

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:

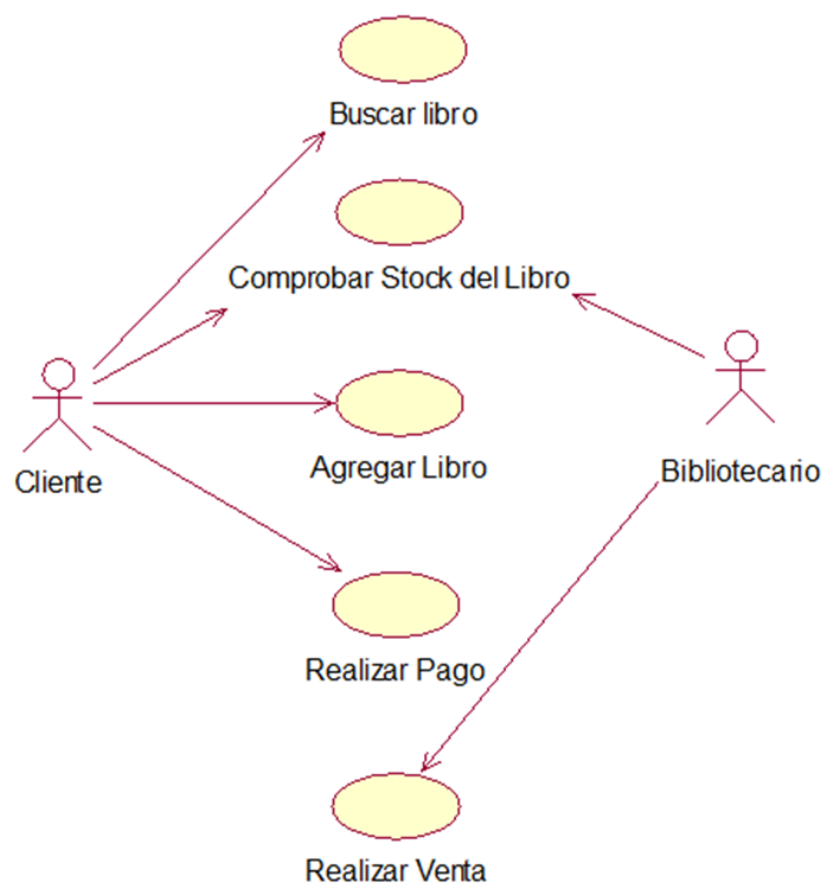


Diagrama de Clases:

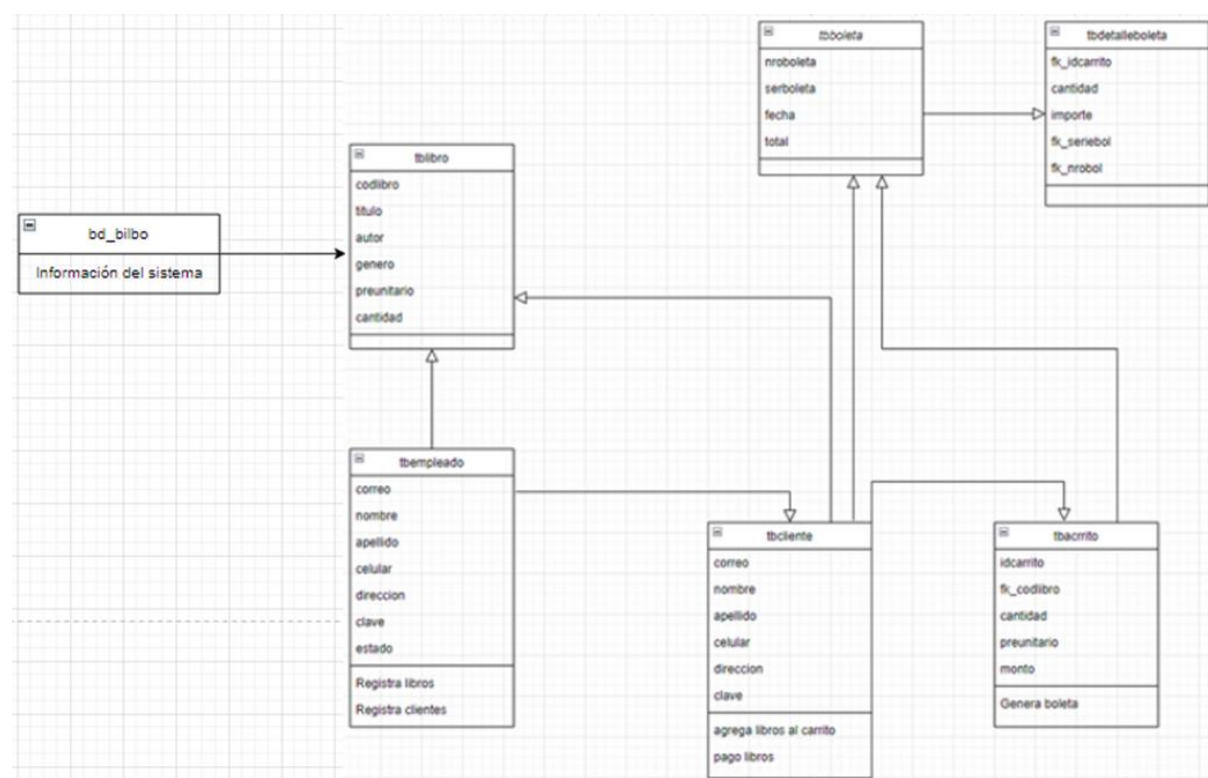
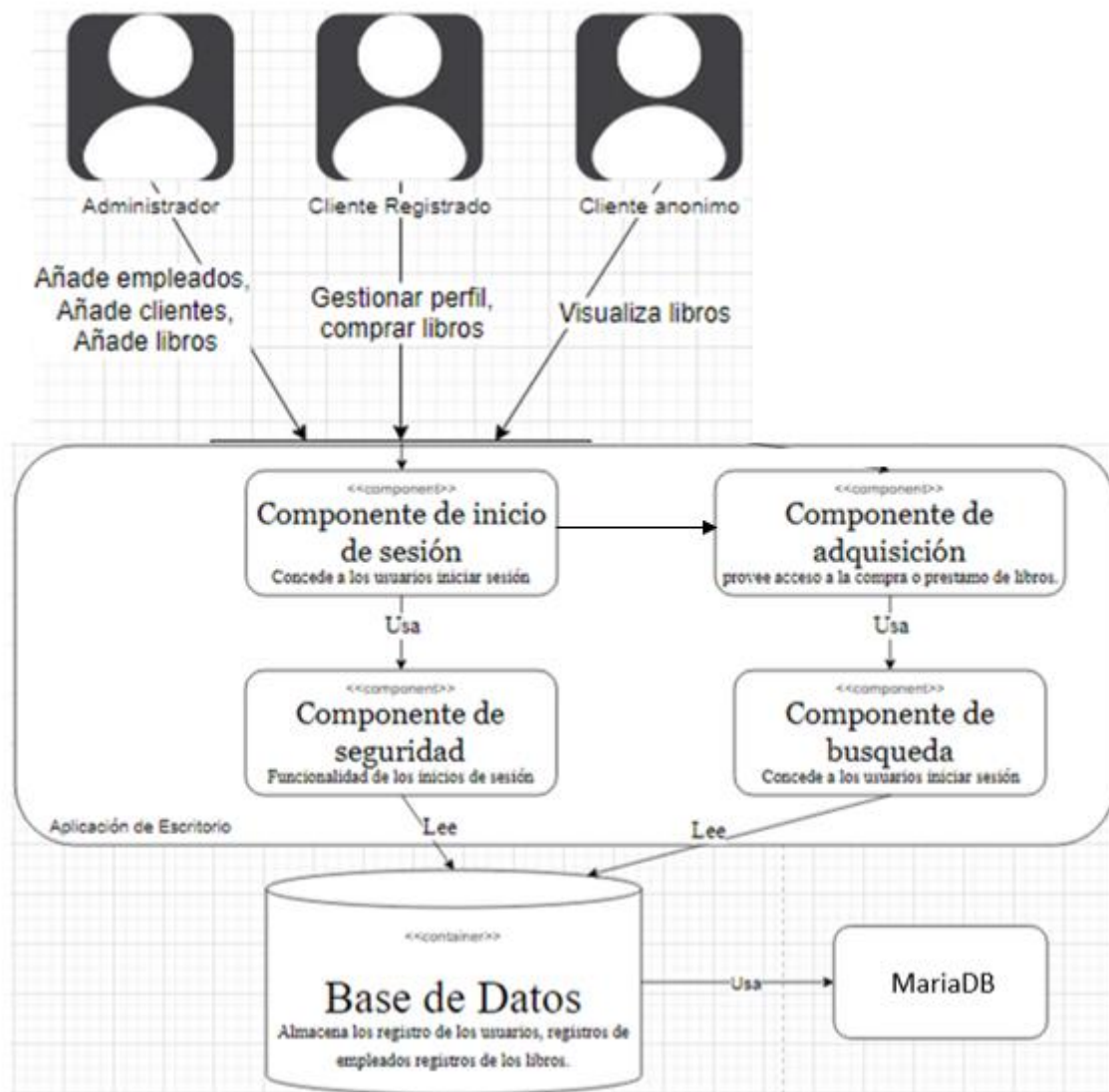


Diagrama de Componentes:



Arquitectura:

Diagrama Contextual:

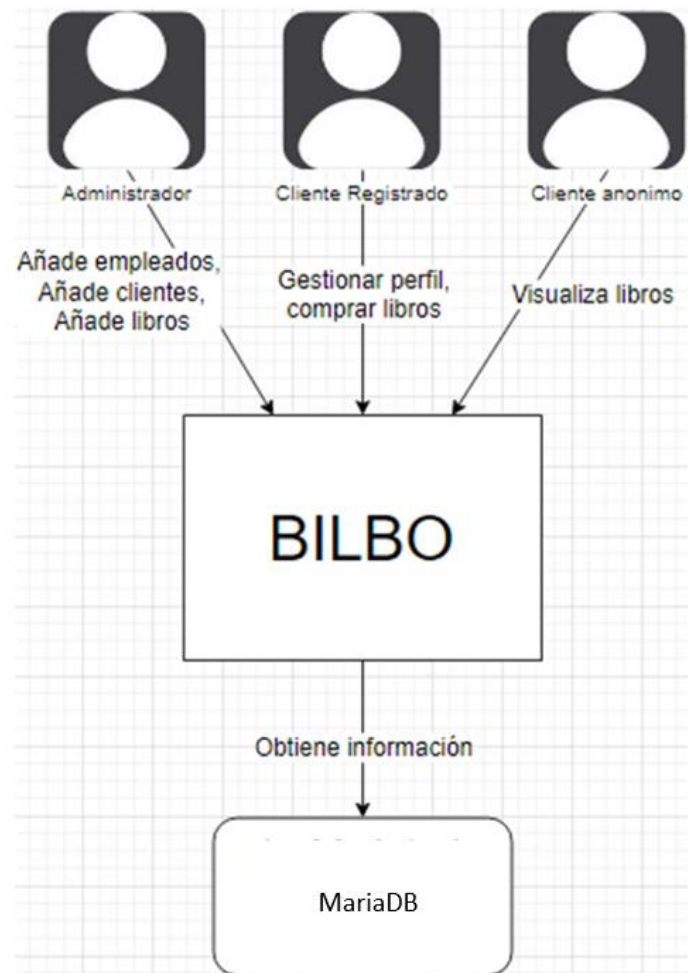
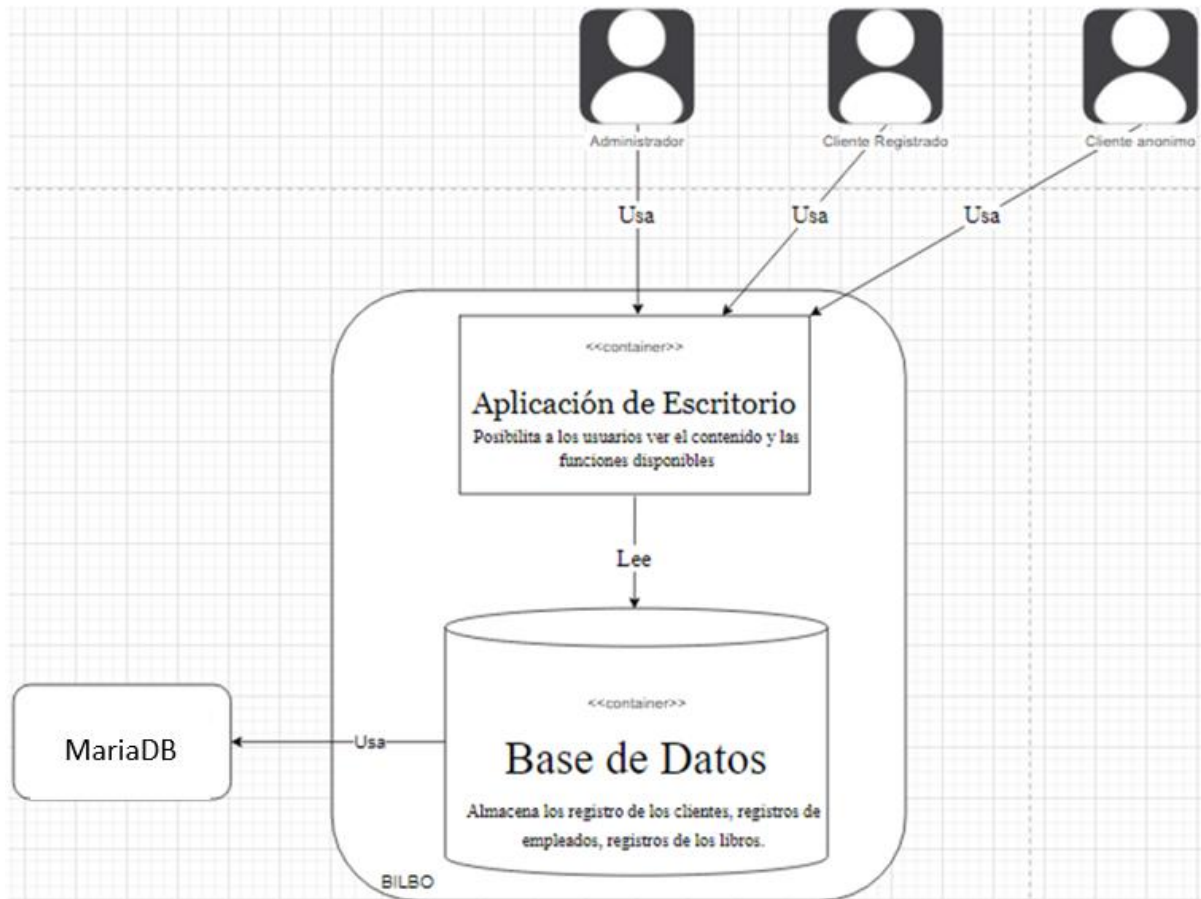


Diagrama de Contenedores:



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.

1  Bugs

Reliability B

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

8  Vulnerabilities

Security E

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

7d Debt

438  Code Smells

Maintainability A

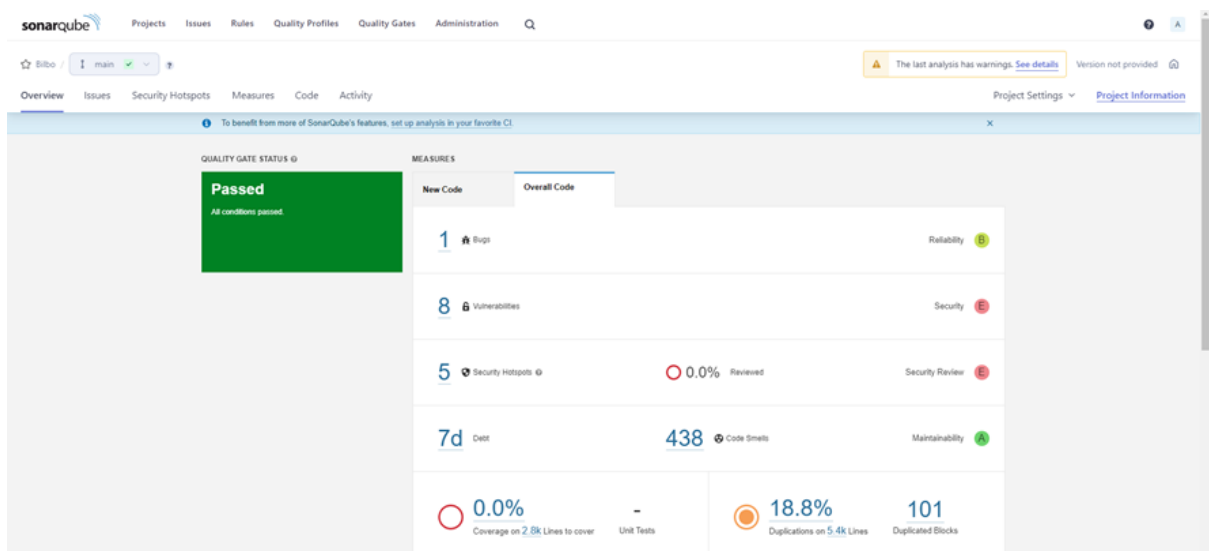
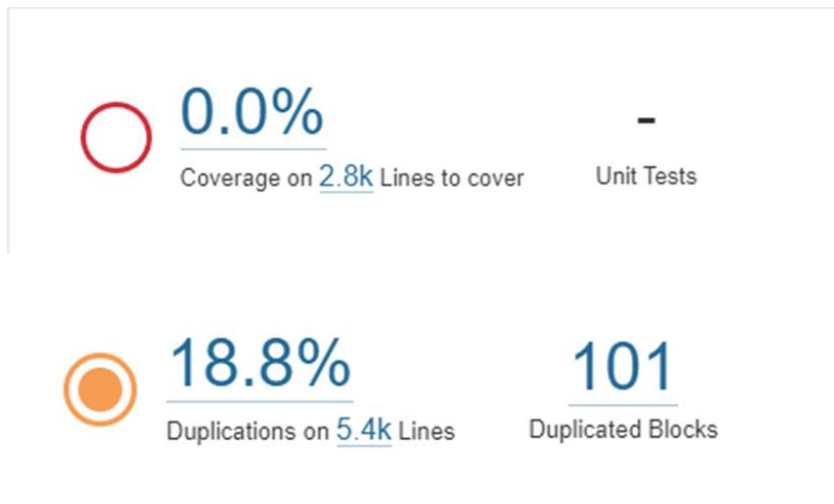
- En los puntos de seguridad tenemos 5 a corregir, sonar nos calificó con “E” en la reseña de seguridad.

5  Security Hotspots 

○ 0.0% Reviewed

Security Review E

- 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:

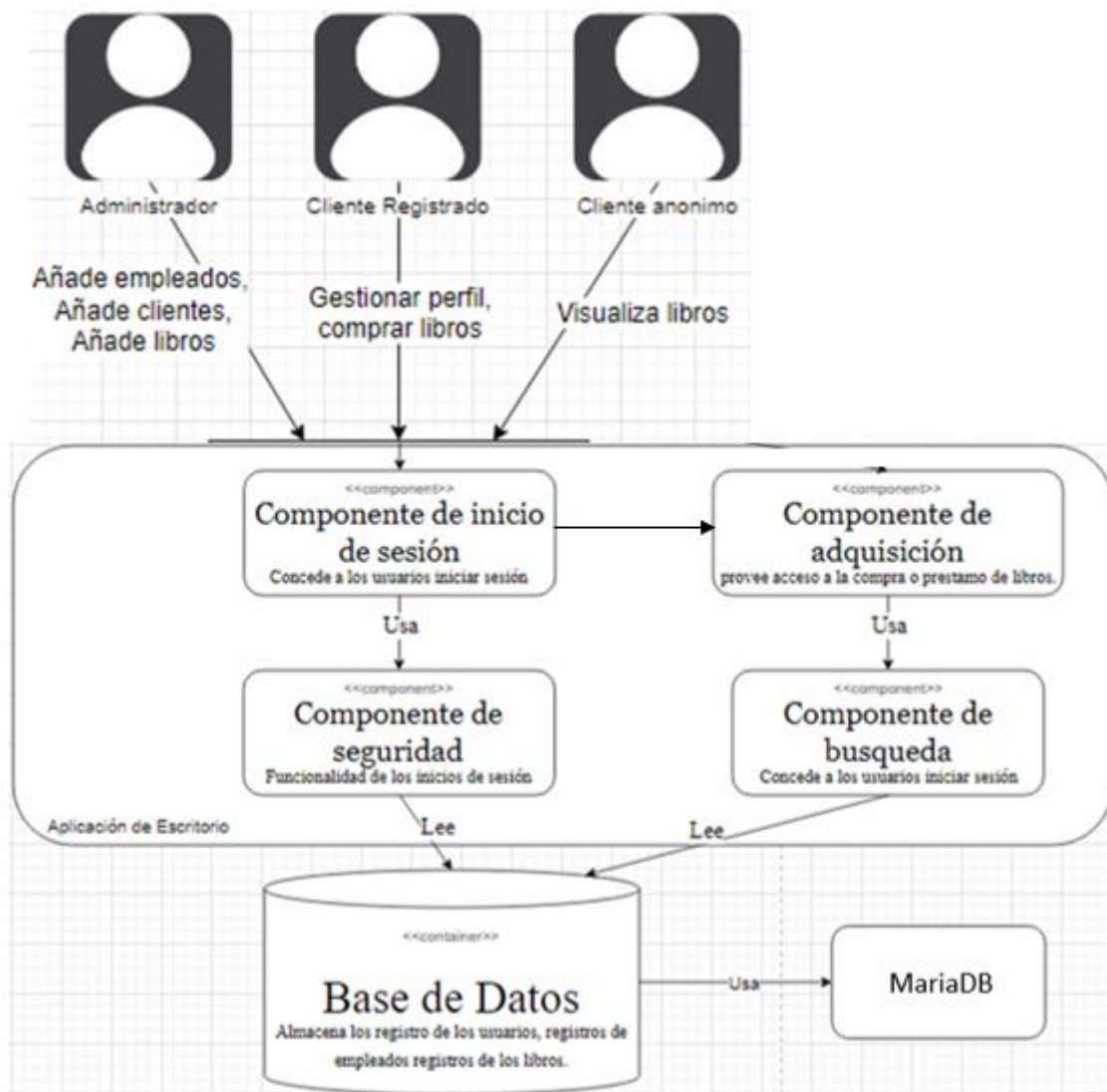


Diagrama Contextual:

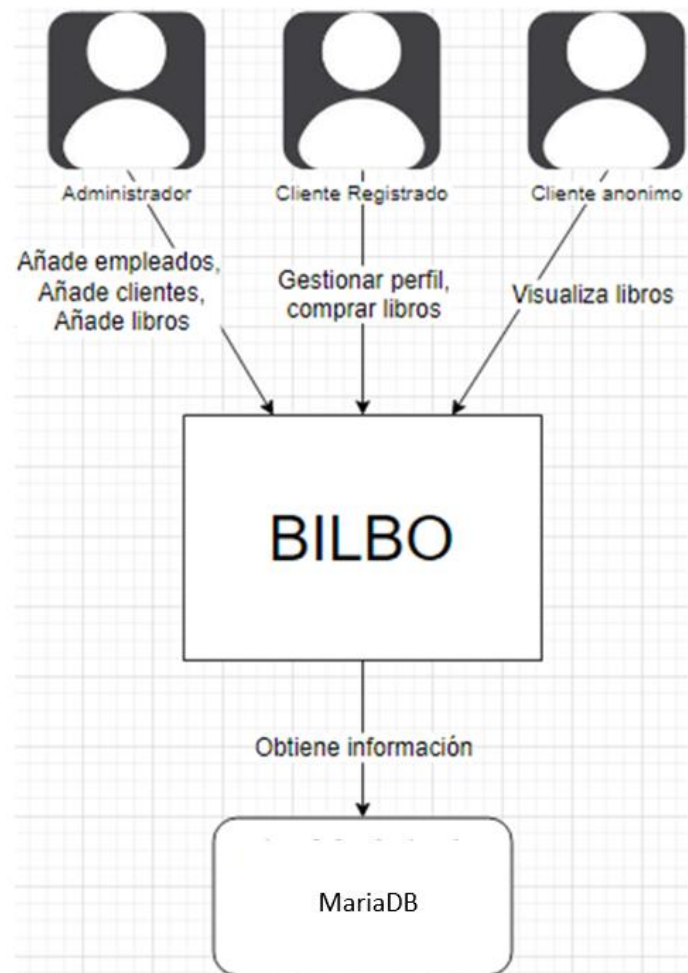
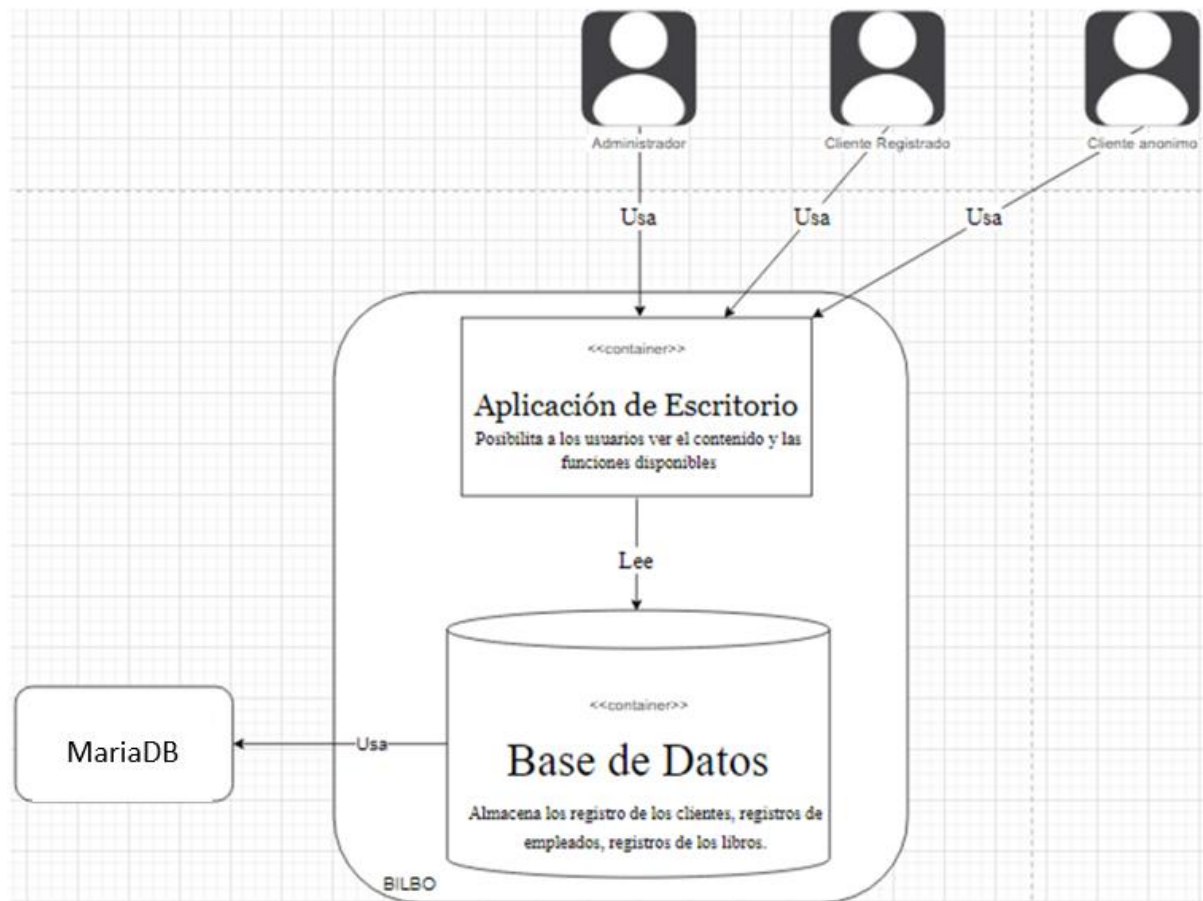
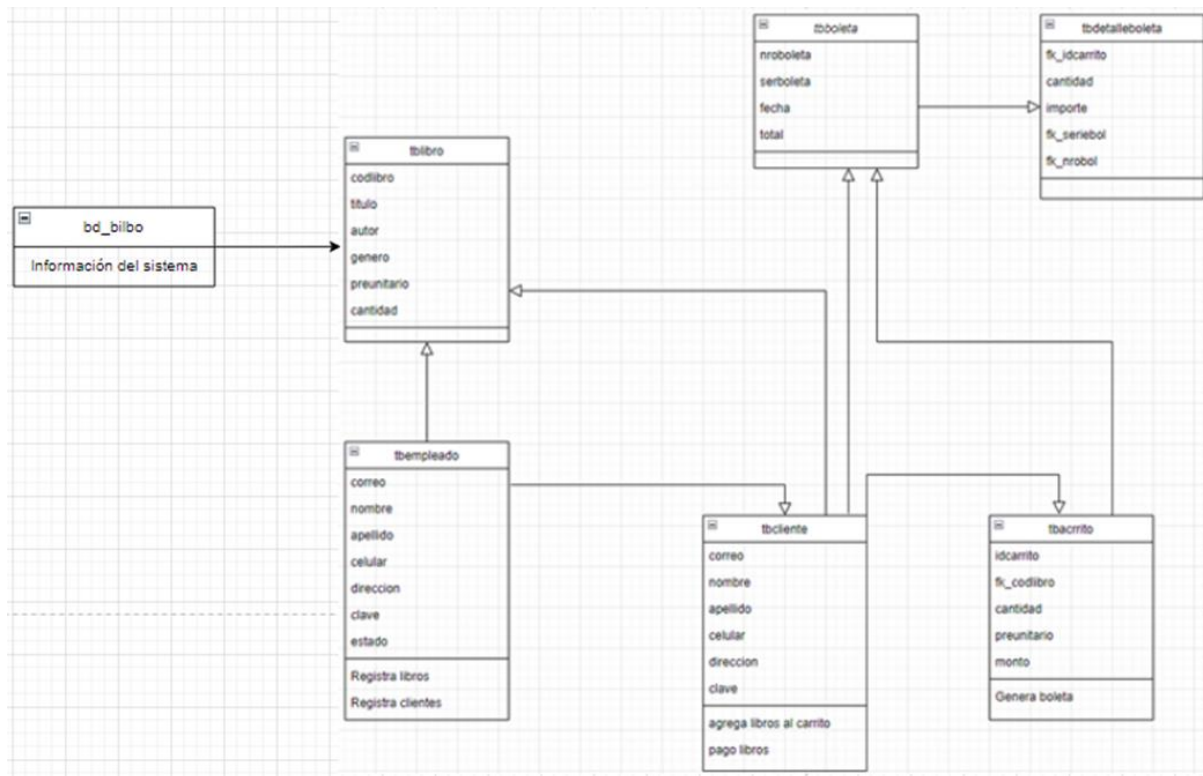


Diagrama de Contenedores:



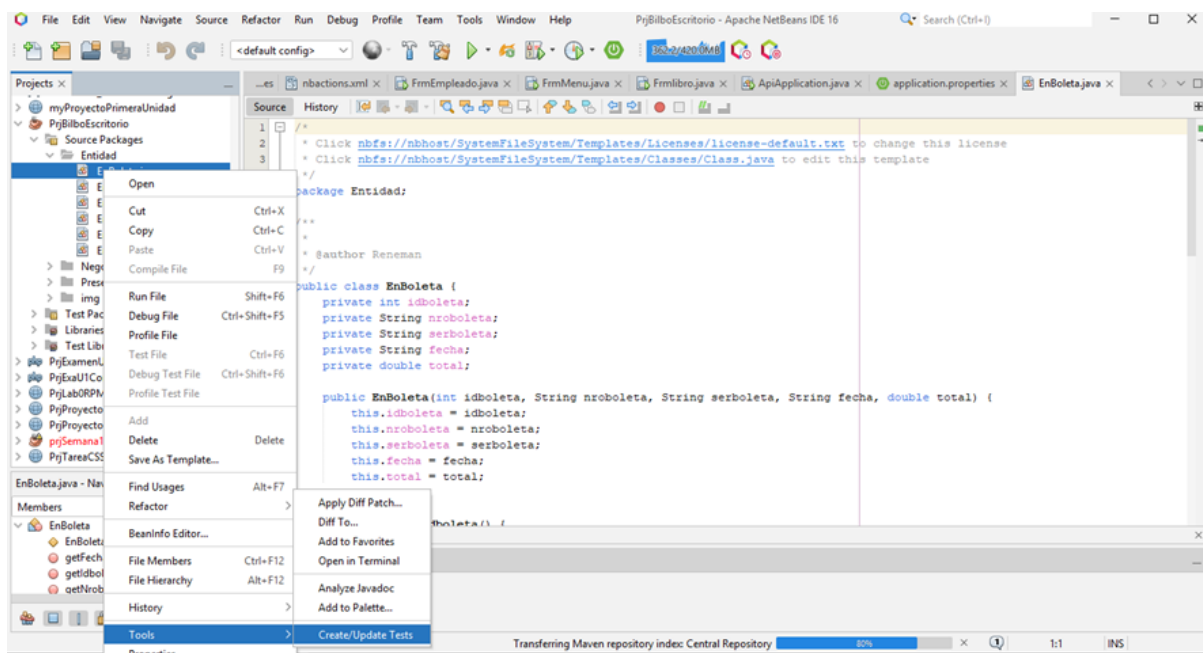
9.2 Diagrama de Clases de la aplicación

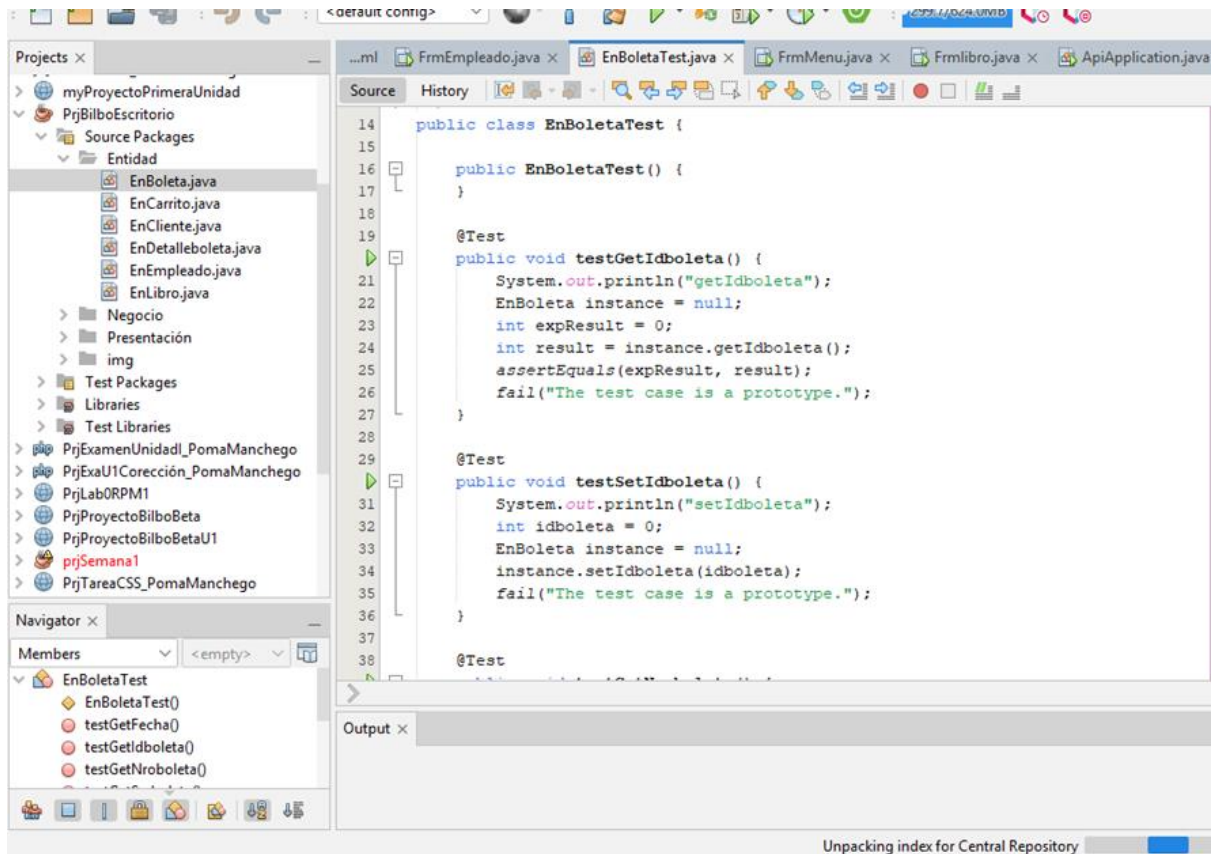


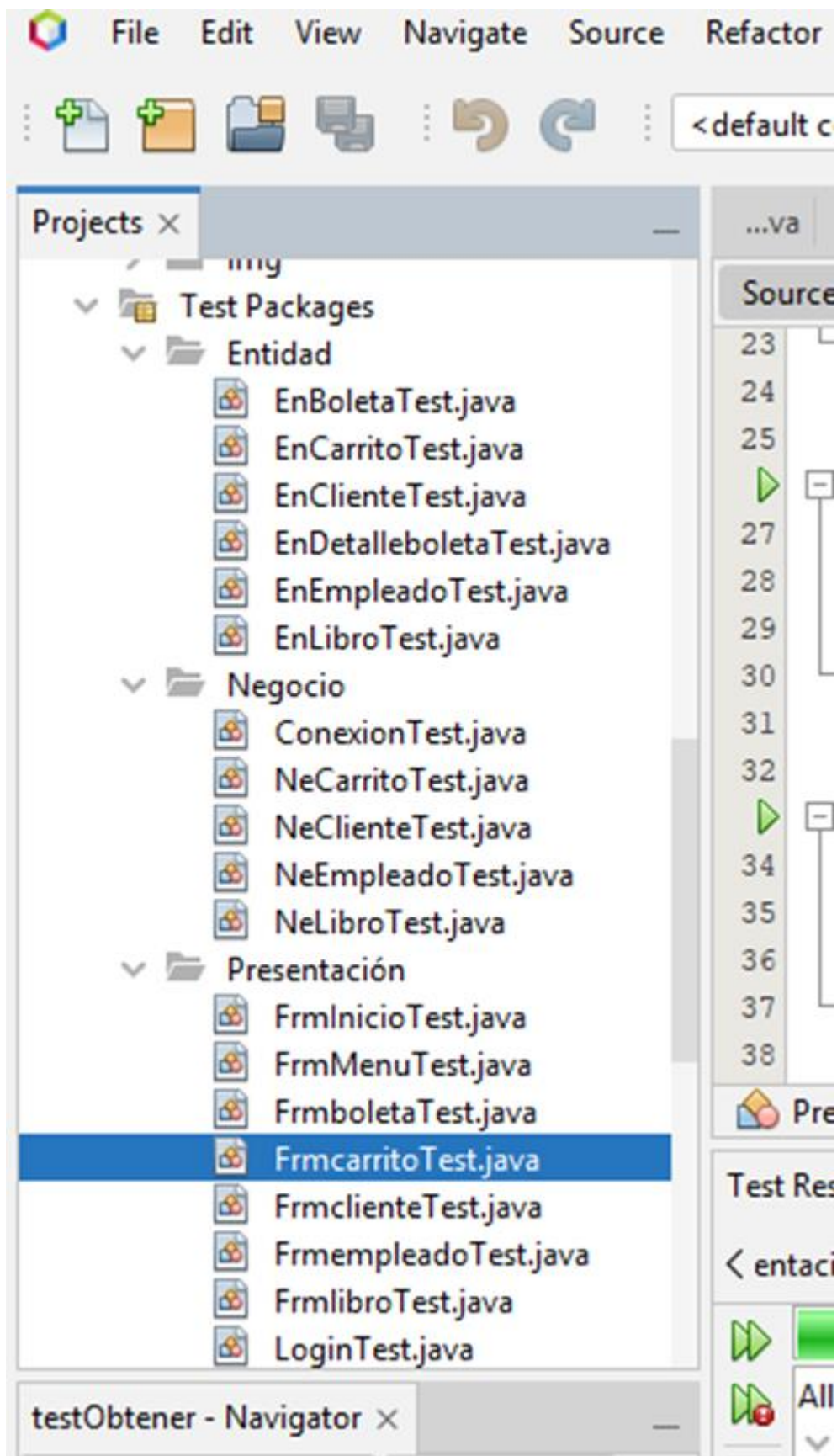
9.3 Métodos de pruebas implementados para cubrir la aplicación

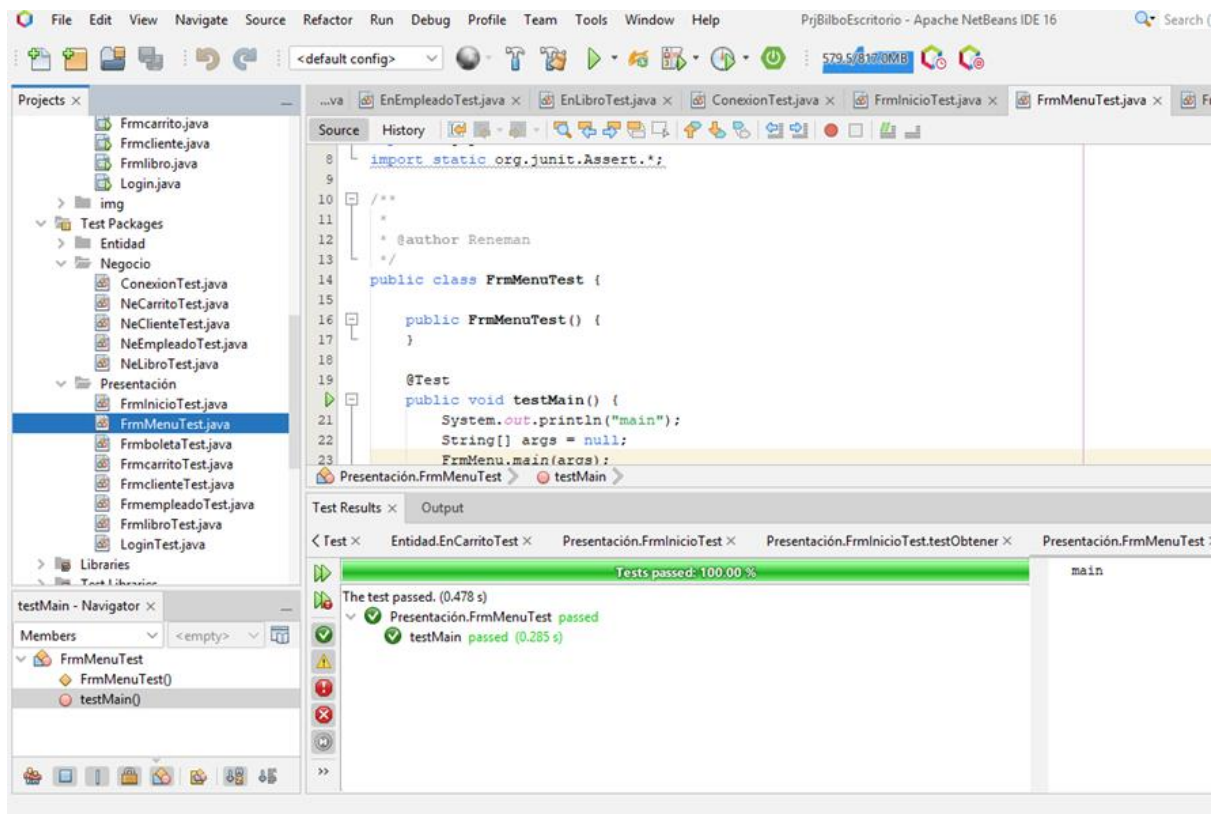
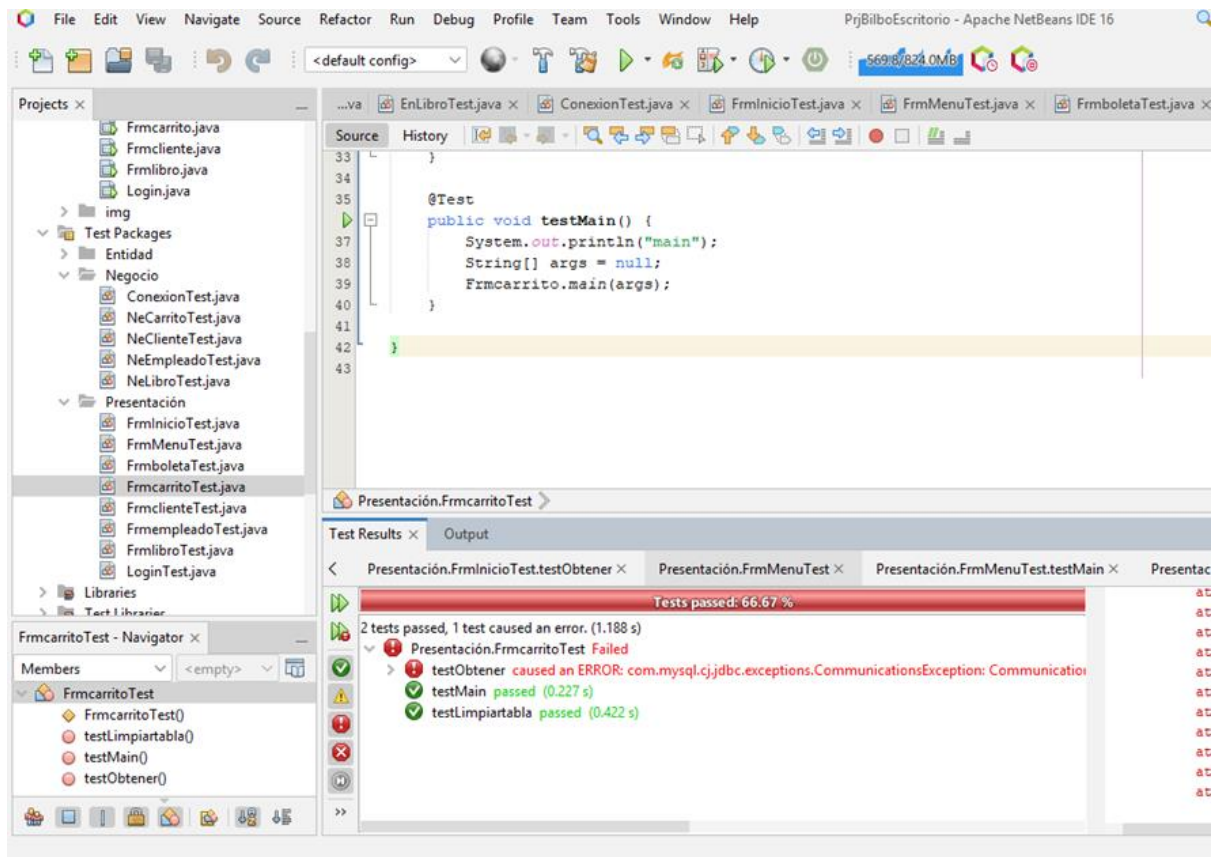
- Reporte de cobertura de pruebas

