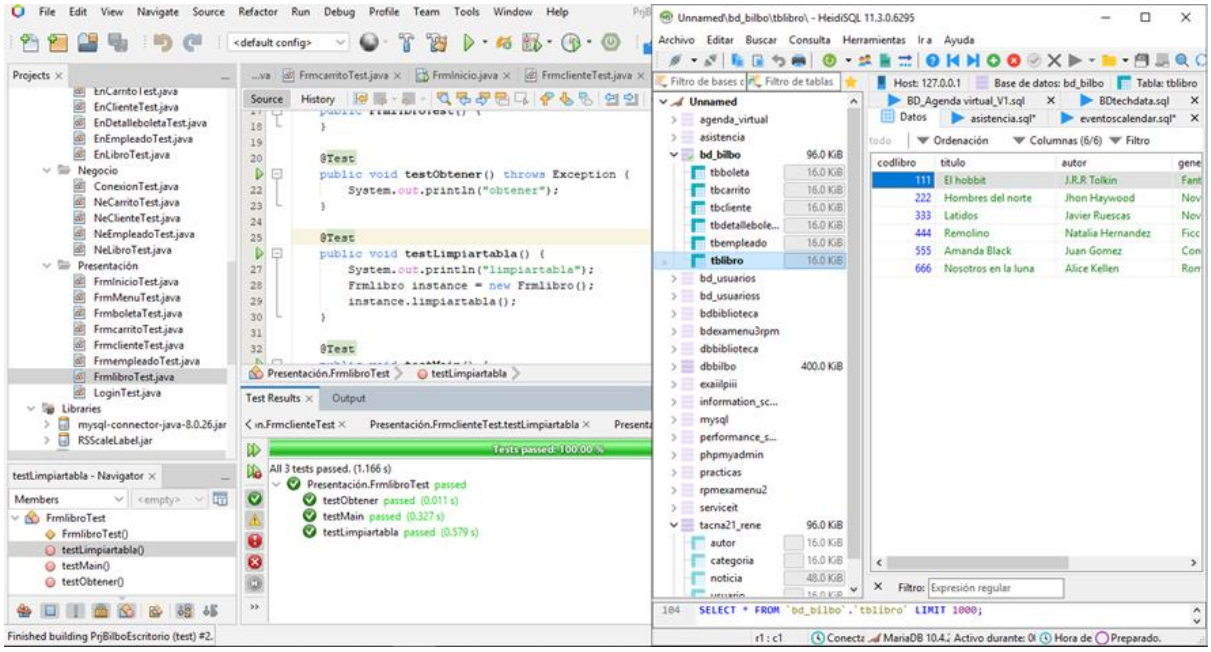
bd_bilbo: 6 tables | r1 : c1 | Conectado: MariaDB 10.4.24 | Activo durante: 00:01 | Hora del se | Preparado.

