1. Deben crear un proyecto maven con los siguientes parámetros:
   a. Creamos el proyecto

```
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS> mvn archetype:generate
```

   b. Definimos los parametros asignados

```
Define value for property 'groupId': edu.eci.cvds
Define value for property 'artifactId': Library
Define value for property 'version' 1.0-SNAPSHOT:
Define value for property 'package' edu.eci.cvds: edu.eci.cvds.tdd
Confirm properties configuration:
groupId: edu.eci.cvds
artifactId: Library
version: 1.0-SNAPSHOT
package: edu.eci.cvds.tdd
 Y: N
Define value for property 'groupId': edu.eci.cvds
Define value for property 'artifactId': maven-archetype-quickstart
Define value for property 'version' 1.0-SNAPSHOT:
Define value for property 'package' edu.eci.cvds: edu.eci.cvds.tdd
Confirm properties configuration:
groupId: edu.eci.cvds
artifactId: maven-archetype-quickstart
version: 1.0-SNAPSHOT
package: edu.eci.cvds.tdd
 Y: Y
```

```
[INFO] -------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] -------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: C:\Users\Esteban Aguilera
[INFO] Parameter: package, Value: edu.eci.cvds.tdd
[INFO] Parameter: groupId, Value: edu.eci.cvds
[INFO] Parameter: artifactId, Value: maven-archetype-quickstart
[INFO] Parameter: packageName, Value: edu.eci.cvds.tdd
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\Esteban Aguilera\maven-archetype-quickstart
[INFO] -------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] -------------------------------------------------------------------------
[INFO] Total time:  01:24 min
[INFO] Finished at: 2025-02-06T13:40:19-05:00
[INFO] -------------------------------------------------------------------------
PS C:\Users\Esteban Aguilera> |
```

AGREGAR DEPENDENCIA JUNIT 5

1. Buscar en maven central la dependencia de JUnit5 en su versión más reciente.
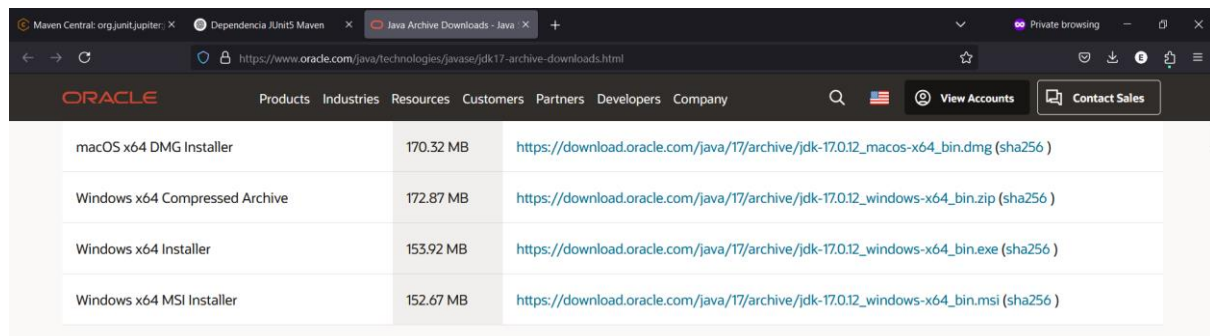   a. Buscamos en Maven central la dependencia mas cercana a Junit5

2. Edite el archivo pom.xml del proyecto para agregar la dependencia.

a. Agregamos la dependencia en el pom.xml

3. Verifique que la versión de java sea la 17
   a. Instalamos la versión 17 de java

b. Miramos que versiones tenemos instaladas de java

c. Cambiamos la versión de java de la 21 a la 17

4. Compile el proyecto para validar que todo este bien
   a. Compilamos con mvn package

5. AGREGAR ESQUELETO DEL PROYECTO
   a. Cree los siguientes paquetes dentro de edu.eci.cvds.tdd

```
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd> mkdir book

    Directorio: C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         6/02/2025   2:09 p. m.              book

PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd> mkdir loan

    Directorio: C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         6/02/2025   2:10 p. m.              loan

PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd> mkdir user

    Directorio: C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd
```

```
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library> tree
Listado de rutas de carpetas para el volumen Acer
El número de serie del volumen es C8DF-2DA4
C:.
├───src
│   └───main
│       └───java
│           └───edu
│               └───eci
│                   └───cvds
│                       └───tdd
│                           └───library
│                               ├───book
│                               ├───loan
│                               └───user
```

