



PLANIFICACIÓN DEL DESARROLLO DE SOFTWARE (SISTEMA DE BIBLIOTECA DIGITAL)

Materia: Taller de Sistemas I

Docente: Ing. Javier Kanqui

Autor	Versión	Fecha
Tomás Álvarez Daza	1.0	21-sep-2022

TOMÁS ÁLVAREZ DAZA

10 de diciembre de 2022

Índice general

I	Requerimientos	5
1.	LISTA DE REQUERIMIENTOS	6
1.1.	REQUERIMIENTOS FUNCIONALES	6
1.2.	REQUERIMIENTOS NO FUNCIONALES	6
1.3.	ESCALABILIDAD	6
II	Análisis del sistema	7
2.	Análisis de sistema	8
III	Diseño	11
	Diseño de base de datos	15
3.1.	Diagrama entidad relación	15
	DISEÑO DE FORMULARIO Y REPORTES	16
4.1.	Diseño de Salida efectiva	16
4.2.	Diseño de Entrada efectiva	16
IV	Implementación	18
	Tecnologías	19
5.	MAVEN	20
5.1.	MVN	20
5.1.1.	PROJECT OBJECT MODEL (POM)	20
5.2.	Dependencias utilizadas en el proyecto	21
5.2.1.	Lombok	21
6.	JAVA	22
6.1.	JDK	22
6.2.	Versión de java que se utiliza en el proyecto	22
7.	MySql/Mariadb	23
9.	SPRING	24
9.1.	Spring Boot	24
9.2.	Fácil manejo de dependencias	24
9.3.	Autoconfiguración	24
9.4.	Soprote integrado de contenedor de servlet	25
9.4.1.	Spring Initializr	25
9.5.	Herramientas de desarrollo	25

10.IntelliJ IDEA	26
11.1. Asistente al codificar	27
11.2. Refactorización	27
11.3. Terminal	27
11.4. Herramientas de construcción	27
11.5. Control de Versiones	27
11.6. Requerimientos del sistema e instalación	27
12.DBEAVER	29
13.SWAGGER	30
13.1. Swagger UI	30
14.Diccionario de datos	31
15.Descripción de API REST	32

PROBLEMA PRINCIPAL

INTRODUCCIÓN

Una biblioteca digital se concibe como una colección organizada de información. La organización es lo que caracteriza a la biblioteca de la world wide web donde hay una anarquía. Además un sitio web tiene que ser editado y actualizado escribiendo y modificando; sin embargo en una biblioteca digital se puede añadir fácilmente nuevo material. Además la librería digital tiene ventajas sobre las convencionales porque son fáciles de acceder y ofrecen búsquedas más poderosas.

Las bibliotecas son el repositorio de conocimientos de la humanidad aunque en la antigüedad solo era útil a un pequeño grupo de población. Pero, ahora con la revolución de la información hay una demanda para guardar, organizar y acceso a la información.



Goethe dijo una vez que visitar una biblioteca era como entrar en el presencia de una gran rico que silenciosamente estaba pagando dividendos incalculables

ANTECEDENTES INSTITUCIONALES

Necesidades del software

Actualmente un estudiante posee los libros digitales o digitalizados y otros como videos, artículos en un disco duro de 2 TB organizado en carpetas sin acceso fácil a ellos. En la actualidad sólo es posible hacer un link con un programa llamado Kiwix Desktop con el cual es posible acceder con un click al libro en formato pdf o cualquier otro video o artículo o audio (aquí llamados documentos). Este método es inefectivo ya que requiere mucho tiempo para hacer los links. Los libros están en diferentes formatos como ser: pdf, pub, djvu, pero mayormente en pdf, algunos en html. Actualmente solo se tiene en discos duros y en memoria de dispositivos móviles como tablets, celular o otro dispositivo, muchos ocupan mucha memoria porque hay redundancia de datos ya que una copia está en un disco y otra copia del mismo en otro dispositivo; sin embargo se sugiere tener una copia por si la copia descargada se corrompe.

Los videos, fueron descargados para que tengan buena calidad y sean de latencia mínima al reproducirlos, aunque no se tiene gran cantidad, pero en el futuro esto se incrementará. Los videos se tienen en discos externos; esto para no sobrecargar el almacenamiento de la computadora personal. Las imágenes no se tiene mucho, sin embargo como el alumno suele ser más visual en el método de aprendizaje tambien se prevee tener una base de datos de imágenes importante y relevantes a cada tema.

En otros se tiene herramientas como calculadoras, IDEs, CASE, diagramadores y páginas web relacionados con simulación y otros necesarios para aprendizaje.

Problemas por la no existencia del software

El alumno o estudiante tiene mucha información generada durante su paso como estudiante pero éstas estan desorganizadas y el acceso a veces se vuelve de forma que no se puede encontrar lo que se busca.

Si el alumno tiene ordenado en su computadora cuando se encuentra lejos de donde estudia, no puede acceder y debe llevar copias pero a veces copia un contenido pero tarda en copiar o simplemente copió otro contenido y un sin fin de problemas que puede tener el no accesos oportuno a la información.

LIMITES Y ALCANCES DEL SISTEMA

En primera instancia sólo se desarrollará para documentos tales como artículos y libros, sin embargo para versiones posteriores será extensible a cualquier objeto digital que el alumno requiera para su aprendizaje.

JUSTIFICACIÓN

Un usuario cualquiera requiere como soporte una colección bien articulada de información organizada y estructurada aquí llamados documentos (texto, audio, video es decir un conjunto de objetos digitales, simuladores, etc) para ayudarse un individuo a lograr un objetivo cualquiera y esto no requiere un gran espacio físico. Se quiere realizar esta biblioteca ya que la red de redes carece de características esenciales de selección y organización a pesar que pueden existir sitios web con contenido bien organizado la libreria digital puede expandirse facilmente añadiendose nuevo material. Además no a todo el territorio boliviano alcanza la red de redes y ademas si fuera esto posible la velocidad del internet no siempre es alta y que un video pueda ser reproducido de inmediato y sin latencia de carga y con el dificultad de que buscar algo en internet a veces suele ser como buscar en un pajar.