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









File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects x

- 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

testLimpiarTabla - Navigator x

Members

- FrmInicioTest
 - FrmInicioTest()
 - testLimpiarTabla()
 - testMain()
 - testObtener()

Source

```

20  @Test
21  public void testObtener() throws Exception {
22      System.out.println("obtener");
23      Connection expResult = null;
24      Connection result = FrmInicio.obtener();
25      assertEquals(expResult, result);
26  }
27
28  @Test
29  public void testLimpiarTabla() {
30      System.out.println("limpiarTabla");
31      FrmInicio instance = new FrmInicio();
32      instance.limpiarTabla();
33  }
34
35  @Test

```

Test Results x Output

< Presentación.FrmMenuTest x Presentación.FrmMenuTest.testMain x Presen

The test passed. (1.947 s)

- Presentación.FrmInicioTest passed
- testLimpiarTabla passed (1.751 s)

testLimpiarTabla - Navigator x

Members

- FrmInicioTest
 - FrmInicioTest()
 - testLimpiarTabla()
 - testMain()
 - testObtener()

Unamed:bd_bilbo\tdcarrito - HeidiSQL 11.3.0.6295

Archivo Editar Buscar Consulta Herramientas Ira Ayuda

Host: 127.0.0.1 Base de datos: bd_bilbo

BD_Agenda virtual_V1.sql x BDtechdata.sql x

Tabla: tbcarrito Datos asistencia.sql x

Ordenación Columnas (5/5) Filtro

idcarrito	fk_codlibro	cantidad	preunitario	mont
1	111	2	20	
2	333	3	15	
3	555	2	18	
4	333	2	15	
5	111	3	20	
6	222	3	25	

84 SELECT * FROM "bd_bilbo"."tbcarrito" LIMIT 1000;

bd_bilbo: 6 tables r1: c1 Conectz MariaDB 10.4.1 Activo durante: 01 Hora de Preparado.

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects x

- 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

Navigator x

Members

- FrmMenuTest
 - FrmMenuTest()
 - testMain()

Source

```

20  @Test
21  public void testObtener() throws Exception {
22      System.out.println("obtener");
23      Connection expResult = null;
24      Connection result = FrmInicio.obtener();
25      assertEquals(expResult, result);
26  }
27
28  @Test
29  public void testLimpiarTabla() {
30      System.out.println("limpiarTabla");
31      FrmInicio instance = new FrmInicio();
32      instance.limpiarTabla();
33  }
34
35  @Test

```

Test Results x Output

< Presentación.FrmMenuTest x Presentación.FrmMenuTest.testMain x Presentación.FrmBoletaTest x Presentación.FrmCarritoTest x

The test passed. (0.512 s)

- Presentación.FrmMenuTest passed
- testMain passed (0.277 s)

main

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

279.3/838.0MB

Projects

- FrmboletaTest.java
- FrmmcarritoTest.java
- FrmmclienteTest.java
- FrmmempleadoTest.java
- FrmmmenuTest.java
- FrmmloginTest.java
- img
- Test Packages
- Entidad
- Negocio
 - ConexionTest.java
 - NeCarritoTest.java
 - NeClienteTest.java
 - NeEmpleadoTest.java
 - NeLibroTest.java
- Presentación
 - FrmmInicioTest.java
 - FrmmMenuTest.java
 - FrmboletaTest.java
 - FrmmcarritoTest.java
 - FrmmclienteTest.java
 - FrmmempleadoTest.java
 - FrmmloginTest.java

Source

```

7  import java.sql.Connection;
8  import org.junit.Test;
9  import static org.junit.Assert.*;
10
11  /**
12   *
13   * @author Reneman
14   */
15  public class FrmboletaTest {
16
17      public FrmboletaTest() {
18      }
19
20      @Test
21      public void testObtener() throws Exception {
22          System.out.println("obtener");
23      }
24  }

```

Test Results

Output

Tests passed: 100.00%

The test passed. (0.189 s)

Presentación.FrmboletaTest passed

testObtener passed (0.008 s)

FrmboletaTest - Navigator

Members

- FrmboletaTest
 - FrmboletaTest()
 - testLimpiarTabla()
 - testMain()
 - testObtener()

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Unnamed:bd_bilbo/tbcarrito - HeidiSQL 11.3.0.6295

Archivo Editar Buscar Consulta Herramientas Ira Ayuda

Filtro de bases de datos

Filtro de tablas

Host: 127.0.0.1 Base de datos: bd_bilbo Tabla: tbcarrito

BD_Agenda virtual_V1.sql X BDtechdata.sql X

Datos asistencia.sql* eventoscaldar.sql*

Ordenación Columnas (5/5) Filtro

idcarrito	fk_codibro	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

84 SELECT * FROM "bd_bilbo"."tbcarrito" LIMIT 1000;

bd_bilbo: 6 tables r1: c1 Conectado: MariaDB 10.4.24 Activo durante: 00:01:0 Hora del se: Preparado.

Projects

- NeCarritoTest.java
- NeClienteTest.java
- NeEmpleadoTest.java
- NeLibroTest.java
- Presentación
 - FrmmInicioTest.java
 - FrmmMenuTest.java
 - FrmboletaTest.java
 - FrmmcarritoTest.java
 - FrmmclienteTest.java
 - FrmmempleadoTest.java
 - FrmmloginTest.java
- Libraries
 - mysql-connector-java-8.0.26.jar
 - RSScaleLabel.jar
 - JDK 20 (Default)
- Test Libraries
 - PrjExamenUnidad_PomaManchego
 - PrjExaUICorrección_PomaManchego
 - PrjLabORPM1
 - PrjProyectoBilboBeta
 - PrjProyectoBilboBetaUI

testObtener - Navigator

Members

- FrmmcarritoTest
 - FrmmcarritoTest()
 - testLimpiarTabla()
 - testMain()
 - testObtener()