6. AGREGAR CLASES
   a. En el paquete edu.eci.cvds.tdd.library.book cree la siguiente clase:
      i. Creamos la clase Book

```
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library> cd "C:\Users\Esteban Aguilera\Documents\Lab-03-CV
src\main\java\edu\eci\cvds\tdd\library\book"
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd\library\book> notep
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd\library\book> notep
a
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd\library\book>
```

      ii. Agregamos el código

```java
package main.java.edu.eci.cvds.tdd.library.book;

public class Book {
    private final String tittle;
    private final String author;
    private final String isbn;

    public Book(String tittle, String author, String isbn) {
        this.tittle = tittle;
        this.author = author;
        this.isbn = isbn;
    }

    public String getTittle() {
        return tittle;
    }

    public String getAuthor() {
        return author;
    }

    public String getIsbn() {
        return isbn;
    }

    @Override
    public boolean equals(Object obj) {
        return isbn.equals(((Book)obj).isbn);
    }
}
```

b. A continuación en el paquete edu.eci.cvds.tdd.library.user cree la siguiente clase:

   i. Creamos la clase User


```
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd\library\book> cd ..
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd\library> cd user
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd\library\user> notepa
d
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\main\java\edu\eci\cvds\tdd\library\user>
```

   ii. Agregamos el código


```java
package main.java.edu.eci.cvds.tdd.library.user;

public class User {
    private String name;
    private String id;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }
}
```

c. En el paquete edu.eci.cvds.tdd.library.loan se deben crear las clases Loan y el enum LoanStatus:

   i. Creamos las clases Loan y LoanStatus
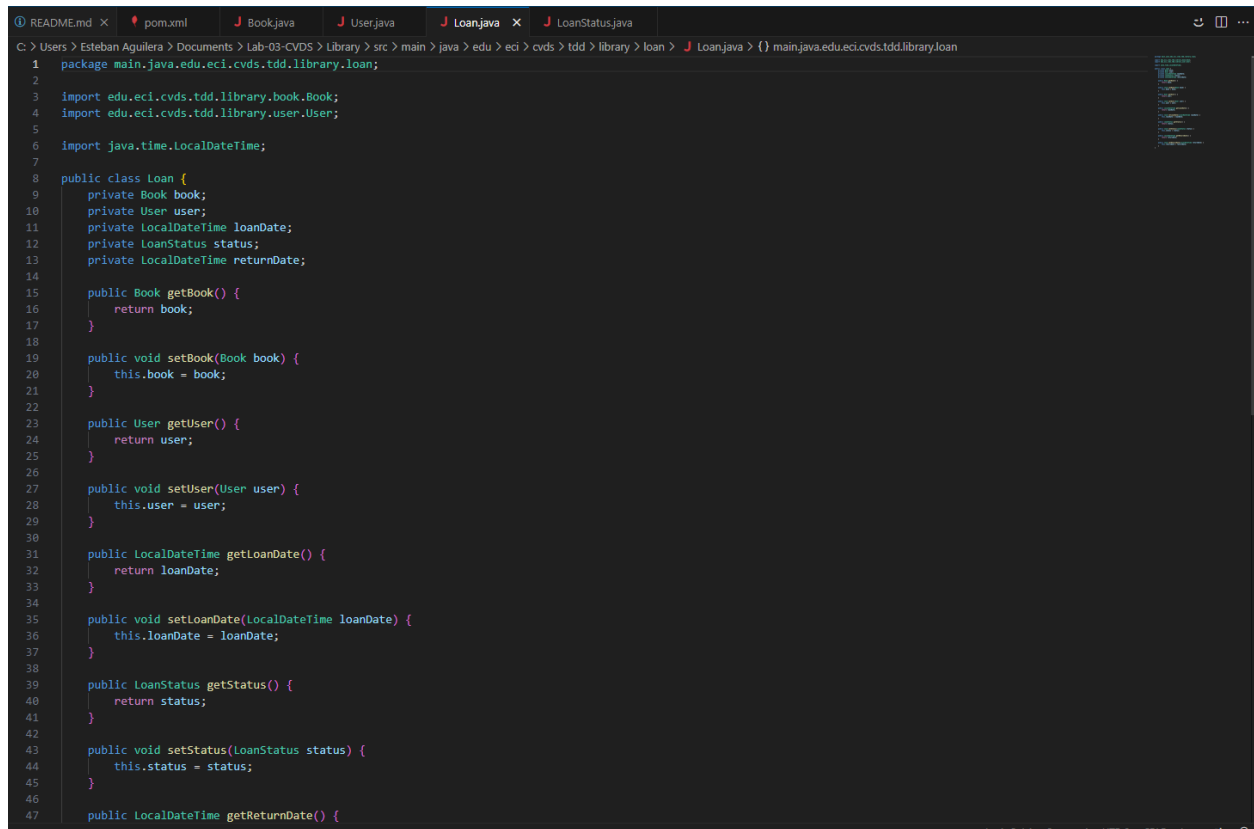
ii. Agregamos el código de las dos clases


```java
package main.java.edu.eci.cvds.tdd.library.loan;

import edu.eci.cvds.tdd.library.book.Book;
import edu.eci.cvds.tdd.library.user.User;

import java.time.LocalDateTime;

public class Loan {
    private Book book;
    private User user;
    private LocalDateTime loanDate;
    private LoanStatus status;
    private LocalDateTime returnDate;

    public Book getBook() {
        return book;
    }

    public void setBook(Book book) {
        this.book = book;
    }

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public LocalDateTime getLoanDate() {
        return loanDate;
    }

    public void setLoanDate(LocalDateTime loanDate) {
        this.loanDate = loanDate;
    }

    public LoanStatus getStatus() {
        return status;
    }

    public void setStatus(LoanStatus status) {
        this.status = status;
    }

    public LocalDateTime getReturnDate() {
```



```java
package main.java.edu.eci.cvds.tdd.library.loan;

public enum LoanStatus {
    ACTIVE, RETURNED
}
```

d. por último se debe crear la siguiente clase en el paquete edu.eci.cvds.tdd.library

i. Creamos la clase library

ii. Agregamos el código

e. Para validar que la estructura del proyecto está bien se debe compilar usando el comando package.