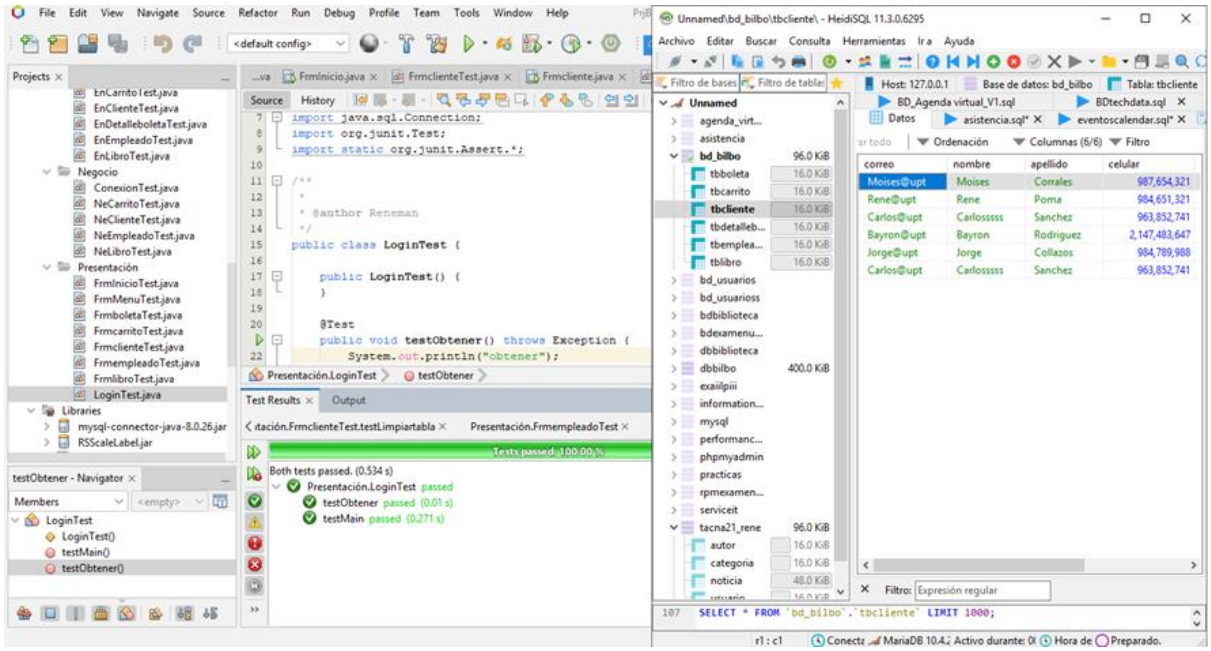

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
PrjBilboEscritorio - Apache NetBeans IDE 16
<default config>
Projects
Login.java
img
Test Packages
Entidad
Negocio
ConexionTest.java
NeCarritoTest.java
NeClienteTest.java
NeEmpleadoTest.java
NeLibroTest.java
Presentación
FrmInicioTest.java
FrmMenuTest.java
FrmboletaTest.java
FrmcarritoTest.java
FrmclienteTest.java
FrmempleadoTest.java
FrmlibroTest.java
LoginTest.java
Libraries
mysql-connector-java-8.0.26.jar
RSScaleLabel.jar
import java.sql.Connection;
import org.junit.Test;
import static org.junit.Assert.*;
/**
* @author Reneman
*/
public class FrmclienteTest {
public FrmclienteTest() {
}
@Test
public void testObtener() throws Exception {
System.out.println("obtener");
Presentación.FrmclienteTest
testLimpiartabla
instance
Test Results
Output
Tests passed: 100.00 %
All 3 tests passed. (1.156 s)
Presentación.FrmclienteTest passed
testObtener passed (0.009 s)
testMain passed (0.304 s)
testLimpiartabla passed (0.65 s)
testLimpiartabla - Navigator
Members
FrmclienteTest
FrmclienteTest()
testLimpiartabla()
testMain()
testObtener()
PrjBilboEscritorio (test)


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
@Test
public void testLimpiartabla() {
System.out.println("limpiartabla");
Frmcarrito instance = new Frmcarrito();
instance.limpiartabla();
}
@Test
public void testMain() {
System.out.println("main");
String[] args = null;
Frmcarrito.main(args);
}
Presentación.FrmcarritoTest
testObtener
Test Results
Output
Tests passed: 100.00 %
All 3 tests passed. (1.037 s)
Presentación.FrmcarritoTest passed
testObtener passed (0.007 s)
testMain passed (0.359 s)
testLimpiartabla passed (0.489 s)
testObtener - Navigator
FrmcarritoTest
FrmcarritoTest()
testLimpiartabla()
testMain()
testObtener()
Unnamed\bd_bilbo\tbcarrito\ - HeidiSQL 11.3.0.6295
Archivo Editar Buscar Consulta Herramientas Ir a Ayuda
Host: 127.0.0.1
Base de datos: bd_bilbo
eventoscalendar.sql*
BD_Agenda virtual_V1.sql
BDtechdata.sql
Tabla: tbcarrito
Datos
asistencia.sql*
Ordenación
Columnas (5/5)
Filtro
idcarrito fk_codlibro cantidad preunitario
1 111 2 20
2 333 3 15
3 555 2 18
4 333 2 15
5 111 3 20
6 222 3 25
SELECT * FROM `bd_bilbo`.`tbcarrito` LIMIT 1000;