Hoy en día con la revolución de la información y el poder de la tecnología hay una demanda sin precedencias para el almacenamiento, organización y acceso a la información y si la información es la moneda de la economia del conocimiento entonces las librerias digitales serán los bancos donde se invierte [?].

Una biblioteca convencional o en carpetas tiene una sola forma de organizar, mientras que en una biblioteca digital se puede organizar de diferentes maneras.

OBJETIVOS DEL SISTEMA

El objetivo es realizar el diseño de una herramienta de estudio efectiva para población de diferentes niveles para ayudar en el proceso de aprendizaje.

OBJETIVOS SECUNDARIOS

- Hacer una biblioteca digital más flexible y adaptable posible.
- Seleccionar organizar y mantener objetos digitales para el estudiante

Parte I

Requerimientos

LISTA DE REQUERIMIENTOS

1.1. REQUERIMIENTOS FUNCIONALES

1. **Es accesible via navegador web**
2. **permite busqueda por texto a algún campo:** Por ejemplo por palabra clave clave.
3. **Busqueda flexible:** El usuario puede buscar por título, autor, fechas, ectructuras de clasificación, etc.
4. **Hace uso de la metadata disponible:** Para incorporar nuevo material a la biblioteca.
5. **Maneja otro tipo de objetos digitales:** Por ejemplo video, imágenes, audio, IDEs, páginas web es decir links.
6. Todo lo que ves lo puedes obtener.
7. El sistema puede funcionar en linea así también de forma local y que el estudiante pueda añadir facilmente nuevo material a su biblioteca digital.
8. El estudiante pueda conectarse de forma remota con su biblioteca digital de cualquier lugar por internet y así tambien por la red local.
9. Debe tener una versión de escritorio para S.O. comunes como Linux/GNU, Window y Mac.
10. Que permita clonar todo el sistema y su contenido de forma fácil a un dispositivo de almacenamiento masivo si se desea utilizarlo de forma local (sin internet)
11. Que puede manejar de forma automática la duplicidad de documentos

1.2. REQUERIMIENTOS NO FUNCIONALES

1. Debe sorporta multiples descargas al menos hata 20 simultaneamente.
2. La seguridad no debe ser tomado en cuenta, puede copiar o clonar el sitema cualquiera que desee si se desea trabajar de forma local.
3. Si se accede a la biblioteca por el navegador, el usuario debe estar suscrito.
4. Debe ser adaptable a cualquier contenido digital

1.3. ESCALABILIDAD

El software debe ser escalable para contener cualquier objeto digital y adecuarse al volumen de almacenamiento.