i. Compilamos el proyecto con mvn-package

7. PRUEBAS UNITARIAS Y TDD
   a. Creamos la clase de pruebas en la carpeta de test


```
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\test\java\edu\eci\cvds\tdd> mkdir Library


    Directorio: C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\test\java\edu\eci\cvds\tdd


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         10/02/2025   7:29 p. m.            Library


PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\test\java\edu\eci\cvds\tdd> cd Library
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\src\test\java\edu\eci\cvds\tdd\Library> notepad LibraryTest.j
```
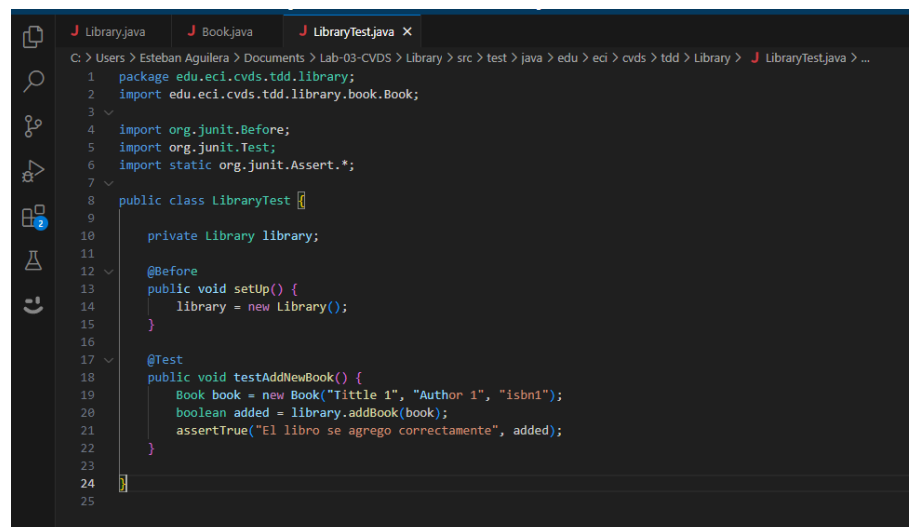
   b. TEST ADDBOOK
      i. Pruebas
         1. Añadir un nuevo libro


```java
package edu.eci.cvds.tdd.library;
import edu.eci.cvds.tdd.library.book.Book;

import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;

public class LibraryTest {

    private Library library;

    @Before
    public void setUp() {
        library = new Library();
    }

    @Test
    public void testAddNewBook() {
        Book book = new Book("Tittle 1", "Author 1", "isbn1");
        boolean added = library.addBook(book);
        assertTrue("El libro se agrego correctamente", added);
    }

}
```



```
[INFO]
[INFO] Results:
[INFO]
[ERROR] Failures:
[ERROR]   LibraryTest.testAddNewBook:21 El libro se agrego correctamente
[INFO]
[ERROR] Tests run: 2, Failures: 1, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.106 s
[INFO] Finished at: 2025-02-10T20:02:16-05:00
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:3.2.5:test (default-test) on project maven
-archetype-quickstart: There are test failures.
[ERROR]
[ERROR] Please refer to C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\target\surefire-reports for the individu
al test results.
[ERROR] Please refer to dump files (if any exist) [date].dump, [date]-jvmRun[N].dump and [date].dumpstream.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
PS C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library>
```

         2. Añadir un libro null

C: > Users > Juan > Desktop > CVDS > Lab-03-CVDS > Library > src > test > java > edu > eci > cvds > tdd > library > J LibraryTest.java > ...

```java
1   package edu.eci.cvds.tdd.library;
2   import edu.eci.cvds.tdd.library.book.Book;
3
4   import org.junit.Before;
5   import org.junit.Test;
6   import static org.junit.Assert.*;
7
8   public class LibraryTest {
9
10      private Library library;
11
12      @Before
13      public void setUp() {
14          library = new Library();
15      }
16
17      @Test
18      public void testAddNewBook() {
19          Book book = new Book("Tittle 1", "Author 1", "isbn1");
20          boolean added = library.addBook(book);
21          assertTrue("El libro se agrego correctamente", added);
22      }
23
24      @Test(expected = IllegalArgumentException.class)
25      public void testAddBookShouldThrowExceptionWhenNull() {
26          library.addBook(null);
27      }
28  }
29
```

```
[INFO]
[INFO] Results:
[INFO]
[ERROR] Failures:
[ERROR]   LibraryTest.testAddBookShouldThrowExceptionWhenNull Expected exception: java.lang.IllegalArgumentException
[ERROR]   LibraryTest.testAddNewBook:21 El libro se agrego correctamente
[INFO]
[ERROR] Tests run: 3, Failures: 2, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.162 s
[INFO] Finished at: 2025-02-10T20:28:42-05:00
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:3.2.5:test (default-test) on project maven
-archetype-quickstart: There are test failures.
[ERROR]
[ERROR] Please refer to C:\Users\Juan\Desktop\CVDS\Lab-03-CVDS\Library\target\surefire-reports for the individual test r
esults.
[ERROR] Please refer to dump files (if any exist) [date].dump, [date]-jvmRun[N].dump and [date].dumpstream.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException

C:\Users\Juan\Desktop\CVDS\Lab-03-CVDS\Library>
```

ii. Implementación

```java
/**
 * Adds a new {@link edu.eci.cvds.tdd.library.book.Book} into the system, the book is store in a Map that contains
 * the {@link edu.eci.cvds.tdd.library.book.Book} and the amount of books available, if the book already exist the
 * amount should increase by 1 and if the book is new the amount should be 1, this method returns true if the
 * operation is successful false otherwise.
 *
 * @param book The book to store in the map.
 *
 * @return true if the book was stored false otherwise.
 */
public boolean addBook(Book book) throws IllegalArgumentException {
    if (book == null) {
        throw new IllegalArgumentException("El libro no puede ser nulo");
    }

    if (books.containsKey(book)){
        books.put(book, books.get(book) + 1);
    }else {
        books.put(book, 1);
    }
    return true;
}
```