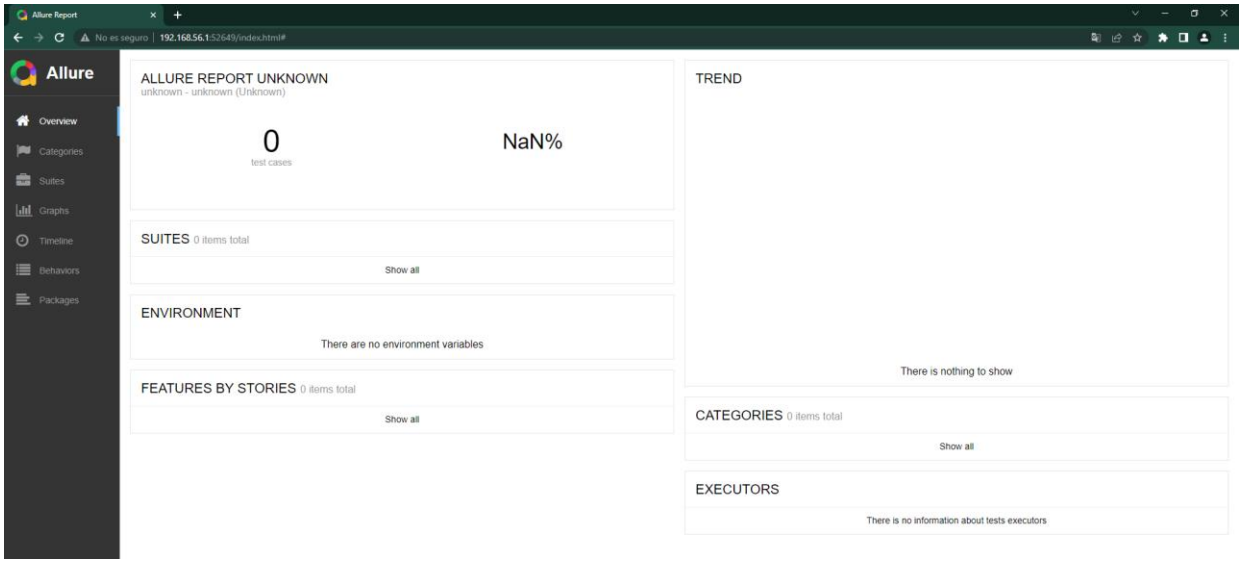

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
EnCarritoTest.java
EnClienteTest.java
EnDetalleboletaTest.java
EnEmpleadoTest.java
EnLibroTest.java
Negocio
ConexionTest.java
NeCarritoTest.java
NeClienteTest.java
NeEmpleadoTest.java
NeLibroTest.java
Presentación
FrmInicioTest.java
FrmMenuTest.java
FrmboletaTest.java
FrmcarritoTest.java
FrmclienteTest.java
FrmempleadoTest.java
FrmlibroTest.java
LoginTest.java
Libraries
mysql-connector-java-8.0.26.jar
RSScaleLabel.jar
testLimpiartabla - Navigator
Members
FrmclienteTest
FrmclienteTest()
testLimpiartabla()
testMain()
testObtener()
public void testObtener() throws Exception {
System.out.println("obtener");
}
@Test
public void testLimpiartabla() {
System.out.println("limpiartabla");
Frmcliente instance = new Frmcliente();
instance.limpiartabla();
}
@Test
public void testMain() {
System.out.println("main");
String[] args = null;
Presentación.FrmclienteTest
Test Results
Output
Tests passed: 100.00 %
All 3 tests passed. (1.505 s)
Presentación.FrmclienteTest passed
testObtener passed (0.007 s)
testMain passed (0.419 s)
testLimpiartabla passed (0.889 s)
Unnamed\bd_bilbo\tbcliente\ - HeidiSQL 11.3.0.6295
Archivo Editar Buscar Consulta Herramientas Ir a Ayuda
Host: 127.0.0.1
Base de datos: bd_bilbo
Unnamed
agenda_virtual
asistencia
bd_bilbo 96.0 KiB
tbboleta 16.0 KiB
tbcarrito 16.0 KiB
tbcliente 16.0 KiB
tbdetallebole... 16.0 KiB
tbempleado 16.0 KiB
tblibro 16.0 KiB
bd_usuarios
bd_usuarioss
bdbiblioteca
bdexamenu3rpm
dbbiblioteca
dbbilbo 400.0 KiB
exaiilpiii
information_sc...
mysql
performance_s...
phpmyadmin
practicas
rpmexamenu2
serviceit
tacna21_rene 96.0 KiB
autor 16.0 KiB
categoria 16.0 KiB
noticia 48.0 KiB
eventoscalendar.sql*
BD_Agenda virtual_V1.sql
BDtechdata.sql
Tabla: tbcliente
Datos
asistencia.sql*
Ordenación
Columnas (6/6)
Filtro
correo nombre apellido celular
Moises@upt Moises Corrales 987,654,32
Rene@upt Rene Poma 984,651,32
Carlos@upt Carlosssss Sanchez 963,852,74
Bayron@upt Bayron Rodriguez 2,147,483,64
Jorge@upt Jorge Collazos 984,789,98
Carlos@upt Carlosssss Sanchez 963,852,74
Filtro: Expresión regular
SELECT * FROM `bd_bilbo`.`tbcliente` LIMIT 1000;
r1 : c1
Conecta MariaDB 10.4. Activo durante: 0(Hora de Preparado.