Parte II

Analisis del sistema

Análisis de sistema

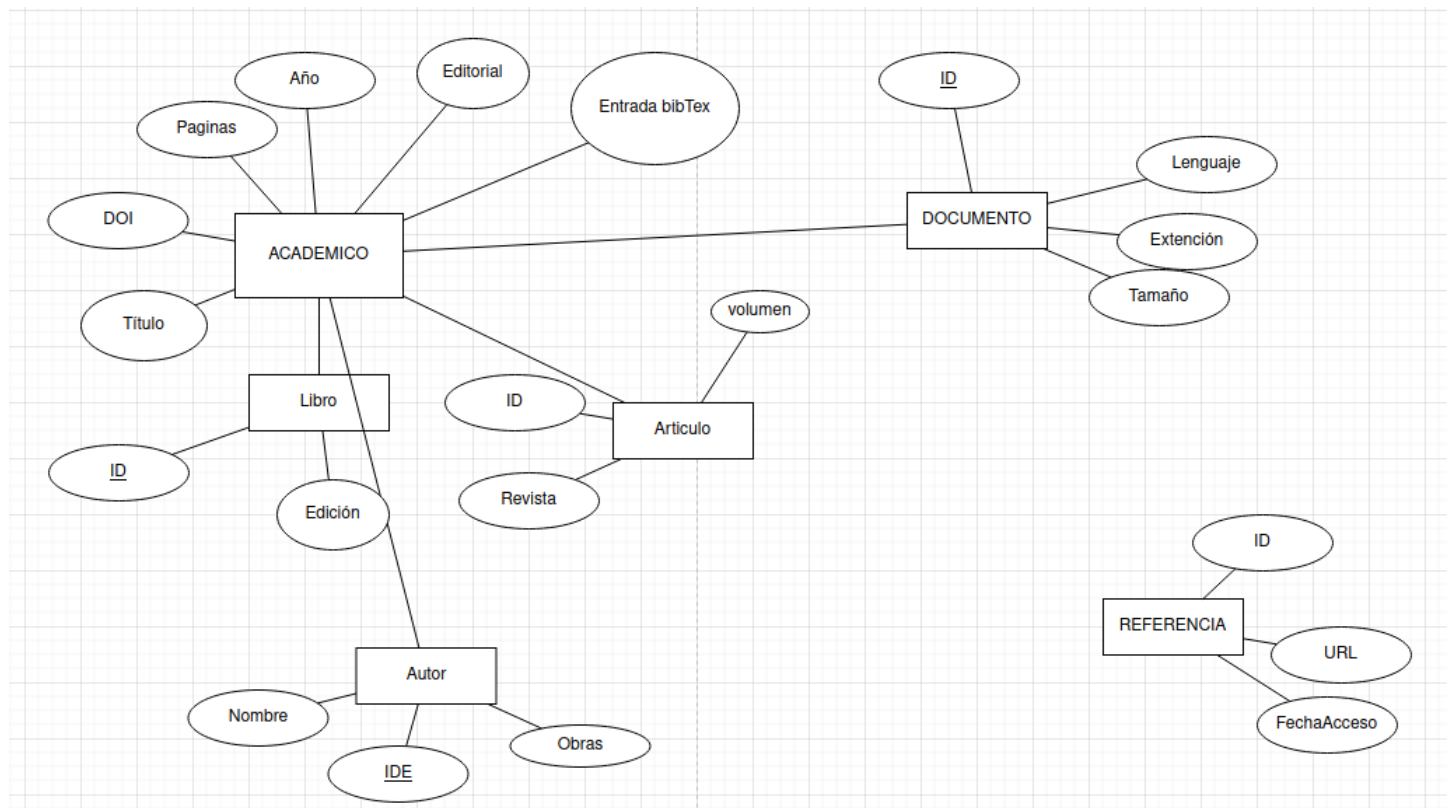


DIAGRAMA DE PROCESOS

En este diagrama tratamos de organizar para mejorar como se pueden hacer para beneficiar a a las partes interesadas. Se muestra lo que requiere un usuario, en este caso es recopilar documentación que piensa que requiere, recomendación de sus docentes para luego obtener un resultado según a los documentos u objetos digitales que pueda alimentar a la biblioteca.

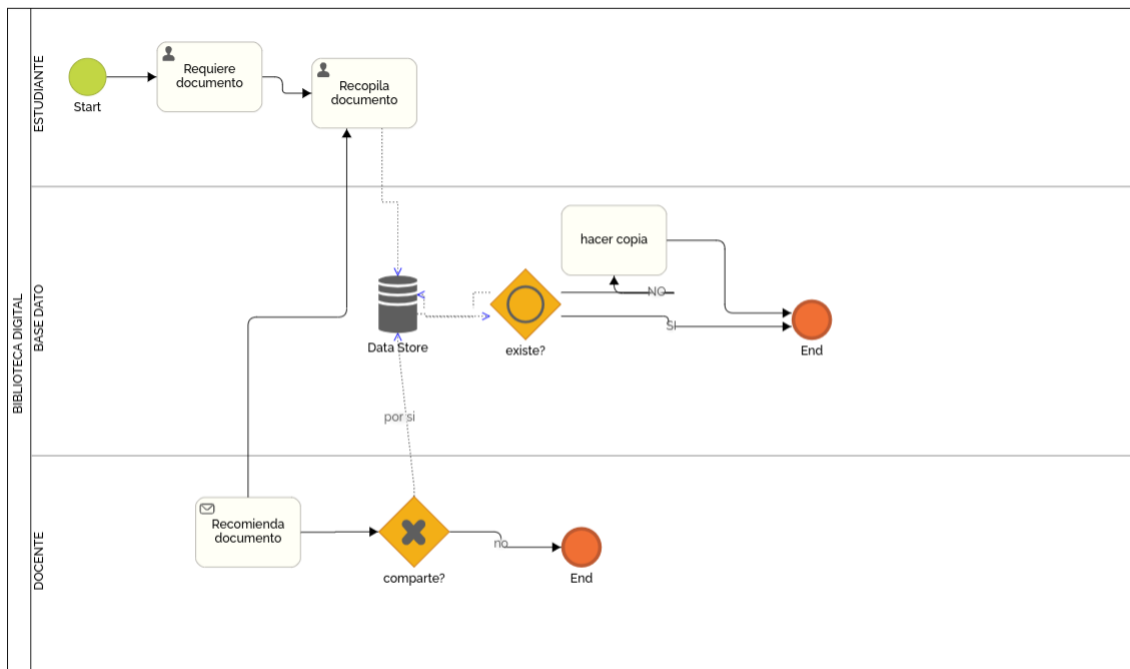


Figura 2.1: Modelado de proceso de la biblioteca digital

DIAGRAMA DE CASOS DE USO

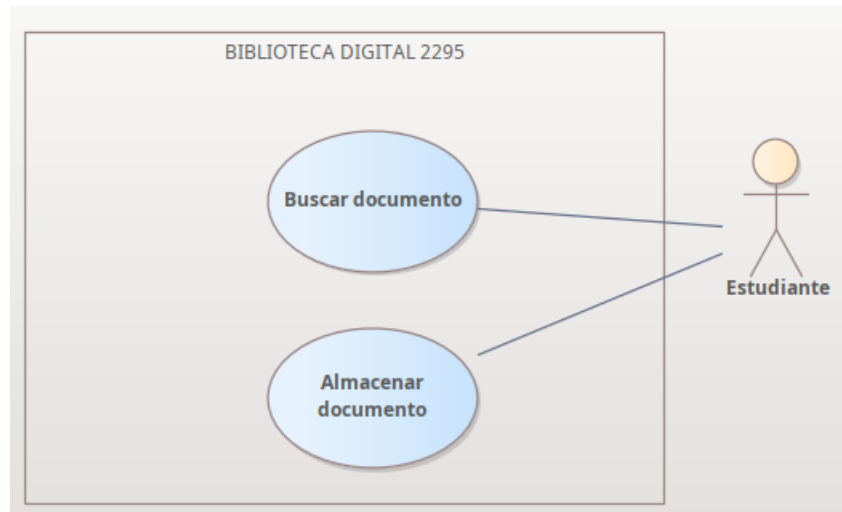


Figura 2.2: Diagrama de casos de uso

Id del caso de uso	CU-1
Caso de uso	Almacenar objeto digital
Actores	Estudiante administrador
Descripción	Almacena cualquier documento o objeto digital como libros, artículos, videos, audio para poder almacenar un objeto digital (documento o multimedia)
Precondición	Usuario ingresa al sistema de forma remota, local o por navegador
Post condición	Almacenamiento actualizado
Id del caso de uso	CU-2
Caso de uso	Consultar por un objeto digital
Actores	Otro estudiante
Descripción	Consulta por cualquier documento o objeto digital como libros, artículos, videos, audio
Precondición	El estudiante debe ingresar al sistema
Post condición	Necesidad de información satisfecha