iii.   Pruebas finales

```
C:\Windows\System32\cmd.e     X      +   ∨                                                          –    □    ×

[INFO] Compiling 2 source files with javac [debug target 1.8] to target\test-classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ maven-archetype-quickstart ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running edu.eci.cvds.tdd.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.060 s -- in edu.eci.cvds.tdd.AppTest
[INFO] Running edu.eci.cvds.tdd.library.LibraryTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.039 s -- in edu.eci.cvds.tdd.library.LibraryTes
t
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ maven-archetype-quickstart ---
[INFO] Building jar: C:\Users\Juan\Desktop\CVDS\Lab-03-CVDS\Library\target\maven-archetype-quickstart-1.0-SNAPSHOT.jar
[INFO] -------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] -------------------------------------------------------
[INFO] Total time:  5.129 s
[INFO] Finished at: 2025-02-10T20:57:34-05:00
[INFO] -------------------------------------------------------

C:\Users\Juan\Desktop\CVDS\Lab-03-CVDS\Library>
```

c. TEST LOANBOOK
   i. Pruebas
      1. Prestamo exitoso

```java
@Test
public void testLoanBookCorrectly() {
    Book book = new Book(tittle:"Tittle 1", author:"Author 1", isbn:"isbn1");
    library.addBook(book);
    User user = new User(name:"user1",id:"id1");
    library.addUser(user);
    Loan loan = library.loanABook(userId:"id1", isbn:"isbn1");
    assertNotNull(message:"El préstamo se realizo porque esta el libro", loan);
}
```

2. Prestamo con libro no existente

```java
@Test(expected = IllegalArgumentException.class)
public void testLoanBookFailIfBookNotExists() {
    User user = new User(name:"user1",id:"id1");
    library.addUser(user);

    Loan loan = library.loanABook(userId:"nonExistentUser", isbn:"isbn1");
    assertNull(message:"El préstamo es nulo porque no hay libros", loan);
}
```

3. Prestamo con usuario no existente

```java
@Test(expected = IllegalArgumentException.class)
public void testLoanBookFailIfUserNotExists() {
    User user = new User(name:"user1",id:"id1");
    library.addUser(user);

    Loan loan = library.loanABook(userId:"nonExistentUser", isbn:"isbn1");
    assertNull(message:"El préstamo es nulo porque no hay libros", loan);
}
```

4. Prestamo con prestamo ya realizado

```java
@Test
public void testLoanBookButWasLoan() {
    Book book = new Book(tittle:"Tittle 1", author:"Author 1", isbn:"isbn1");
    library.addBook(book);
    User user = new User(name:"user1",id:"id1");
    library.addUser(user);
    Loan loan = library.loanABook(userId:"id1", isbn:"isbn1");
    Loan loan2 = library.loanABook(userId:"id1", isbn:"isbn1");
    assertNull(message:"El préstamo no se realizo porque no hay libro", loan2);
}
```

5. Prestamo pero no hay cantidad de libros suficientes.

```java
@Test(expected = IllegalArgumentException.class)
public void testLoanBookNoAmount() {
    Book book = new Book(tittle:"Tittle 1", author:"Author 1", isbn:"isbn1");
    library.addBook(book);
    User user = new User(name:"user1",id:"id1");
    User user2 = new User(name:"user2",id:"id2");
    library.addUser(user);
    library.addUser(user2);
    Loan loan = library.loanABook(userId:"id1", isbn:"isbn1");
    Loan loan2 = library.loanABook(userId:"id2", isbn:"isbn1");
    assertNull(message:"El préstamo no se realizo porque no hay libro", loan2);
}
```