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
EnCarritoTest.java
EnClienteTest.java
EnDetalleboletaTest.java
EnEmpleadoTest.java
EnLibroTest.java
Negocio
ConexionTest.java
NeCarritoTest.java
NeClienteTest.java
NeEmpleadoTest.java
NeLibroTest.java
Presentación
FrmInicioTest.java
FrmMenuTest.java
FrmboletaTest.java
FrmcarritoTest.java
FrmclienteTest.java
FrmempleadoTest.java
FrmlibroTest.java
LoginTest.java
Libraries
mysql-connector-java-8.0.26.jar
RSScaleLabel.jar
testObtener - Navigator
Members
FrmempleadoTest
FrmempleadoTest()
testLimpiartabla()
testMain()
testObtener()
}
@Test
public void testLimpiartabla() {
System.out.println("limpiartabla");
Frmempleado instance = new Frmempleado();
instance.limpiartabla();
}
@Test
public void testMain() {
System.out.println("main");
String[] args = null;
Frmempleado.main(args);
}
Presentación.FrmempleadoTest
testObtener
Test Results
Output
Tests passed: 100.00 %
All 3 tests passed. (1.498 s)
Presentación.FrmempleadoTest passed
testObtener passed (0.007 s)
testMain passed (0.449 s)
testLimpiartabla passed (0.842 s)
Unnamed\bd_bilbo\tbempleado\ - HeidiSQL 11.3.0.6295
Archivo Editar Buscar Consulta Herramientas Ir a Ayuda
Host: 127.0.0.1
Base de datos: bd_bilbo
Unnamed
agenda_virtual
asistencia
bd_bilbo 96.0 KiB
tbboleta 16.0 KiB
tbcarrito 16.0 KiB
tbcliente 16.0 KiB
tbdetallebole... 16.0 KiB
tbempleado 16.0 KiB
tblibro 16.0 KiB
bd_usuarios
bd_usuarioss
bdbiblioteca
bdexamenu3rpm
dbbiblioteca
dbbilbo 400.0 KiB
exaiilpiii
information_sc...
mysql
performance_s...
phpmyadmin
practicas
rpmexamenu2
serviceit
tacna21_rene 96.0 KiB
autor 16.0 KiB
categoria 16.0 KiB
noticia 48.0 KiB
eventoscalendar.sql*
BD_Agenda virtual_V1.sql
BDtechdata.sql
Tabla: tbempleado
Datos
asistencia.sql*
Ordenación
Columnas (7/7)
Filtro
correo nombre apellido celular
Daniel@bilbo Daniel Gonzales 987,878,456
Carlos@bilbo Carlos Aguilar 984,567,894
Alex@bilbo Alex Ramos 963,524,115
elard@bilbo Elardooo Rodriguez 98,765,154
Filtro: Expresión regular
SELECT * FROM `bd_bilbo`.`tbempleado` LIMIT 1000;
r1 : c1
Conecta MariaDB 10.4. Activo durante: 0(Hora de Preparado.