Parte III

Diseño

DIAGRAMA DE CLASES

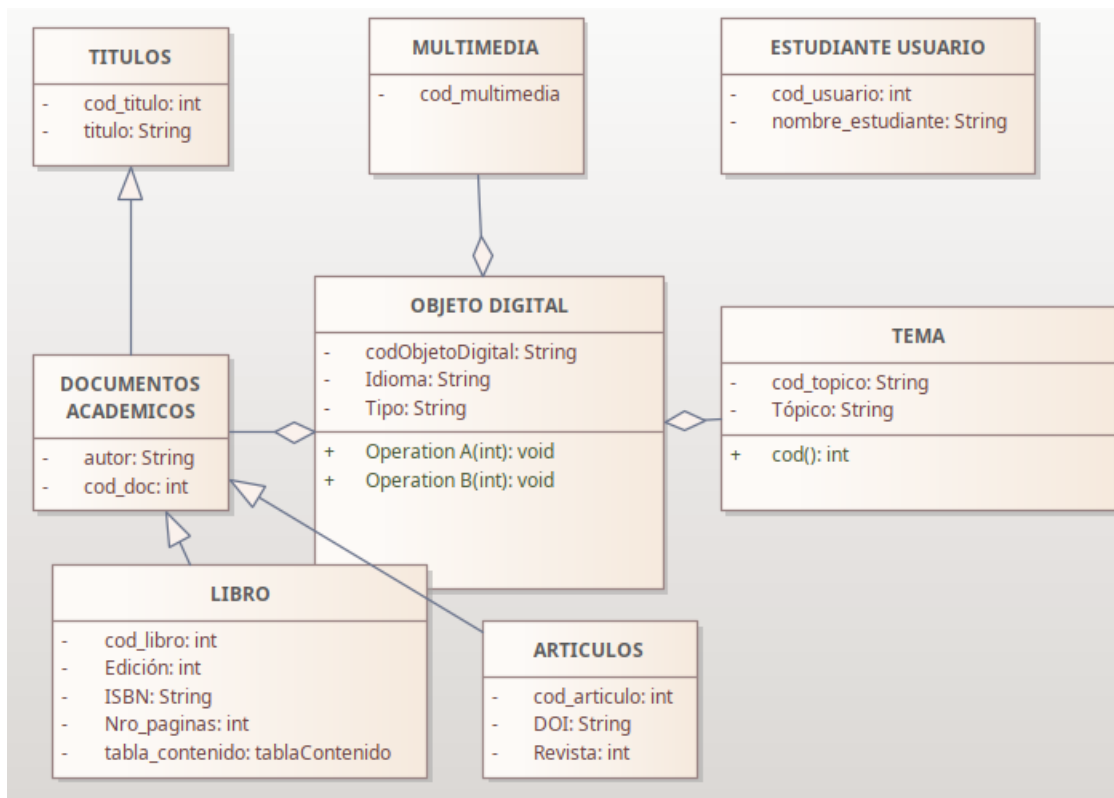


Figura 2.3: Diagrama de clases

DIAGRAMA DE SECUENCIA

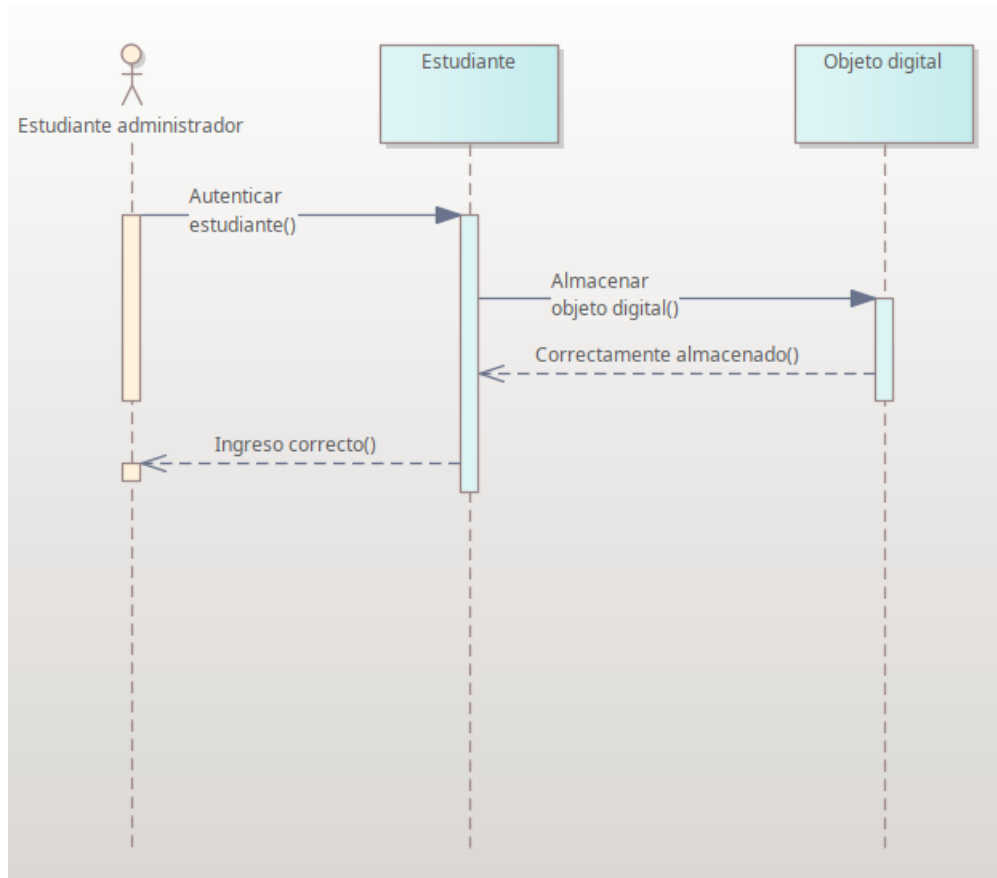


Figura 2.4: Diagrama de secuencia

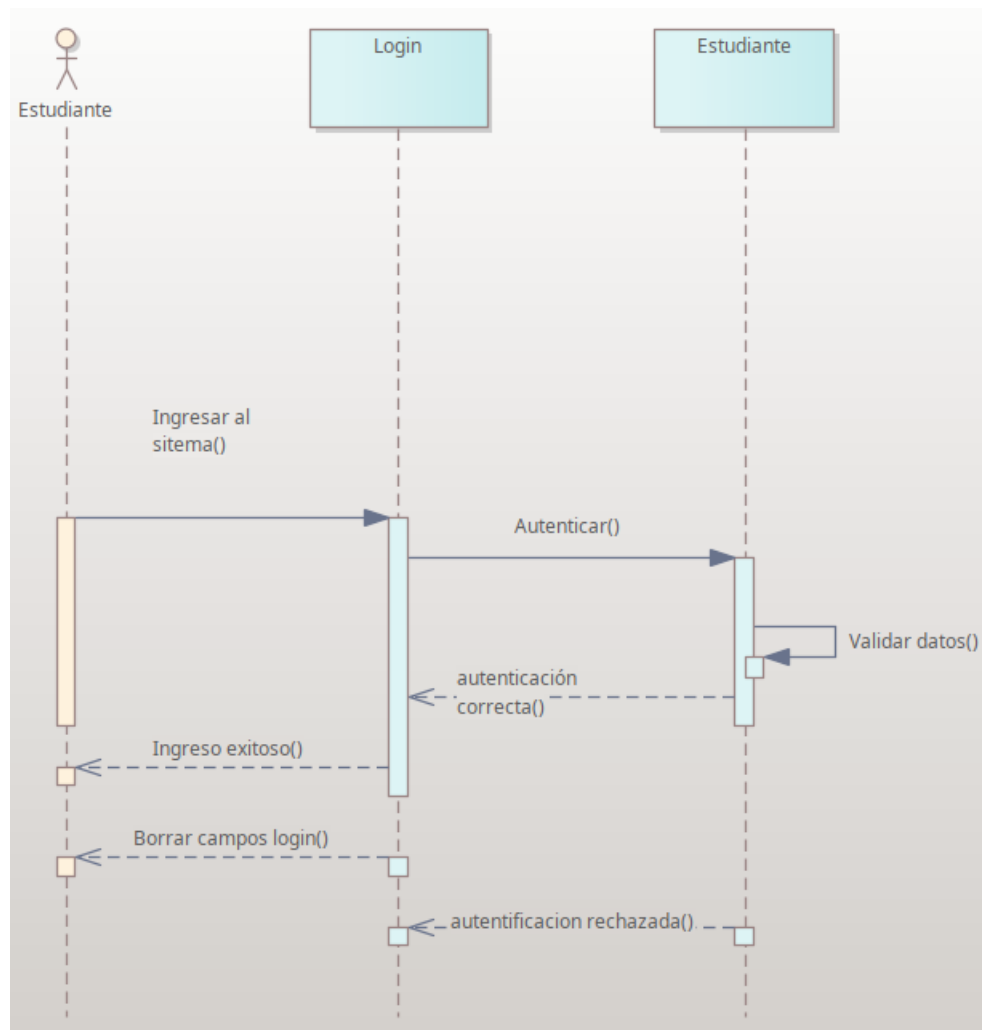


Figura 2.5: Diagrama de secuencia para que el estudiante se autentique en el sistema

Diseño de base de datos

3.1. Diagrama entidad relación

Se abstrae del problema real. Nuestras entidades serán el autor, Para el diseño entidad relación de base datos se utiliza la herramienta *Draw io*.

1. **DOI**, es un identificador único que permite identificar un artículo durante su vida útil aunque la revista en la que fue publicado desaparezca o cambia de nombre. Está compuesto por una parte que identifica a la entidad que registra el DOI y otra parte que identifica al objeto, en este caso, el artículo científico.

Diseño de formularios y reportes

Los formularios serán utilizados tanto para entradas o salida, así también los reportes para transmitir información sobre una colección de datos. El diseño de los formularios y reportes serán la clave para que este sistema sea exitoso. La información es recolectado y formateado de diferentes formas.

