





UNIVERSIDAD DE CASTILLA-LA MANCHA ESCUELA SUPERIOR DE INFORMÁTICA

Álvaro Ávila Cabello Daniel López Paredes Luis Miguel Barreiro Cabello Víctor Lay Gómez

Asignatura: Ingeniería del Software II

Titulación: Grado en Ingeniería Informática

Fecha: 15 de octubre del 2023





ÍNDICE

1. Introducción	l
1.1 Enunciado de la práctica	1
2. Análisis de requisitos	1
3. Casos de usos y diagrama clase	2
3.1. Diagrama de casos de usos	2
3.2 Diagrama de clase	2
4. Construcción de la estructura	3
4.1 Maven y Spring-Boot:	3
MAVEN	3
SPRING-BOOT	3
4.1. Clases.	3
4.3 Maven y spring-boot	4
5. Control del proyecto	

1. Introducción

1.1 Enunciado de la práctica

Se pretende desarrollar un sistema web para la gestión de una biblioteca con los siguientes requisitos, que lo pueden usar tres usuarios diferentes con la siguiente funcionalidad

2. Análisis de requisitos

Se pretende desarrollar un sistema web para la gestión de una biblioteca con los siguientes requisitos, que lo pueden usar tres usuarios diferentes con la siguiente funcionalidad

• Administrador de biblioteca

- O Da de alta a nuevos títulos en el catálogo, para lo cual es requisito indispensable que al menos añada un ejemplar.
- o Añadir nuevos ejemplares, es decir, un nuevo libro (copia) de un título.
- o Borrar o actualizar un título existente.
- o Borrar un ejemplar (por ejemplo, por estar en mal estado, extravío, etc.)

• Bibliotecario/a:

- Prestar ejemplar. En este caso el sistema comprobará que el usuario no tiene el cupo de libros para sacar completo, ni tiene penalizaciones pendientes.
- Gestionar devolución de un ejemplar. En el caso de devolución con retardo, el sistema aplicará una penalización al usuario que se tendrá en cuenta para futuros préstamos.
- El trabajador podrá hacer la reserva de un ejemplar cuando un usuario desee un ejemplar que no está disponible. Al realizar un préstamo, todas las reservas para dicho ejemplar se borrarán.

• Usuario:

 A través del terminal en la biblioteca podrá de forma autónoma hacer la misma funcionalidad que el trabajador de la biblioteca.

REQUISITOS FUNCIONALES

- 1. Añadir título
- 2. Borrar título
- 3. Actualizar título
- 4. Añadir ejemplar (copia de un título)
- 5. Borrar ejemplar (y si es el último borrar el título)
- 6. Prestar un ejemplar
- 7. Comprobar cupo de libros (núm. máx. de libros prestados por usuario)
- 8. Comprobar penalizaciones
- 9. Recoger devolución de ejemplar
- 10. Aplicar penalización por retardo de ejemplar
- 11. Reservar ejemplar cuando no esté disponible

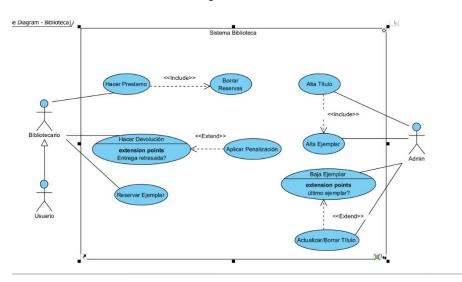




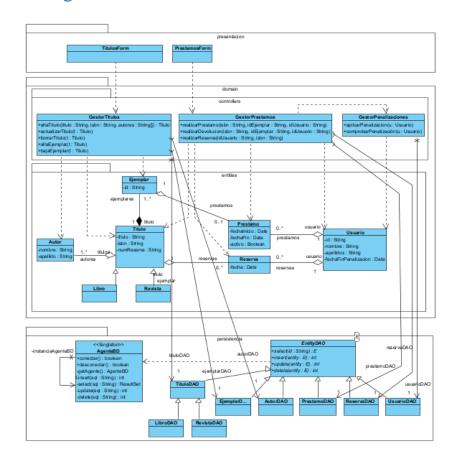
3. Casos de usos y diagrama clase

3.1. Diagrama de casos de usos

A continuación, se muestra el diagrama de caso de usos:



3.2 Diagrama de clase







ByteMasters_Library [ByteMasters_Library develop]

- > # src/main/java
- > @ src/main/resources
- > # src/test/java
- > JRE System Library [jre]
- > Maven Dependencies
- > 🗁 database
- > 🗁 src
- > 🗁 target
 - derby.log
 - HELP.md
 - mvnw
 - mvnw.cmd

 - README.md

nplearemos Maven, ya que nos permite ura del proyecto con el fin de

ación, construcciones, documentación, publicaciones.

a imagen, como se puede apreciar la e esta estructura es la estándar, además as dependencias mediante las descargas

Arquitectura de Maven: hay un archivo que es muy importante destacar en la construcción de la estructura del proyecto que es el POM (*Project Object Model*), contiene los metadatos sobre el proyecto, la localización de los directorios, desarrolladores y contribuidores, dependencias....