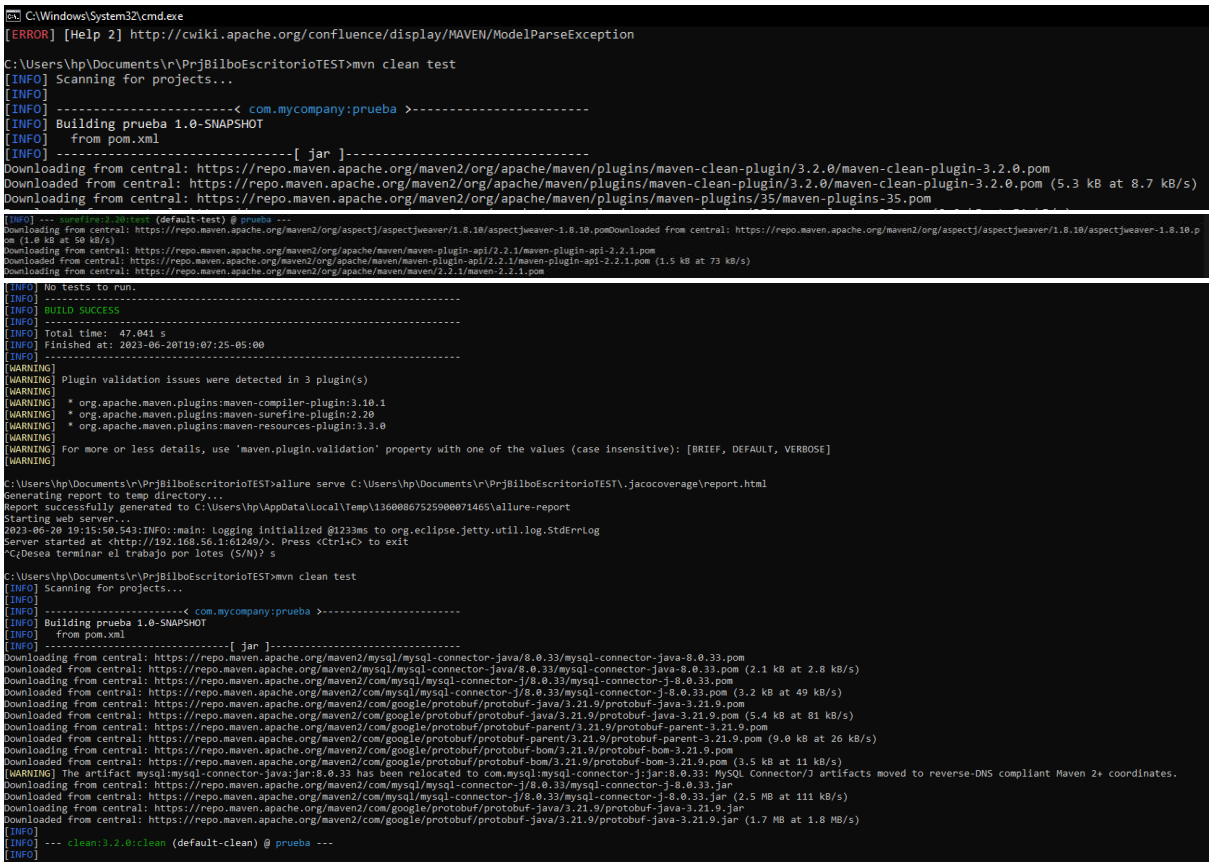




- Reporte de Pruebas guiadas por el comportamiento (BDD Given When Then)