4.1. Diseño de Salida efectiva

La salida es rápida ya que no todos los datos están almacenados en la máquina local y los datos no requieren un procesamiento complejo si se trabaja de forma local. La salida sera preferentemente en pantalla pero se podrá imprimir si se requiere algún reporte de los libros almacenados.

La salida facilita al usuario a encontrar lo que realmente busca de manera útil y fácil dependiendo de sus datos o objetos digitales almacenados. Los documentos de pueden abrir con diferentes aplicaciones, reproductores o visualizadores si es un multimedia. Cambien se puede localizar la dirección donde el objeto está almacenado.

RESULTADOS DE BUSQUEDA				
Autor	Tipo documento	Año	Almacendo el:	Acciones
Siracus	Libro (pdf)	2020	11/sep/2022	Abrir en, localizar
Linores	Articulo (pdf)	2011	01/dic/2004	Abrir en, Localizar
Tomas Tilano	Audio (mp3)	2000	01/dic/2004	Reproducir, Localizar
Leonel Sambo	Imagen (jpg)	2002	01/dic/2004	Ver en, localizar

Volver a principal


Imprimir

Figura 4.1: Resultado "de la busqueda

4.2. Diseño de Entrada efectiva

El ingreso es mayormente de forma automática usando los metadatos de los objetos.