SPRING-BOOT

Utilizaremos Spring-boot para la creación de la aplicación, ya que nos da soporte al utilizar el lenguaje en java.

4.1. Clases.





4.3 Maven y spring-boot

5. Control del proyecto

Para el sistema de control de versiones, utilizaremos la herramienta **git**, esta nos permite almacenar los cambios que se produzcan y continuar seguir trabajando, también poder volver hacia atrás a versiones anteriores, para garantizar la estabilidad del software y compartir distintas versiones realizadas por diferentes miembros del equipo de desarrollo y una de las características más importantes es la seguridad frente a eliminaciones erróneas o fallos en los discos.

Git permite el desarrollo del software de forma no lineal, a través de las ramas en las cuales se pueden desarrollar las partes de las diferentes funcionalidades del software sin que se altere la estructura en otras.

Para ello se emplearán 4 tipo de ramas, la rama *main*, la rama develop, la rama de funcionalidades y la rama de *hotfix*.

Como se puede ver en el proyecto las diferentes ramas, en el apartado de git se nos muestra toda la configuración de las ramas.

- - 🦠 develop bbb9868 Merge pull request #51 añadida ventana inicio y login
 - \land feature-login 766a490 login hecho
- ✓ № Remote Tracking
 - origin/develop bbb9868 Merge pull request #51 añadida ventana inicio y login
 - 🏟 origin/feature-añadir-titulo ac79c03 añadir autores a un titulo separados por comas y verificar que no se generen duplicados en la base de datos
 - nigin/feature-detalles-titulo 715fa1d Merge branch 'develop' into feature-detalles-titulo
 - \land origin/feature-editar-numero-ejemplares 5d3e6b6 cambio en la jerarquía de archivos
 - nigin/feature-interfaz-inicio 1e91a44 actualizando gitignore
 - 🚸 origin/feature-login 766a490 login hecho
 - 🚸 origin/feature-persistencia c604ec6 añadido login
 - \land origin/feature-registrar-devolucion-ejemplar 48f1ed7 añadida feature para registrar la devolucion de un ejemplar
 - origin/feature-sonar 4cea830 Merge pull request #49 feature devolucion ejemplar
 - noigin/feature-usuario-prestar-ejemplar 1c08596 feature pedir prestamo desde la ventana de usuario
 - \clubsuit origin/hotfixes fe61bdb README.md añadido
 - origin/main fe61bdb README.md añadido
 - nigin/releases fe61bdb README.md añadido

A continuación, se muestra el historial en el cual se ve aprecian las ramas, y cuando se termina se hace merge con la rama de desarrollo





Id	Me	ssage	Author	Authored Date	Committer	Committed Da
09da128		empiece de interfaz, formulario de dar de alta titulo (solo nombre y isbn) pero no	Daniel	4 weeks ago	Daniel	4 weeks ago
baf1ef7		cambio aplication.properties y en entidades para solucionar problemas en la BBE	Daniel	4 weeks ago	Daniel	4 weeks ago
c1675b9		cambios persistencia entidade, estilos aplicados, base de datos cambio a create-	Daniel	4 weeks ago	Daniel	4 weeks ago
759ae5b	l	Merge pull request #29 from alvaro-avilac/develop	DaniLopez23	4 weeks ago	GitHub	4 weeks ago
e70c701	۰	borradas entidades innecesarias	Daniel	4 weeks ago	Daniel	4 weeks ago
569fed1	٩	conflictos solucionados	Daniel	4 weeks ago	Daniel	4 weeks ago
f2b35e9	þ	Merge branch 'develop' of https://github.com/alvaro-avilac/ByteMasters_Library	Daniel	4 weeks ago	Daniel	4 weeks ago
eca4d4a	þ	Update .gitignore	Álvaro Ávila Cabello	4 weeks ago	GitHub	4 weeks ago
5218c20	۰	eliminados entidades y DAOS innecesarios hasta ahora	Daniel	4 weeks ago	Daniel	4 weeks ago
20548d6		DAO comenzados (falta cardinalidades), gestorTitulos reducido	Daniel	4 weeks ago	Daniel	4 weeks ago
08a387d		comienzo de formulario alta titulo	Alvaro Avila	4 weeks ago	Alvaro Avila	4 weeks ago
2f30cdc	Ш	Merge pull request #26 from alvaro-avilac/develop	Álvaro Ávila Cabello	4 weeks ago	GitHub	4 weeks ago
5c3c259	Q	Merge branch 'feature-añadir-titulo' into develop	Álvaro Ávila Cabello	4 weeks ago	GitHub	4 weeks ago
8f287e2	þ	Merge branch 'develop' of https://github.com/alvaro-avilac/ByteMasters_Library	Daniel	4 weeks ago	Daniel	4 weeks ago
5b4df09	þ	cambios en los packages de los controllers y vaciarlos	Alvaro Avila	4 weeks ago	Alvaro Avila	4 weeks ago
b862712	þ	cambios en los packages de las entidades	Alvaro Avila	4 weeks ago	Alvaro Avila	4 weeks ago
92d3c35	0	DAO corregido	Daniel	4 weeks ago	Daniel	4 weeks ago
7f078ae	Ш	capa servicio titulo, controlador titulos empezado	Daniel	4 weeks ago	Daniel	4 weeks ago
3d81254	(Template, archivos html y aplication.properties	Daniel	4 weeks ago	Daniel	4 weeks ago
e7ea834		Esqueleto, archivos base visual paradigm	Daniel	4 weeks ago	Daniel	4 weeks ago
fe61bdb		origin/HEAD origin/hotfixes origin/main origin/releases README.md añadido	Alvaro Avila	5 weeks ago	Alvaro Avila	5 weeks ago
593c079		commit project	Alvaro Avila	5 weeks ago	Alvaro Avila	5 weeks ago
2a095fb		Initial commit	Daniel	5 weeks ago	Daniel	5 weeks ago