ii. Compilación fallida

iii. Implementación

```java
public Loan loanABook(String userId, String isbn) {

    Book book = searchBookByIsbn(isbn);

    if (book==null){
        throw new IllegalArgumentException("El libro con ISBN " + isbn + " no existe.");
    }
    if(books.containsKey(book) && verifyUser(userId) && verifyLoans(userId,book) ){

        if(books.get(book) > 0){
            User user = searchUserById(userId);
            Loan loan = new Loan(book,user);
            loans.add(loan);
            books.put(book, books.get(book) - 1);
            return loan;
        }else {
            throw new IllegalArgumentException(s:"No hay libros disponibles para prestar.");
        }
    }
    return null;
}
public User searchUserById(String userId){
    for(User u : users){
        if (u.getId().equals(userId)){
            return u;
        }
    }
    return null;
}
public Book searchBookByIsbn(String isbn) {
    for (Book book : books.keySet()) {
        if (book.getIsbn().equals(isbn)) {
            return book;
        }
    }
    return null;
}

public boolean verifyUser(String userId) {
    for (User u : users) {
        if (u.getId().equals(userId)) {
            return true;
        }
    }
    return false;
}

public boolean verifyLoans(String userId, Book book){
    for (Loan l: loans){
        if (l.getUser().getId().equals(userId) && l.getBook().equals(book) && l.getStatus() == LoanStatus.ACTIVE){
            return false;
        }
    }
    return true;
}
```

iv. Pruebas Finales

```
C:\Windows\System32\cmd.e    X    +   v                                                                    —   □   X
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ maven-archetype-quickstart ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ maven-archetype-quickstart ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running edu.eci.cvds.tdd.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.047 s -- in edu.eci.cvds.tdd.AppTest
[INFO] Running edu.eci.cvds.tdd.library.LibraryTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.034 s -- in edu.eci.cvds.tdd.library.Li
braryTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ maven-archetype-quickstart ---
[INFO] Building jar: C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\target\maven-archetype-quickstart-1
.0-SNAPSHOT.jar
[INFO] -------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] -------------------------------------------------------
[INFO] Total time:  2.566 s
[INFO] Finished at: 2025-02-11T19:39:49-05:00
[INFO] -------------------------------------------------------

C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library>
```

d. Test returnLoan
   i. Pruebas

1. Retorno de préstamo valido

```java
@Test
public void testReturnLoanValid(){
    Book book = new Book(tittle:"Tittle 1", author:"Author 1", isbn:"isbn1");
    library.addBook(book);
    User user = new User(name:"user1",id:"id1");
    library.addUser(user);
    Loan loan = library.loanABook(userId:"id1", isbn:"isbn1");
    assertEquals(library.returnLoan(loan).getStatus(),LoanStatus.RETURNED);
}
```

2. Préstamo no esta en el sistema

```java
@Test(expected = IllegalArgumentException.class)
public void testReturnLoanNoPresentInLoansOfSystem(){
    Book book = new Book(tittle:"Tittle 1", author:"Author 1", isbn:"isbn1");
    User user = new User(name:"user1",id:"id1");
    Loan loan = new Loan(book,user);
    assertEquals(library.returnLoan(loan).getStatus(),LoanStatus.RETURNED);
}
```

3. Préstamo null

```java
@Test(expected = IllegalArgumentException.class)
public void testReturnLoanNull(){
    assertEquals(library.returnLoan(loan:null).getStatus(),LoanStatus.RETURNED);
}
```

4. Préstamo que ya ha sido retornado

```java
@Test(expected = IllegalArgumentException.class)
public void testReturnLoanAlreadyReturned(){
    Book book = new Book(tittle:"Tittle 1", author:"Author 1", isbn:"isbn1");
    library.addBook(book);
    User user = new User(name:"user1",id:"id1");
    library.addUser(user);
    Loan loan = library.loanABook(userId:"id1", isbn:"isbn1");
    library.returnLoan(loan);
    assertEquals(library.returnLoan(loan).getStatus(),LoanStatus.RETURNED);
}
```