ELIJA RECURSO Y TÓPICO

Buscar 

2015 

< October 2014 >

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11


☐ Todos
☐ Video
☐ Audio
☐ Imagen
☐ Libro
☐ Artículos

☐ Todo
☐ Materia personalizada 1
☐ Materia personalizada 2
☐ Materia personalizada 3
☐ Materia personalizada 4
☐ Materia personalizada 5
☐ Materia personalizada 6

Figura 4.2: Pantalla principal de búsqueda

ALMACENAMIENTO DE DOCUMENTOS

Libro | **Artículo** | Imagen | Audio | Video | Otro




Datos del artículo

Título

Autor

Revista

Año de publicación 

DOI

Usar bibTex

CANCELAR **ALMACENAR**

Figura 4.3: Almacenamiento de artículos

Parte IV

Implementación

TECNOLOGIAS

Para la implementación se utiliza las siguientes tecnología:

- **Gestor de dependencias**, MVN
- **Lenguaje de programación**, JAVA 17
- **Framework**, Spring Boot
- **IDE**, GetBrain IDEA

Además de otros se expone continuación éstas tecnologías:

MAVEN

5.1. MVN

Apache Maven es una herramienta de comprensión y gestión de proyectos de software. Basado en el concepto de un modelo de objetos de proyecto (POM), Maven puede administrar la construcción, los informes y la documentación de un proyecto desde una pieza central de información. Se descarga de la siguiente dirección <https://maven.apache.org/download.cgi>, se instala y se verifica como sigue

```
tomas@debian:~$ mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.4, vendor: Debian, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en_GB, platform encoding: UTF-8
OS name: "linux", version: "5.10.0-19-amd64", arch: "amd64", family: "unix"
```

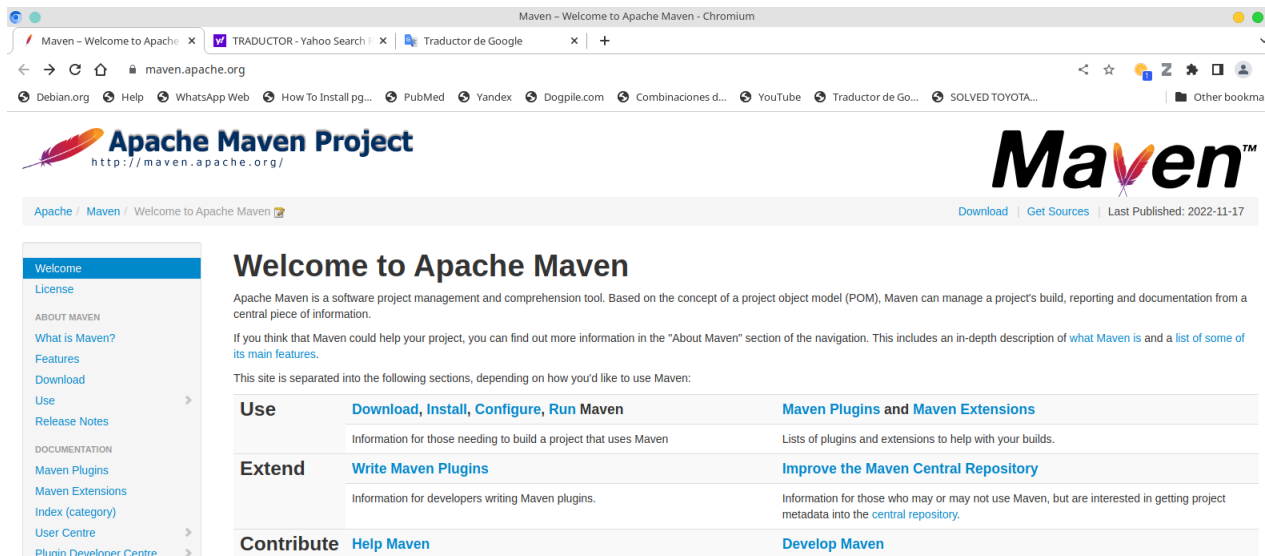


Figura 5.1: Página principal de maven

Se debe descara el binario comprimido en .tar (tape archiv); en window se puede descargar lo que puedes descomprimir (generalmente .zip). En windows se descomprime y se configura la carpeta binaria en los variable de entorno.

5.1.1. PROJECT OBJECT MODEL (POM)

Es un xml que contiene información a cerca del proyecto y configuración usado por Maven para compilar el proyecto. Project Object Model o POM es la parte básica de la funcionalidad de Maven. Este es un archivo XML que tiene información sobre las dependencias, configuraciones y otra información importante sobre el proyecto. Maven revisa esta información y luego realiza la tarea designada.

Spring Initializr es una API ¹ que permite la generación de proyectos con sus dependencias permitiendo simplificar esta etapa inicial de arranque de nuevos proyectos.

5.2. Dependencias utilizadas en el proyecto

Se instalaron varias dependencias en el proyecto:

5.2.1. Lombok

```
<dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

¹interfaz de programación de aplicaciones API es una forma en que dos o más programas de computadora se comunican entre sí. Es un tipo de interfaz de software que ofrece un servicio a otras piezas de software

JAVA

Fue desarrollado por el equipo liderado por James Gosling en Sun Microsystems. Sun microsystems fue comprado por Oracle en 2010. Originalmente llamado Oak, Java fue diseñado en 1991 para uso embebido de chips en electrodomésticos de consumo. En 1995, renombrado Java, fue re diseñado para desarrollo de aplicaciones Web.

Java se volvió enormemente popular. Fue descrito por sus diseñadores como *simple, orientado a objetos, distribuido, interpretado, robusto, seguro, neutral architecture, portable, de alto rendimiento, multiproceso y dinámico*.

Ahora Java es muy popular para desarrollo de aplicaciones en Servidores Web. Éstas aplicaciones procesan datos, realizan cálculos, y generan páginas web dinámicas. Muchos sitios web comerciales son desarrollados usando java en el backend.

Java bien es 3 ediciones: *Standard Edition*, *java Enterprise Edition (Java EE)*, y *Java Micro Edition (Java ME)*. Además éstos están en diferentes ediciones.

Java EE Es para desarrollar aplicaciones en el lado del servidor, tales como Java servlets, páginas JavaServer (JSP), y JavaServer Faces (JSF).

6.1. JDK

Consiste de un conjunto de programas separados. El JDK está en su versión 19 en el momento actual pero LTS es la versión 17 [[his](#),].

6.2. Versión de java que se utiliza en el proyecto

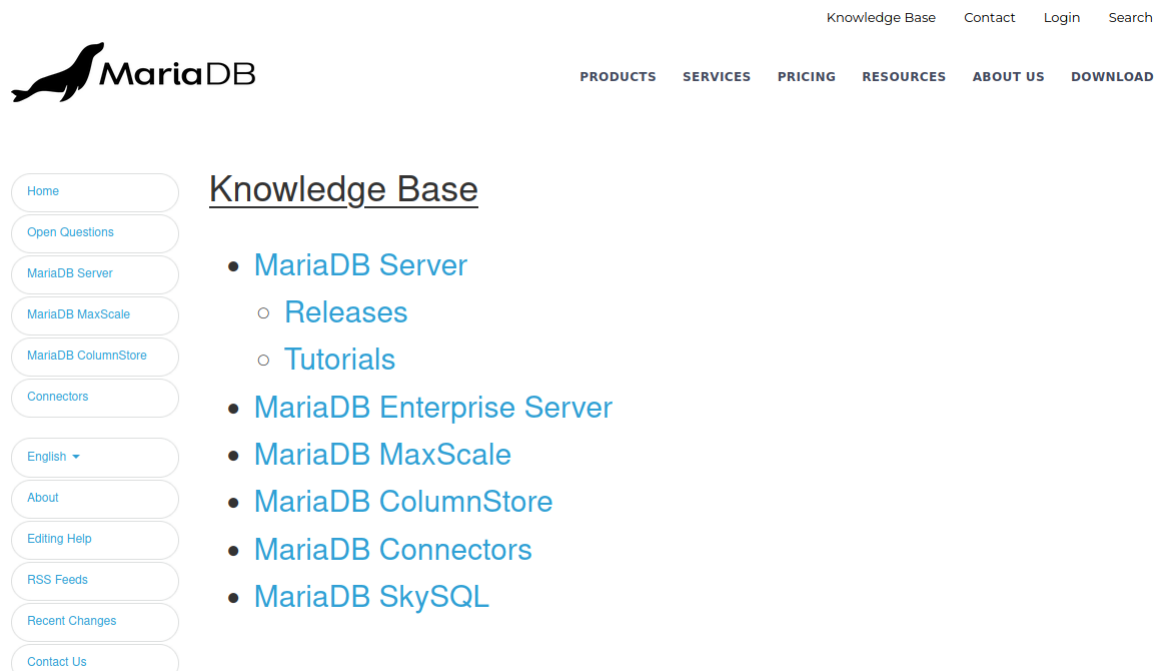
```
tomas@debian:~$ java --version
openjdk 17.0.4 2022-07-19
OpenJDK Runtime Environment (build 17.0.4+8-Debian-1deb11u1)
OpenJDK 64-Bit Server VM (build 17.0.4+8-Debian-1deb11u1, mixed mode, sharing)
```

MySql/Mariadb


MariaDB es una bifurcación de MySQL impulsada por la comunidad que Monty inició en 2009 Widenius, el autor original de MySQL, después de que Oracle adquiriera el antiguo proyecto. La primera versión de MariaDB se basó en MySQL 5.1 y las mejoras a El código base de MySQL se fusiona regularmente en el proyecto MariaDB. Otras características también se fusionan desde Percona Server¹, otra bifurcación que es muy similar a la producto principal.

Se utiliza el la siguiente versión:

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```



Knowledge Base Contact Login Search

 **MariaDB**

PRODUCTS SERVICES PRICING RESOURCES ABOUT US DOWNLOAD

Home
Open Questions
MariaDB Server
MariaDB MaxScale
MariaDB ColumnStore
Connectors
English ▾
About
Editing Help
RSS Feeds
Recent Changes
Contact Us

Knowledge Base

- [MariaDB Server](#)
 - [Releases](#)
 - [Tutorials](#)
- [MariaDB Enterprise Server](#)
- [MariaDB MaxScale](#)
- [MariaDB ColumnStore](#)
- [MariaDB Connectors](#)
- [MariaDB SkySQL](#)

¹Percona Server for MySQL® es un reemplazo gratuito, totalmente compatible, mejorado y de código abierto para cualquier base de datos MySQL. Proporciona un rendimiento superior, escalabilidad e instrumentación.

SPRING

El entorno de trabajo de Spring es popular y ampliamente usado entorno de trabajo de Java para construir aplicaciones web y de empresa. Su eje es la inyección de dependencias que provee flexibilidad para configurar de múltiple maneras, tales como XML, Comentarios, y configuraciones java. El entorno de trabajo Spring creció respondiendo a modernas necesidades de empresas como la seguridad, soporte para almacenamientos NoSql, manejo de GrandesDatos, proceso de lotes, integración con otros sistemas, etc.

El entorno de trabajo Spring es flexible, tiene capacidad y fácil de uso transacciones de base de datos. simplifica la integración con otros entornos de trabajo, tiene una tecnología de punta para construir aplicaciones web en Modelo Vista Controlador.

9.1. Spring Boot

Spring Boot es un entorno de trabajo que ayuda a desarrollar aplicaciones basados en Spring de forma fácil y rápida. Sus característica son:

- Iniciador de Spring Boot
- Auto configuración
- Manejo de configuración elegante
- Actuador Spring Boot
- Soporte de contenidos servlet y fácil de usar integrado

Spring Boot configura los componentes de la aplicación de forma automática, pero permitiendo invalidar los por defectos si requiere. Los pasos son:

1. Crear proyectos de Spring Boot basado en Maven y configurar las dependencias en el archivo *pom.xml*
2. Configurar los recursos de datos/propiedades JPA en *src/main/resources/application.properties*.
3. Crear una entidad JPA llamada *Usuario.java*, una interfaz de repositorio de dato JPA Spring llamada *RepositorioUsuario.java*, y un controlador llamado *HomeController.java*
4. Crear un la vista para mostrar la lista de usuarios.
5. Crea una una ClasePuntoEntrada *Aplicacion.java* con el método principal
6. Ejecutar la aplicación.java y dirigirse en el navegador al *localhost:8080/*

Explicamos lo que está pasando:

9.2. Fásil manejo de dependencias

La dependencia llamada *spring-boot-starter-**. este jala todas las librerías comunes al desarrollar aplicaciones Spring MVC tales como: *spring-webmvc*, *jackson-json*, *validation-api*, y *tomcat*.

9.3. Autoconfiguración

Es su `@EnableAutoConfiguration` anote.

9.4. Soporte integrado de contenedor de servlet

Se crea un clase simple comentado con (@SpringApplication),

9.4.1. Spring Initializr

Para comenzar con el proyecto de Spring Boot se deber ir a la página <https://start.spring.io/>, configurar y pulsar generar. Se verifica java para configurar.



The screenshot shows the Spring Initializr web form with the following configuration:

- Project:** ☐ Gradle - Groovy, ☐ Gradle - Kotlin, ☒ Maven
- Language:** ☒ Java, ☐ Kotlin, ☐ Groovy
- Spring Boot:** ☐ 3.0.1 (SNAPSHOT), ☒ 3.0.0, ☐ 2.7.7 (SNAPSHOT), ☐ 2.7.6
- Project Metadata:**
 - Group: com.example
 - Artifact: biblioteca
 - Name: biblioteca
 - Description: Demo project for Spring Boot
 - Package name: com.bib.biblioteca
 - Packaging: ☒ Jar, ☐ War
 - Java: ☐ 19, ☒ 17, ☐ 11, ☐ 8
- Dependencies:**
 - Lombok** **DEVELOPER TOOLS**
Java annotation library which helps to reduce boilerplate code.

Figura 9.1: La configuración al iniciar un proyecto Spring Boot