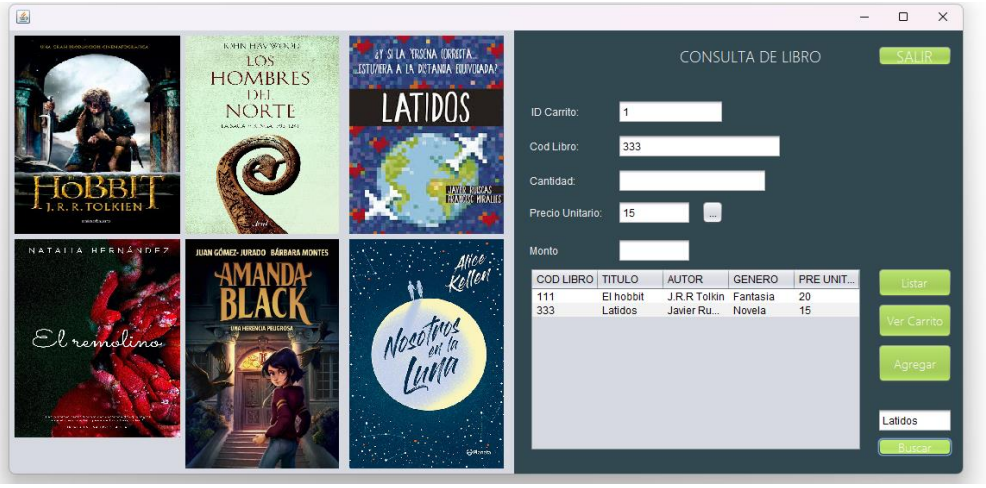
Cmd:



b) Pruebas de aceptación basadas en Desarrollo Guiado por el Comportamiento una por cada caso de uso o historia de usuario.

- Buscar libro

```java
private void btnbuscarActionPerformed(java.awt.event.ActionEvent evt) {
    java.sql.Connection con = null;
    try {
        con = obtener();
        ps = con.prepareStatement("SELECT * FROM tblibro where codlibro = ?");
        ps.setString(1, txtcod.getText());
        rs = ps.executeQuery();
        if (rs.next()) {
            txtcodigo.setText(rs.getString("codlibro"));
            txttitulo.setText(rs.getString("titulo"));
            txtautor.setText(rs.getString("autor"));
            txtgenero.setText(rs.getString("genero"));
            txtprecio.setText(rs.getString("preunitario"));
            txtcantidad.setText(rs.getString("cantidad"));
            String[] dato = new String[7];
            try {
                rs = ps.executeQuery();
                while (rs.next()) {
                    dato[0] = rs.getString("codlibro");
                    dato[1] = rs.getString("titulo");
                    dato[2] = rs.getString("autor");
                    dato[3] = rs.getString("genero");
                    dato[4] = rs.getString("preunitario");
                    dato[5] = rs.getString("cantidad");
                    model.addRow(dato);
                }
                tblibro.setModel(model);
            } catch (SQLException ex) {
                System.out.println(ex.getMessage());
            }
        } else {
            JOptionPane.showMessageDialog(null, "No existe un libro asi");
        }
    } catch (Exception e) {
        System.err.println(e);
    }
}
```



- Comprar stock del libro

```java
private void btnlistarActionPerformed(java.awt.event.ActionEvent evt) {
    ResultSet rs;
    limpiarcajas();
        limpiartabla();
    NeLibro objNcli=new NeLibro();
    String[] dato=new String[6];
    try {
        rs= objNcli.MtdListar();
        while(rs.next())
        {
            dato[0]=rs.getString("codlibro");
            dato[1]=rs.getString("titulo");
            dato[2]=rs.getString("autor");
            dato[3]=rs.getString("genero");
            dato[4]=rs.getString("preunitario");
            model.addRow(dato);
        }
        tblibro.setModel(model);
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }
}
```