ii. Compilación fallida

iii. Implementación


```java
/**
 * This method return a loan, meaning that the amount of books should be increased by 1, the status of the Loan
 * in the loan list should be {@link edu.eci.cvds.tdd.library.loan.LoanStatus#RETURNED} and the loan return
 * date should be the current date, validate that the loan exist.
 *
 * @param loan loan to return.
 *
 * @return the loan with the RETURNED status.
 */
public Loan returnLoan(Loan loan) {
    if (loan == null || !loans.contains(loan) || loan.getStatus() != LoanStatus.ACTIVE) {
        throw new IllegalArgumentException(s:"El préstamo no es válido o ya fue devuelto.");
    }

    loan.setStatus(LoanStatus.RETURNED);
    loan.setReturnDate(LocalDateTime.now());
    Book book = loan.getBook();
    if (book != null && books.containsKey(book)) {
        books.put(book, books.get(book) + 1);
    }
    return loan;
}
```

iv. Pruebas Finales

```
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ maven-archetype-quickstart ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ maven-archetype-quickstart ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running edu.eci.cvds.tdd.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.047 s -- in edu.eci.cvds.tdd.AppTest
[INFO] Running edu.eci.cvds.tdd.library.LibraryTest
[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.062 s -- in edu.eci.cvds.tdd.library.L
ibraryTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 12, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ maven-archetype-quickstart ---
[INFO] Building jar: C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library\target\maven-archetype-quickstart-1
.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.901 s
[INFO] Finished at: 2025-02-11T19:51:12-05:00
[INFO] ------------------------------------------------------------------------

C:\Users\Esteban Aguilera\Documents\Lab-03-CVDS\Library>
```

8. Cobertura

   a. Agregamos la dependencia de jacoco, compilamos y miramos el archivo generado

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
edu.eci.cvds.tdd.library		92 %		75 %	11	31	5	52	0	9	0	1
edu.eci.cvds.tdd.library.loan		73 %		n/a	5	12	8	24	5	12	0	2
edu.eci.cvds.tdd.library.user		52 %		n/a	3	5	5	10	3	5	0	1
edu.eci.cvds.tdd		0 %		n/a	2	2	3	3	2	2	1	1
edu.eci.cvds.tdd.library.book		78 %		n/a	2	5	2	9	2	5	0	1
Total	63 of 407	84 %	11 of 44	75 %	23	55	23	98	12	33	1	6

   b. Al revisar los metodos no usamos, notamos que son gets y sets de las clases loan, user y book. Decidimos borrarlos ya que no hacemos uso de estos

   c. Compilamos nuevamente para ver el nuevo %

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
edu.eci.cvds.tdd.library		93 %		75 %	11	31	4	49	0	9	0	1
edu.eci.cvds.tdd.library.loan		87 %		n/a	2	9	3	19	2	9	0	2
edu.eci.cvds.tdd		0 %		n/a	2	2	3	3	2	2	1	1
edu.eci.cvds.tdd.library.book		100 %		n/a	0	3	0	7	0	3	0	1
edu.eci.cvds.tdd.library.user		100 %		n/a	0	2	0	5	0	2	0	1
Total	33 of 378	91 %	11 of 44	75 %	15	47	10	83	4	25	1	6