Source

```

18  @Test
19  public void testObtener()
20  {
21      System.out.println("ob
22      Connection expResult =
23      Connection result = Fa
24      assertEquals(expResult
25  }
26
27  @Test
28  public void testLimpiarTab
29  {
30      System.out.println("Li
31      Frmmcarrito instance =
32      instance.limpiarTabla
33  }
34
35  @Test
36  public void testObtener()
37  {
38      System.out.println("ob
39      Connection expResult =
40      Connection result = Fa
41      assertEquals(expResult
42  }
43
44  @Test
45  public void testLimpiarTab
46  {
47      System.out.println("Li
48      Frmmcarrito instance =
49      instance.limpiarTabla
50  }
51
52  @Test
53  public void testObtener()
54  {
55      System.out.println("ob
56      Connection expResult =
57      Connection result = Fa
58      assertEquals(expResult
59  }
60
61  @Test
62  public void testLimpiarTab
63  {
64      System.out.println("Li
65      Frmmcarrito instance =
66      instance.limpiarTabla
67  }
68
69  @Test
70  public void testObtener()
71  {
72      System.out.println("ob
73      Connection expResult =
74      Connection result = Fa
75      assertEquals(expResult
76  }
77
78  @Test
79  public void testLimpiarTab
80  {
81      System.out.println("Li
82      Frmmcarrito instance =
83      instance.limpiarTabla
84  }
85
86  @Test
87  public void testObtener()
88  {
89      System.out.println("ob
90      Connection expResult =
91      Connection result = Fa
92      assertEquals(expResult
93  }
94
95  @Test
96  public void testLimpiarTab
97  {
98      System.out.println("Li
99      Frmmcarrito instance =
100     instance.limpiarTabla
101  }
102
103  @Test
104  public void testObtener()
105  {
106      System.out.println("ob
107      Connection expResult =
108      Connection result = Fa
109      assertEquals(expResult
110  }
111
112  @Test
113  public void testLimpiarTab
114  {
115      System.out.println("Li
116      Frmmcarrito instance =
117      instance.limpiarTabla
118  }
119
120  @Test
121  public void testObtener()
122  {
123      System.out.println("ob
124      Connection expResult =
125      Connection result = Fa
126      assertEquals(expResult
127  }
128
129  @Test
130  public void testLimpiarTab
131  {
132      System.out.println("Li
133      Frmmcarrito instance =
134      instance.limpiarTabla
135  }
136
137  @Test
138  public void testObtener()
139  {
140      System.out.println("ob
141      Connection expResult =
142      Connection result = Fa
143      assertEquals(expResult
144  }
145
146  @Test
147  public void testLimpiarTab
148  {
149      System.out.println("Li
150      Frmmcarrito instance =
151      instance.limpiarTabla
152  }
153
154  @Test
155  public void testObtener()
156  {
157      System.out.println("ob
158      Connection expResult =
159      Connection result = Fa
160      assertEquals(expResult
161  }
162
163  @Test
164  public void testLimpiarTab
165  {
166      System.out.println("Li
167      Frmmcarrito instance =
168      instance.limpiarTabla
169  }
170
171  @Test
172  public void testObtener()
173  {
174      System.out.println("ob
175      Connection expResult =
176      Connection result = Fa
177      assertEquals(expResult
178  }
179
180  @Test
181  public void testLimpiarTab
182  {
183      System.out.println("Li
184      Frmmcarrito instance =
185      instance.limpiarTabla
186  }
187
188  @Test
189  public void testObtener()
190  {
191      System.out.println("ob
192      Connection expResult =
193      Connection result = Fa
194      assertEquals(expResult
195  }
196
197  @Test
198  public void testLimpiarTab
199  {
200      System.out.println("Li
201      Frmmcarrito instance =
202      instance.limpiarTabla
203  }
204
205  @Test
206  public void testObtener()
207  {
208      System.out.println("ob
209      Connection expResult =
210      Connection result = Fa
211      assertEquals(expResult
212  }
213
214  @Test
215  public void testLimpiarTab
216  {
217      System.out.println("Li
218      Frmmcarrito instance =
219      instance.limpiarTabla
220  }
221
222  @Test
223  public void testObtener()
224  {
225      System.out.println("ob
226      Connection expResult =
227      Connection result = Fa
228      assertEquals(expResult
229  }
230
231  @Test
232  public void testLimpiarTab
233  {
234      System.out.println("Li
235      Frmmcarrito instance =
236      instance.limpiarTabla
237  }
238
239  @Test
240  public void testObtener()
241  {
242      System.out.println("ob
243      Connection expResult =
244      Connection result = Fa
245      assertEquals(expResult
246  }
247
248  @Test
249  public void testLimpiarTab
250  {
251      System.out.println("Li
252      Frmmcarrito instance =
253      instance.limpiarTabla
254  }
255
256  @Test
257  public void testObtener()
258  {
259      System.out.println("ob
260      Connection expResult =
261      Connection result = Fa
262      assertEquals(expResult
263  }
264
265  @Test
266  public void testLimpiarTab
267  {
268      System.out.println("Li
269      Frmmcarrito instance =
270      instance.limpiarTabla
271  }
272
273  @Test
274  public void testObtener()
275  {
276      System.out.println("ob
277      Connection expResult =
278      Connection result = Fa
279      assertEquals(expResult
280  }
281
282  @Test
283  public void testLimpiarTab
284  {
285      System.out.println("Li
286      Frmmcarrito instance =
287      instance.limpiarTabla
288  }
289
290  @Test
291  public void testObtener()
292  {
293      System.out.println("ob
294      Connection expResult =
295      Connection result = Fa
296      assertEquals(expResult
297  }
298
299  @Test
300  public void testLimpiarTab
301  {
302      System.out.println("Li
303      Frmmcarrito instance =
304      instance.limpiarTabla
305  }
306
307  @Test
308  public void testObtener()
309  {
310      System.out.println("ob
311      Connection expResult =
312      Connection result = Fa
313      assertEquals(expResult
314  }
315
316  @Test
317  public void testLimpiarTab
318  {
319      System.out.println("Li
320      Frmmcarrito instance =
321      instance.limpiarTabla
322  }
323
324  @Test
325  public void testObtener()
326  {
327      System.out.println("ob
328      Connection expResult =
329      Connection result = Fa
330      assertEquals(expResult
331  }
332
333  @Test
334  public void testLimpiarTab
335  {
336      System.out.println("Li
337      Frmmcarrito instance =
338      instance.limpiarTabla
339  }
340
341  @Test
342  public void testObtener()
343  {
344      System.out.println("ob
345      Connection expResult =
346      Connection result = Fa
347      assertEquals(expResult
348  }
349
350  @Test
351  public void testLimpiarTab
352  {
353      System.out.println("Li
354      Frmmcarrito instance =
355      instance.limpiarTabla
356  }
357
358  @Test
359  public void testObtener()
360  {
361      System.out.println("ob
362      Connection expResult =
363      Connection result = Fa
364      assertEquals(expResult
365  }
366
367  @Test
368  public void testLimpiarTab
369  {
370      System.out.println("Li
371      Frmmcarrito instance =
372      instance.limpiarTabla
373  }
374
375  @Test
376  public void testObtener()
377  {
378      System.out.println("ob
379      Connection expResult =
380      Connection result = Fa
381      assertEquals(expResult
382  }
383
384  @Test
385  public void testLimpiarTab
386  {
387      System.out.println("Li
388      Frmmcarrito instance =
389      instance.limpiarTabla
390  }
391
392  @Test
393  public void testObtener()
394  {
395      System.out.println("ob
396      Connection expResult =
397      Connection result = Fa
398      assertEquals(expResult
399  }
400
401  @Test
402  public void testLimpiarTab
403  {
404      System.out.println("Li
405      Frmmcarrito instance =
406      instance.limpiarTabla
407  }
408
409  @Test
410  public void testObtener()
411  {
412      System.out.println("ob
413      Connection expResult =
414      Connection result = Fa
415      assertEquals(expResult
416  }
417
418  @Test
419  public void testLimpiarTab
420  {
421      System.out.println("Li
422      Frmmcarrito instance =
423      instance.limpiarTabla
424  }
425
426  @Test
427  public void testObtener()
428  {
429      System.out.println("ob
430      Connection expResult =
431      Connection result = Fa
432      assertEquals(expResult
433  }
434
435  @Test
436  public void testLimpiarTab
437  {
438      System.out.println("Li
439      Frmmcarrito instance =
440      instance.limpiarTabla
441  }
442
443  @Test
444  public void testObtener()
445  {
446      System.out.println("ob
447      Connection expResult =
448      Connection result = Fa
449      assertEquals(expResult
450  }
451
452  @Test
453  public void testLimpiarTab
454  {
455      System.out.println("Li
456      Frmmcarrito instance =
457      instance.limpiarTabla
458  }
459
460  @Test
461  public void testObtener()
462  {
463      System.out.println("ob
464      Connection expResult =
465      Connection result = Fa
466      assertEquals(expResult
467  }
468
469  @Test
470  public void testLimpiarTab
471  {
472      System.out.println("Li
473      Frmmcarrito instance =
474      instance.limpiarTabla
475  }
476
477  @Test
478  public void testObtener()
479  {
480      System.out.println("ob
481      Connection expResult =
482      Connection result = Fa
483      assertEquals(expResult
484  }
485
486  @Test
487  public void testLimpiarTab
488  {
489      System.out.println("Li
490      Frmmcarrito instance =
491      instance.limpiarTabla
492  }
493
494  @Test
495  public void testObtener()
496  {
497      System.out.println("ob
498      Connection expResult =
499      Connection result = Fa
500      assertEquals(expResult
501  }
502
503  @Test
504  public void testLimpiarTab
505  {
506      System.out.println("Li
507      Frmmcarrito instance =
508      instance.limpiarTabla
509  }
510
511  @Test
512  public void testObtener()
513  {
514      System.out.println("ob
515      Connection expResult =
516      Connection result = Fa
517      assertEquals(expResult
518  }
519
520  @Test
521  public void testLimpiarTab
522  {
523      System.out.println("Li
524      Frmmcarrito instance =
525      instance.limpiarTabla
526  }
527
528  @Test
529  public void testObtener()
530  {
531      System.out.println("ob
532      Connection expResult =
533      Connection result = Fa
534      assertEquals(expResult
535  }
536
537  @Test
538  public void testLimpiarTab
539  {
540      System.out.println("Li
541      Frmmcarrito instance =
542      instance.limpiarTabla
543  }
544
545  @Test
546  public void testObtener()
547  {
548      System.out.println("ob
549      Connection expResult =
550      Connection result = Fa
551      assertEquals(expResult
552  }
553
554  @Test
555  public void testLimpiarTab
556  {
557      System.out.println("Li
558      Frmmcarrito instance =
559      instance.limpiarTabla
560  }
561
562  @Test
563  public void testObtener()
564  {
565      System.out.println("ob
566      Connection expResult =
567      Connection result = Fa
568      assertEquals(expResult
569  }
570
571  @Test
572  public void testLimpiarTab
573  {
574      System.out.println("Li
575      Frmmcarrito instance =
576      instance.limpiarTabla
577  }
578
579  @Test
580  public void testObtener()
581  {
582      System.out.println("ob
583      Connection expResult =
584      Connection result = Fa
585      assertEquals(expResult
586  }
587
588  @Test
589  public void testLimpiarTab
590  {
591      System.out.println("Li
592      Frmmcarrito instance =
593      instance.limpiarTabla
594  }
595
596  @Test
597  public void testObtener()
598  {
599      System.out.println("ob
600      Connection expResult =
601      Connection result = Fa
602      assertEquals(expResult
603  }
604
605  @Test
606  public void testLimpiarTab
607  {
608      System.out.println("Li
609      Frmmcarrito instance =
610      instance.limpiarTabla
611  }
612
613  @Test
614  public void testObtener()
615  {
616      System.out.println("ob
617      Connection expResult =
618      Connection result = Fa
619      assertEquals(expResult
620  }
621
622  @Test
623  public void testLimpiarTab
624  {
625      System.out.println("Li
626      Frmmcarrito instance =
627      instance.limpiarTabla
628  }
629
630  @Test
631  public void testObtener()
632  {
633      System.out.println("ob
634      Connection expResult =
635      Connection result = Fa
636      assertEquals(expResult
637  }
638
639  @Test
640  public void testLimpiarTab
641  {
642      System.out.println("Li
643      Frmmcarrito instance =
644      instance.limpiarTabla
645  }
646
647  @Test
648  public void testObtener()
649  {
650      System.out.println("ob
651      Connection expResult =
652      Connection result = Fa
653      assertEquals(expResult
654  }
655
656  @Test
657  public void testLimpiarTab
658  {
659      System.out.println("Li
660      Frmmcarrito instance =
661      instance.limpiarTabla
662  }
663
664  @Test
665  public void testObtener()
666  {
667      System.out.println("ob
668      Connection expResult =
669      Connection result = Fa
670      assertEquals(expResult
671  }
672
673  @Test
674  public void testLimpiarTab
675  {
676      System.out.println("Li
677      Frmmcarrito instance =
678      instance.limpiarTabla
679  }
680
681  @Test
682  public void testObtener()
683  {
684      System.out.println("ob
685      Connection expResult =
686      Connection result = Fa
687      assertEquals(expResult
688  }
689
690  @Test
691  public void testLimpiarTab
692  {
693      System.out.println("Li
694      Frmmcarrito instance =
695      instance.limpiarTabla
696  }
697
698  @Test
699  public void testObtener()
700  {
701      System.out.println("ob
702      Connection expResult =
703      Connection result = Fa
704      assertEquals(expResult
705  }
706
707  @Test
708  public void testLimpiarTab
709  {
710      System.out.println("Li
711      Frmmcarrito instance =
712      instance.limpiarTabla
713  }
714
715  @Test
716  public void testObtener()
717  {
718      System.out.println("ob
719      Connection expResult =
720      Connection result = Fa
721      assertEquals(expResult
722  }
723
724  @Test
725  public void testLimpiarTab
726  {
727      System.out.println("Li
728      Frmmcarrito instance =
729      instance.limpiarTabla
730  }
731
732  @Test
733  public void testObtener()
734  {
735      System.out.println("ob
736      Connection expResult =
737      Connection result = Fa
738      assertEquals(expResult
739  }
740
741  @Test
742  public void testLimpiarTab
743  {
744      System.out.println("Li
745      Frmmcarrito instance =
746      instance.limpiarTabla
747  }
748
749  @Test
750  public void testObtener()
751  {
752      System.out.println("ob
753      Connection expResult =
754      Connection result = Fa
755      assertEquals(expResult
756  }
757
758  @Test
759  public void testLimpiarTab
760  {
761      System.out.println("Li
762      Frmmcarrito instance =
763      instance.limpiarTabla
764  }
765
766  @Test
767  public void testObtener()
768  {
769      System.out.println("ob
770      Connection expResult =
771      Connection result = Fa
772      assertEquals(expResult
773  }
774
775  @Test
776  public void testLimpiarTab
777  {
778      System.out.println("Li
779      Frmmcarrito instance =
780      instance.limpiarTabla
781  }
782
783  @Test
784  public void testObtener()
785  {
786      System.out.println("ob
787      Connection expResult =
788      Connection result = Fa
789      assertEquals(expResult
790  }
791
792  @Test
793  public void testLimpiarTab
794  {
795      System.out.println("Li
796      Frmmcarrito instance =
797      instance.limpiarTabla
798  }
799
800  @Test
801  public void testObtener()
802  {
803      System.out.println("ob
804      Connection expResult =
805      Connection result = Fa
806      assertEquals(expResult
807  }
808
809  @Test
810  public void testLimpiarTab
811  {
812      System.out.println("Li
813      Frmmcarrito instance =
814      instance.limpiarTabla
815  }
816
817  @Test
818  public void testObtener()
819  {
820      System.out.println("ob
821      Connection expResult =
822      Connection result = Fa
823      assertEquals(expResult
824  }
825
826  @Test
827  public void testLimpiarTab
828  {
829      System.out.println("Li
830      Frmmcarrito instance =
831      instance.limpiarTabla
832  }
833
834  @Test
835  public void testObtener()
836  {
837      System.out.println("ob
838      Connection expResult =
839      Connection result = Fa
840      assertEquals(expResult
841  }
842
843  @Test
844  public void testLimpiarTab
845  {
846      System.out.println("Li
847      Frmmcarrito instance =
848      instance.limpiarTabla
849  }
850
851  @Test
852  public void testObtener()
853  {
854      System.out.println("ob
855      Connection expResult =
856      Connection result = Fa
857      assertEquals(expResult
858  }
859
860  @Test
861  public void testLimpiarTab
862  {
863      System.out.println("Li
864      Frmmcarrito instance =
865      instance.limpiarTabla
866  }
867
868  @Test
869  public void testObtener()
870  {
871      System.out.println("ob
872      Connection expResult =
873      Connection result = Fa
874      assertEquals(expResult
875  }
876
877  @Test
878  public void testLimpiarTab
879  {
880      System.out.println("Li
881      Frmmcarrito instance =
882      instance.limpiarTabla
883  }
884
885  @Test
886  public void testObtener()
887  {
888      System.out.println("ob
889      Connection expResult =
890      Connection result = Fa
891      assertEquals(expResult
892  }
893
894  @Test
895  public void testLimpiarTab
896  {
897      System.out.println("Li
898      Frmmcarrito instance =
899      instance.limpiarTabla
900  }
901
902  @Test
903  public void testObtener()
904  {
905      System.out.println("ob
906      Connection expResult =
907      Connection result = Fa
908      assertEquals(expResult
909  }
910
911  @Test
912  public void testLimpiarTab
913  {
914      System.out.println("Li
915      Frmmcarrito instance =
916      instance.limpiarTabla
917  }
918
919  @Test
920  public void testObtener()
921  {
922      System.out.println("ob
923      Connection expResult =
924      Connection result = Fa
925      assertEquals(expResult
926  }
927
928  @Test
929  public void testLimpiarTab
930  {
931      System.out.println("Li
932      Frmmcarrito instance =
933      instance.limpiarTabla
934  }
935
936  @Test
937  public void testObtener()
938  {
939      System.out.println("ob
940      Connection expResult =
941      Connection result = Fa
942      assertEquals(expResult
943  }
944
945  @Test
946  public void testLimpiarTab
947  {
948      System.out.println("Li
949      Frmmcarrito instance =
950      instance.limpiarTabla
951  }
952
953  @Test
954  public void testObtener()
955  {
956      System.out.println("ob
957      Connection expResult =
958      Connection result = Fa
959      assertEquals(expResult
960  }
961
962  @Test
963  public void testLimpiarTab
964  {
965      System.out.println("Li
966      Frmmcarrito instance =
967      instance.limpiarTabla
968  }
969
970  @Test
971  public void testObtener()
972  {
973      System.out.println("ob
974      Connection expResult =
975      Connection result = Fa
976      assertEquals(expResult
977  }
978
979  @Test
980  public void testLimpiarTab
981  {
982      System.out.println("Li
983      Frmmcarrito instance =
984      instance.limpiarTabla
985  }
986
987  @Test
988  public void testObtener()
989  {
990      System.out.println("ob
991      Connection expResult =
992      Connection result = Fa
993      assertEquals(expResult
994  }
995
996  @Test
997  public void testLimpiarTab
998  {
999      System.out.println("Li
1000     Frmmcarrito instance =
1001     instance.limpiarTabla
1002  }
1003
1004  @Test
1005  public void testObtener()
1006  {
1007      System.out.println("ob
1008      Connection expResult =
1009      Connection result = Fa
1010      assertEquals(expResult
1011  }
1012
1013  @Test
1014  public void testLimpiarTab
1015  {
1016      System.out.println("Li
1017      Frmmcarrito instance =
1018      instance.limpiarTabla
1019  }
1020
1021  @Test
1022  public void testObtener()
1023  {
1024      System.out.println("ob
1025      Connection expResult =
1026      Connection result = Fa
1027      assertEquals(expResult
1028  }
1029
1030  @Test
1031  public void testLimpiarTab
1032  {
1033      System.out.println("Li
1034      Frmmcarrito instance =
1035      instance.limpiarTabla
1036  }
1037
1038  @Test
1039  public void testObtener()
1040  {
1041      System.out.println("ob
1042      Connection expResult =
1043      Connection result = Fa
1044      assertEquals(expResult
1045  }
1046
1047  @Test
1048  public void testLimpiarTab
1049  {
1050      System.out.println("Li
1051      Frmmcarrito instance =
1052      instance.limpiarTabla
1053  }
1054
1055  @Test
1056  public void testObtener()
1057  {
1058      System.out.println("ob
1059      Connection expResult =
1060      Connection result = Fa
1061      assertEquals(expResult
1062  }
1063
1064  @Test
1065  public void testLimpiarTab
1066  {
1067      System.out.println("Li
1068      Frmmcarrito instance =
1069      instance.limpiarTabla
1070  }
1071
1072  @Test
1073  public void testObtener()
1074  {
1075      System.out.println("ob
1076      Connection expResult =
1077      Connection result = Fa
1078      assertEquals(expResult
1079  }
1080
1081  @Test
1082  public void testLimpiarTab
1083  {
1084      System.out.println("Li
1085      Frmmcarrito instance =
1086      instance.limpiarTabla
1087  }
1088
1089  @Test
1090  public void testObtener()
1091  {
1092      System.out.println("ob
1093      Connection expResult =
1094      Connection result = Fa
1095      assertEquals(expResult
1096  }
1097
1098  @Test
1099  public void testLimpiarTab
1100  {
1101      System.out.println("Li
1102      Frmmcarrito instance =
1103      instance.limpiarTabla
1104  }
1105
1106  @Test
1107  public void testObtener()
1108  {
1109      System.out.println("ob
1110      Connection expResult =
1111      Connection result = Fa
1112      assertEquals(expResult
1113  }
1114
1115  @Test
1116  public void testLimpiarTab
1117  {
1118      System.out.println("Li
1119      Frmmcarrito instance =
1120      instance.limpiarTabla
1121  }
1122
1123  @Test
1124  public void testObtener()
1125  {
1126      System.out.println("ob
1127      Connection expResult =
1128      Connection result = Fa
1129      assertEquals(expResult
1130  }
1131
1132  @Test
1133  public void testLimpiarTab
1134  {
1135      System.out.println("Li
1136      Frmmcarrito instance =
1137      instance.limpiarTabla
1138  }
1139
1140  @Test
1141  public void testObtener()
1142  {
1143      System.out.println("ob
1144      Connection expResult =
1145      Connection result = Fa
1146      assertEquals(expResult
1147  }
1148
1149  @Test
1150  public void testLimpiarTab
1151  {
1152      System.out.println("Li
1153      Frmmcarrito instance =
1154      instance.limpiarTabla
1155  }
1156
1157  @Test
1158  public void testObtener()
1159  {
1160      System.out.println("ob
1161      Connection expResult =
1162      Connection result = Fa
1163      assertEquals(expResult
1164  }
1165
1166  @Test
1167  public void testLimpiarTab
1168  {
1169      System.out.println("Li
1170      Frmmcarrito instance =
1171      instance.limpiarTabla
1172  }
1173
1174  @Test
1175  public void testObtener()
1176  {
1177      System.out.println("ob
1178      Connection expResult =
1179      Connection result = Fa
1180      assertEquals(expResult
1181  }
1182
1183  @Test
1184  public void testLimpiarTab
1185  {
1186      System.out.println("Li
1187      Frmmcarrito instance =
1188      instance.limpiarTabla
1189  }
1190
1191  @Test
1192  public void testObtener()
1193  {
1194      System.out.println("ob
1195      Connection expResult =
1196      Connection result = Fa
1197      assertEquals(expResult
1198  }
1199
1200  @Test
1201  public void testLimpiarTab
1202  {
1203      System.out.println("Li
1204      Frmmcarrito instance =
1205      instance.limpiarTabla
1206  }
1207
1208  @Test
1209  public void testObtener()
1210  {
1211      System.out.println("ob
1212      Connection expResult =
1213      Connection result = Fa
1214      assertEquals(expResult
1215  }
1216
1217  @Test
1218  public void testLimpiarTab
1219  {
1220      System.out.println("Li
1221      Frmmcarrito instance =
1222      instance.limpiarTabla
1223  }
1224
1225  @Test
1226  public void testObtener()
1227  {
1228      System.out.println("ob
1229      Connection expResult =
1230      Connection result = Fa
1231      assertEquals(expResult
1232  }
1233
1234  @Test
1235  public void testLimpiarTab
1236  {
1237      System.out.println("Li
1238      Frmmcarrito instance =
1239      instance.limpiarTabla
1240  }
1241
1242  @Test
1243  public void testObtener()
1244  {
1245      System.out.println("ob
1246      Connection expResult =
1247      Connection result = Fa
1248      assertEquals(expResult
1249  }
1250
1251  @Test
1252  public void testLimpiarTab
1253  {
1254      System.out.println("Li
1255      Frmmcarrito instance =
1256      instance.limpiarTabla
1257  }
1258
1259  @Test
1260  public void testObtener()
1261  {
1262      System.out.println("ob
1263      Connection expResult =
1264      Connection result = Fa
1265      assertEquals(expResult
1266  }
1267
1268  @Test
1269  public void testLimpiarTab
1270  {
1271      System.out.println("Li
1272      Frmmcarrito instance =
1273      instance.limpiarTabla
1274  }
1275
1276  @Test
1277  public void testObtener()
1278  {
1279      System.out.println("ob
1280      Connection expResult =
1281      Connection result = Fa
1282      assertEquals(expResult
1283  }
1284
1285  @Test
1286  public void testLimpiarTab
1287  {
1288      System.out.println("Li
1289      Frmmcarrito instance =
1290      instance.limpiarTabla
1291  }
1292
1293  @Test
1294  public void testObtener()
1295  {
1296      System.out.println("ob
1297      Connection expResult =
1298      Connection result = Fa
1299      assertEquals(expResult
1300  }
1301
1302  @Test
1303  public void testLimpiarTab
1304  {
1305      System.out.println("Li
1306      Frmmcarrito instance =
1307      instance.limpiarTabla
1308  }
1309
1310  @Test
1311  public void testObtener()
1312  {
1313      System.out.println("ob
1314      Connection expResult =
1315      Connection result = Fa
1316      assertEquals(expResult
1317  }
1318
1319  @Test
1320  public void testLimpiarTab
1321  {
1322      System.out.println("Li
1323      Frmmcarrito instance =
1324      instance.limpiarTabla
1325  }
1326
1327  @Test
1328  public void testObtener()
1329  {
1330      System.out.println("ob
1331      Connection expResult =
1332      Connection
```