- Agregar libro

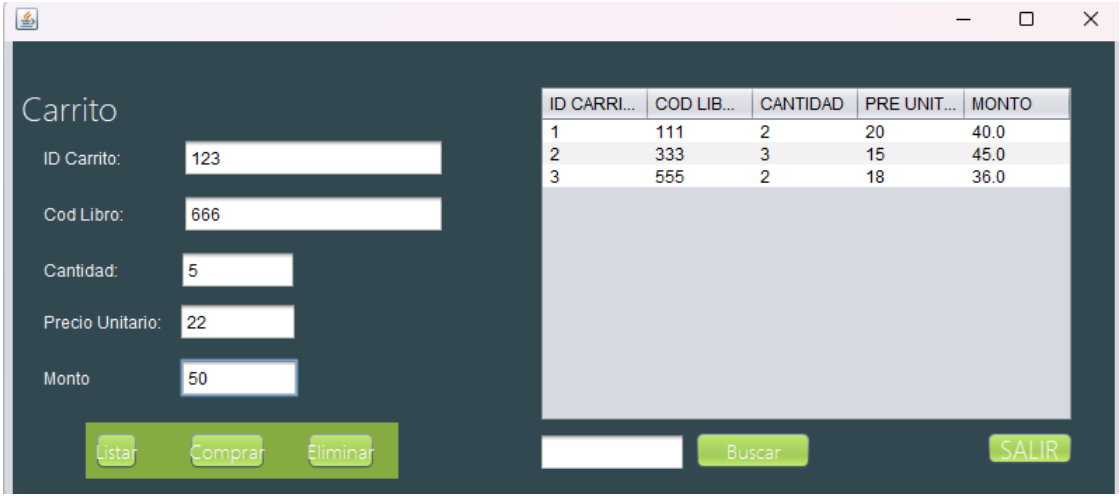
```
private void btnagregarActionPerformed(java.awt.event.ActionEvent evt) {

    EnLibro objcli = new EnLibro();
    NeLibro objNcli = new NeLibro();
    objcli.setCodlibro( codlibro: Integer.parseInt( s: txtcodigo.getText()));
    objcli.setTitulo( titulo: txttitulo.getText());
    objcli.setAutor( autor: txtautor.getText());
    objcli.setGenero( genero: txtgenero.getText());
    objcli.setPreunitario( preunitario: Integer.parseInt( s: txtprecio.getText()));
    objcli.setCantidad( cantidad: Integer.parseInt( s: txtcantidad.getText()));

    if (objNcli.MtdAgregar( lib: objcli)==true)
    {
        JOptionPane.showMessageDialog( parentComponent: null, message: "Se guardó con éxito");
        limpiarcajas();
        limpiartabla();
    }
}
```

- Realizar pago

```
private void btncomprarActionPerformed(java.awt.event.ActionEvent evt) {
    Frmboleta frm = new Frmboleta();
    frm.show();
    this.hide();
}
```



- Realizar venta

```
private void ImprimirActionPerformed(java.awt.event.ActionEvent evt) {
    java.sql.Connection con = null;

    try {
        con = obtener();
        ps = con.prepareStatement("insert into tbboleta(nroboleta,serboleta,fecha,total) values (?,?,?,?)");
        ps.setString(1, txtnro.getText());
        ps.setString(2, txtserie.getText());
        ps.setString(3, txtfecha.getText());
        ps.setString(4, txttotal.getText());
        int res = ps.executeUpdate();
        if (res > 0) {
        JOptionPane.showMessageDialog(null, "Impreso con exito");
        btnsalir.setVisible(true);
        ps = con.prepareStatement("insert into tbdetalleboleta(fk_idcarrito,cantidad,importe,fk_seriebol,fk_nrobol) values (?,?,?,?,?)");
        ps.setString(1, txtidcarro.getText());
        ps.setString(2, txtcantidad.getText());
        ps.setString(3, txttotal.getText());
        ps.setString(4, txtserie.getText());
        ps.setString(5, txtnro.getText());
        int res2 = ps.executeUpdate();
        limpiarcajas();
            limpiartabla();

        } else {
            JOptionPane.showMessageDialog(null, "Hubo un error al imprimir");
            limpiarcajas();
            limpiartabla();
        }
    } catch (Exception e) {
        System.err.println(e);
    }

}
```

BOLETA
Serie:
Nro:
Fecha:
DATOS CLIENTE:
E-mail:
...
Nombre:
Apellido:
Celular:
Diercción:
DATOS DE LIBRO:
Title 1
Title 2
Title 3
Title 4
codigo:
cant.
TOTAL:
SALIR
Imprimir