9. SONARQUBE

   a. Instalamos docker
   b. Arrancamos el servicio

c. Probamos el funcionamiento

d. Instalamos la extension de sonar en visual
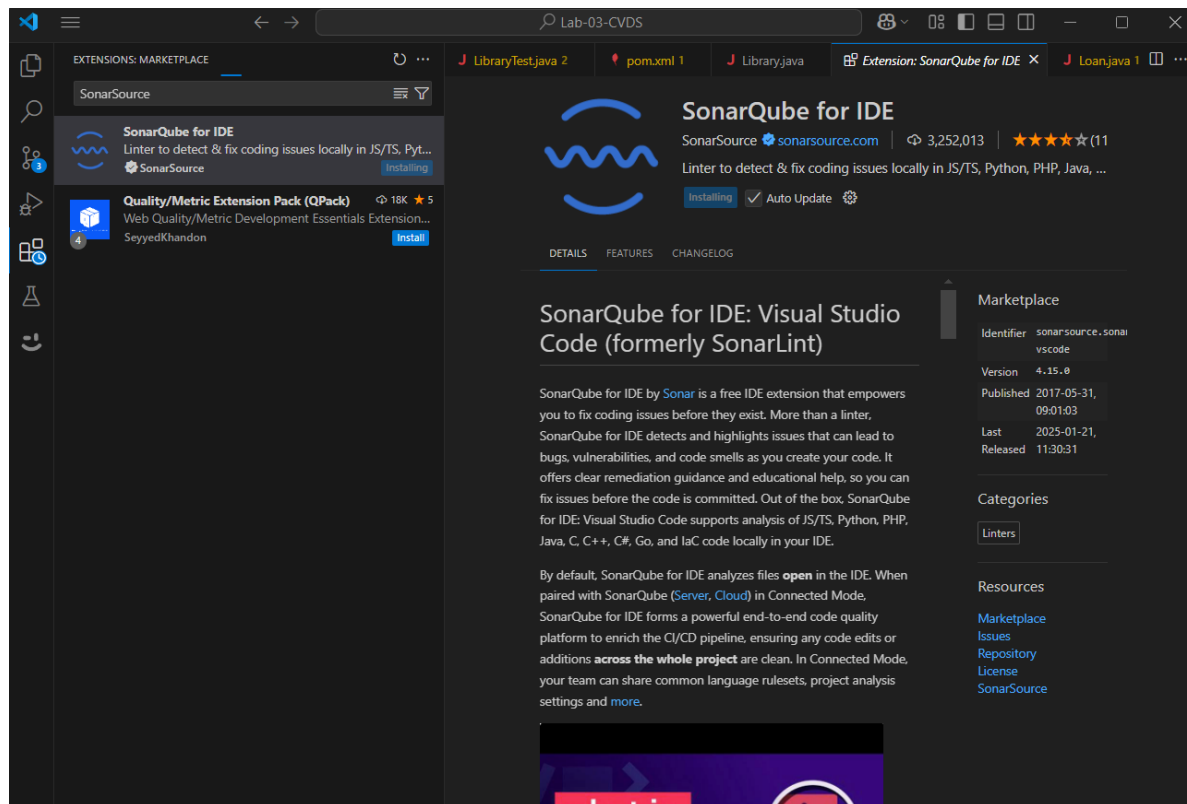
e. Creamos el token en la pagina

Security

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name	Type	Expires in	
Enter Token Name	Select Token Type	30 days	Generate

✓ New token "token" has been created. ...opy it now, you won't be able to see it again!

Copy to clipboard

sqa_d39511e934dbd5f96270d94b5efa381835c972fc

Name	Type	Project	Last use	Created	Expiration	
token	Global		Never	February 11, 2025	March 12, 2025	Revoke

Enter a new password

All fields marked with * are required

Old Password *

Password *

f.   Compilamos el proyecto y miramos la cobertura:

My Favorites | All

Filters

Quality Gate
✓ Passed   1
✗ Failed   0

Security
A ≥ 0 info issues   1
B ≥ 1 low issue   0
C ≥ 1 medium issue   0
D ≥ 1 high issue   0
E ≥ 1 blocker issue   0

Reliability
A ≥ 0 info issues   0
B ≥ 1 low issue   0
C ≥ 1 medium issue   1
D ≥ 1 high issue   0
E ≥ 1 blocker issue   0

Maintainability
Ratio of the estimated time needed to fix all outstanding maintainability issues to the size of the project
A ≤ 5% to 0%   1
B ≥ 5% to <10%   0
C ≥ 10% to <20%   0
D ≥ 20% to <50%   0
E ≥ 50%   0

Security Review
The percentage of reviewed (fixed or safe) security hotspots
A ≥ 80%   1

Create Project ⌄

🔍 Search for projects... | **Perspective** Overall Status ⌄ | **Sort by** Name ⌄ ⇅ 1 project(s) 🏠

⭐ library **PUBLIC**   ✓ Passed
Last analysis: 39 seconds ago • 249 Lines of Code • Java, XML

| A 0 | C 3 | A 20 | A — | ○ 85.0% | • 0.0% |
| Security | Reliability | Maintainability | Hotspots Reviewed | Coverage | Duplications |

1 of 1 shown