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

291/9/838 OMB

Projects

- 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

testLimpiartabla - Navigator

Members

- FrmclienteTest
 - testObtener()
 - testLimpiartabla()
 - testMain()
 - testObtener()

Source

```

7 import java.sql.Connection;
8 import org.junit.Test;
9 import static org.junit.Assert.*;
10
11 /**
12  *
13  * @author Reneman
14  */
15 public class FrmclienteTest {
16
17     public FrmclienteTest() {
18
19     }
20
21     @Test
22     public void testObtener() throws Exception {
23         System.out.println("obtener");
24     }
25 }

```

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)

PrjBilboEscritorio (test)

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help PrjBilboEscritorio - HeidiSQL 11.3.0.6295

Archivo Editar Buscar Consulta Herramientas Ira Ayuda

Host: 127.0.0.1 Base de datos: bd_bilbo

Filter: Filtro de bases de datos Filtro de tablas

eventoscalendar.sql

BD_Agenda_virtual_V1.sql

BDtechdata.sql

Tabla: tbcarrito Datos asistencia.sql

Ordenación Columnas (5/5) Filtro

idcarrito	fk_codlibro	cantidad	preunitario	mont
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;

r1: c1 Conectado MariaDB 10.4.2 Activo durante: 0:00 Hora de: Preparado.