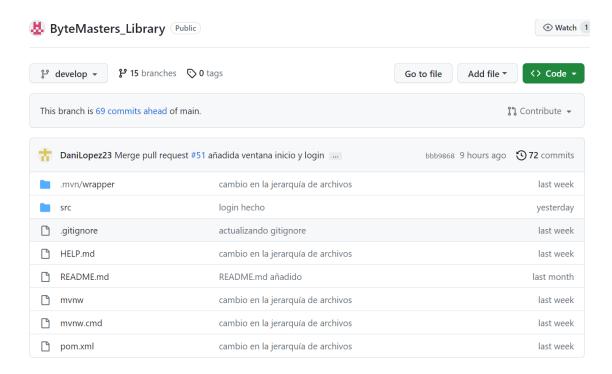
También se optó por tomar como servicio en la nube el alojar el repositorio con el proyecto, para ello utilizamos github, que es un servicio en la nube para repositorios remotos que utilizan git como sistema de control de versiones.

Github permite la colaboración en proyectos y poder compartir el código de una manera sencilla a través de github clonamos en los terminales la ubicación del repositorio, en el cual se realizará el proyecto, pero además github permite realizar muchas más funciones a la hora de gestión del proyecto, en las que a continuación detallaremos.

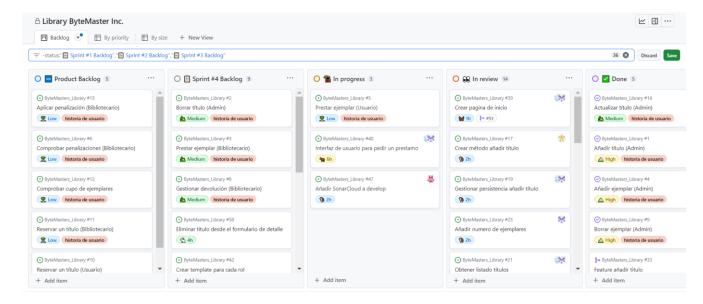
En github también se puede ver la estructura del proyecto, como se aprecia en la imagen inferior que corresponde con la rama develop







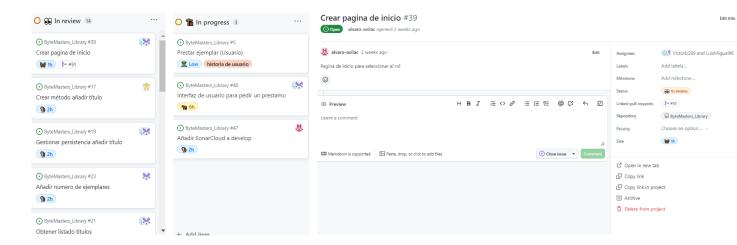
En github también permite poder establecer el backlog en su ventana Project en cada reunión de los sprint se definían los *porduct backlogs*, los cuales se iban añadiendo en el sprint de acuerdo con los principios de prioridad, como se puede apreciar en la imagen



En la imagen superior se muestra de manera general los diferentes estados en los que se encuentra los ítems del producto backlog. Una vez asignado un *item* al backlog del sprint correspondiente, se identificaban los miembros del equipo en cargados, el tiempo estimado y su prioridad. A continuación, se muestra en las imágenes los estados de *in progress*, *in rewiev*, así como una historia de usuario detallada







Cada historia de usuario es asignada en una nueva *issue* donde se va a indicar en que rama se va a desarrollar esa nueva funcionalidad ejemplo devolución ejemplar, como se puede apreciar en el apartado *developement*, esta asignada a una rama de funcionalidad específica para ella.