Projects

- 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

testObtener - Navigator

Members

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

Source

```

23 }
24
25 @Test
26 public void testLimpiartabla() {
27     System.out.println("limpiartabla");
28     Frmcarrito instance = new Frmcarrito();
29     instance.limpiartabla();
30 }
31
32 @Test
33 public void testMain() {
34     System.out.println("main");
35     String[] args = null;
36     Frmcarrito.main(args);
37 }
38 }

```

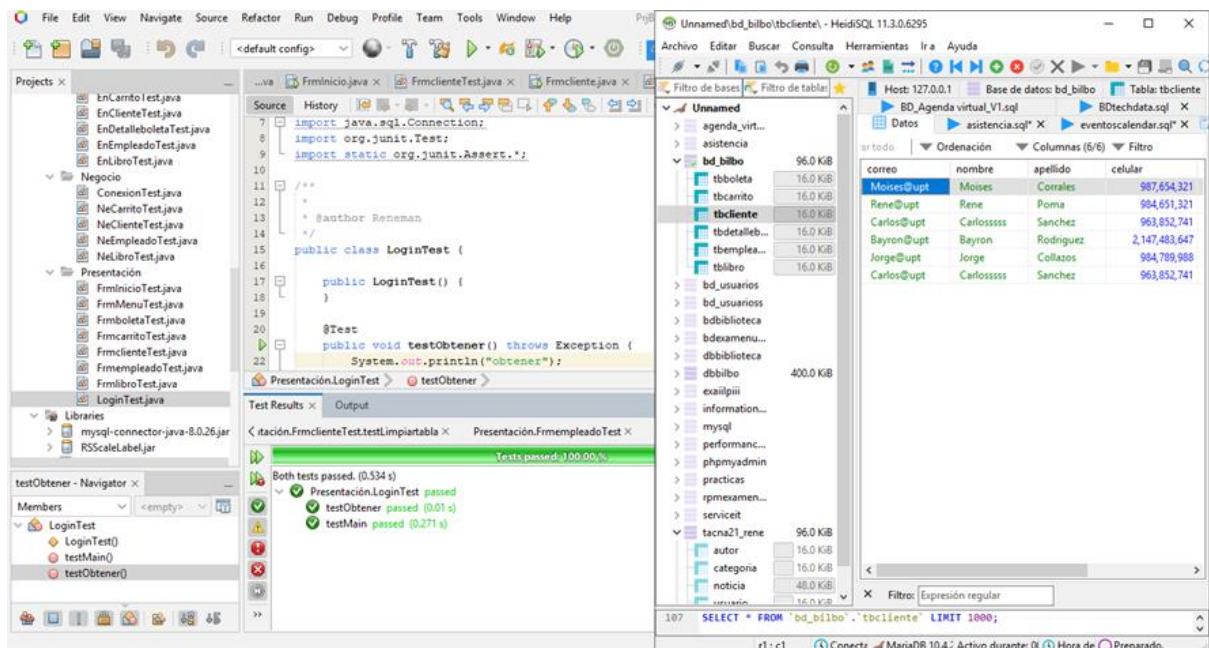
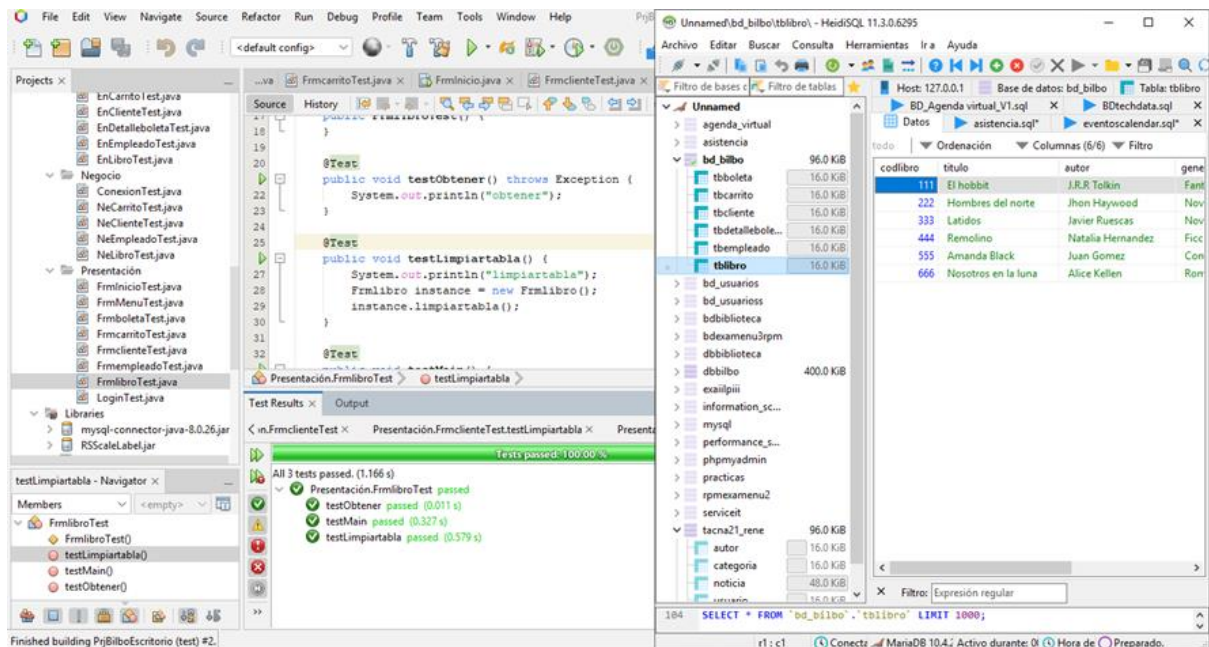
Test Results

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)



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

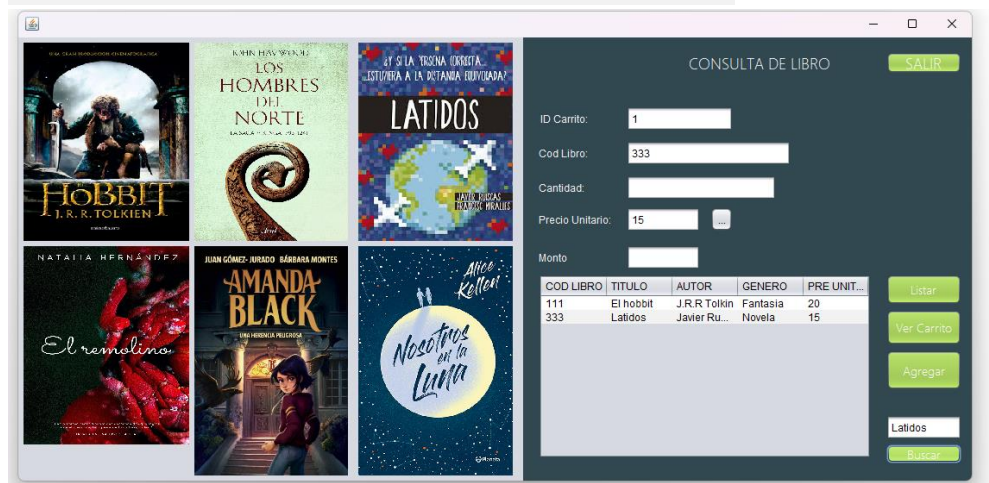
b) Pruebas de aceptación basadas en Desarrollo Guiado por el Comportamiento una por cada caso de uso o historia de usuario.

- Buscar libro

```

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(rowData: dato);
                }
                tblibro.setModel(dataModel: model);
            } catch (SQLException ex) {
                System.out.println(":" + ex.getMessage());
            }
        } else {
            JOptionPane.showMessageDialog(parentComponent: null, message: "No existe un libro asi");
        }
    } catch (Exception e) {
        System.err.println(":" + e);
    }
}

```

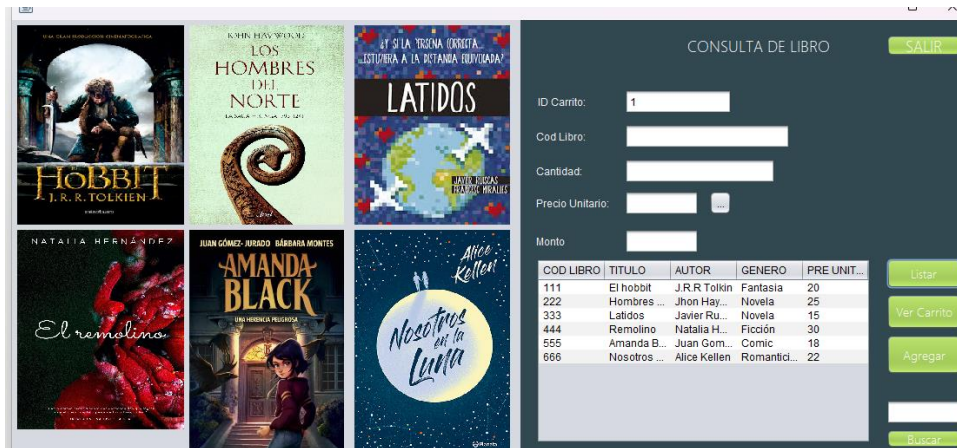


- Comprar stock del libro

```

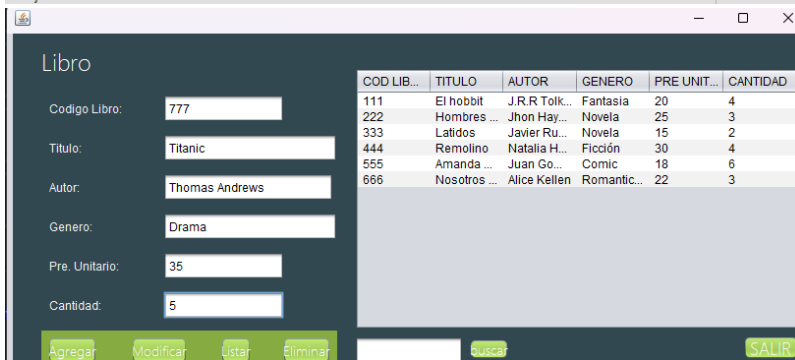
private void btnlistarActionPerformed(java.awt.event.ActionEvent evt) {
    ResultSet rs;
    limpiarcajas();
    limpiartabla();
    NeLibro objNcli=new NeLibro();
    String[] dato=new String[6];
    try {
        rs= objNcli.MtdListar();
        while(rs.next())
        {
            dato[0]=rs.getString("codlibro");
            dato[1]=rs.getString("titulo");
            dato[2]=rs.getString("autor");
            dato[3]=rs.getString("genero");
            dato[4]=rs.getString("preunitario");
            model.addRow(rowData: dato);
        }
        tblibro.setModel(dataModel: 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) {  
  
    Frmboleto frm = new Frmboleto();  
    frm.show();  
    this.hide();  
}
```

Carrito

ID Carrito:

Cod Libro:

Cantidad:

Precio Unitario:

Monto:

ID CARRI...	COD LIB...	CANTIDAD	PRE UNIT...	MONTO
1	111	2	20	40.0
2	333	3	15	45.0
3	555	2	18	36.0

- Realizar venta

```

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

    try {
        con = obtener();
        ps = con.prepareStatement("insert into tbboleto(nroboleto,serboleto,fecha,total) values (?, ?, ?, ?)");
        ps.setString(1, txtidcarro.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(parentComponent, null, "Impreso con exito");
            btnsalir.setVisible(true);
            ps = con.prepareStatement("insert into tbdetalleboleto(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, txtidcarro.getText());
            int res2 = ps.executeUpdate();
            limpiarcajas();
            limpiartabla();
        } else {
            JOptionPane.showMessageDialog(parentComponent, 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](#)