```
tomas@debian:~$ java --version
openjdk 17.0.4 2022-07-19
OpenJDK Runtime Environment (build 17.0.4+8-Debian-1deb11u1)
OpenJDK 64-Bit Server VM (build 17.0.4+8-Debian-1deb11u1, mixed mode, sharing)
```

9.5. Herramientas de desarrollo

IntelliJ IDEA

IntelliJ IDEA es un Entorno de desarrollo Integrado (EDI) o Integrated Development Environment (IDE) para lenguajes JVM¹ diseñado para maximizar la productividad del desarrollador. Hace la tareas rutinarias y repetitivas proporcionando un completado inteligente de código, análisis de código estático, refactoriza, y permite enfocarse en el lado brillante del desarrollo de software.

IntelliJ IDEA es multiplataforma, que provee una experiencia consistente en Windows, macOS, y Linux. Soporta múltiples lenguajes, herramientas, frameworks, y tecnologías.

IntelliJ IDEA tiene soporte para Spring Boot, se puede crear seleccionando archivo/proyecto/Spring Initializr/siguiente.

IntelliJ IDEA viene en tres ediciones:

- **IntelliJ IDE Ultimate:** Es la edición comercial para JVM, web, y desarrollo corporativo. Incluye todas las características de la edición Community, soporta una variedad de frameworks al lado del servidor así como del front-end, servidor de aplicaciones, integración con base de datos y soporte de herramientas y mucho más.
- **IntelliJ IDEA Community Edition:** Es la edición gratuita basado en Open-source para JVM y desarrollo Android.
- **IntelliJ IDEA Edu:** Es la edición libre con lecciones incorporadas para aprender Java, Kotlin, y Scala. Tiene características especiales para profesores para crear su propio curso y gestionar el proceso de aprendizaje.

La interfaz de usuario sigue el contexto y trae herramientas necesarias de forma automática para ayudar a minimizar el riesgo de interrumpir el flujo del desarrollador.

Lo mejor de IntelliJ IDEA es su configurabilidad, es decir se puede configurar cualquier cosa. Por ejemplo el color del fuente, salida de la consola, depurador, resultados de búsqueda, etc.

Tiene atajos de teclado para toda acción, casi, incluyendo la selección y intercambiar el editor y varias herramientas de windows.

Uno de los más útiles accesos por teclado es el doble shift que abre el dialogo **Buscar en todo**, buscará entre todos los archivos, clases, y símbolos que pertenecen al proyecto e incluso entre las acciones del IDE. Puedes usar

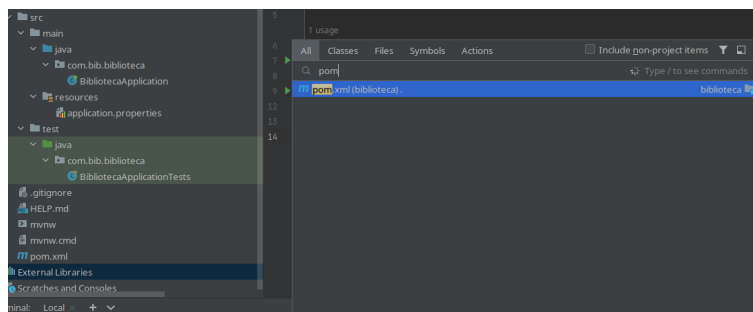


Figura 11.1: Ventana de dialogo de buscar en todo

también esta acción para abrir cualquier herramienta de ventana.

¹Java Kotlin Scala Groovy Clojure Fantom Ceylon Jython JRuby Frege Xtend Golo Concurraas Yeti

11.1. Asistente al codificar

El completado de código ayuda en la velocidad el proceso de codificar. El completado básico ayuda a completar los nombres de las clases, métodos, campos, y palabras reservadas.

El **completado inteligente** sugiere el más relevante símbolo aplicable en el contexto actual, cuando IntelliJ IDEA puede determinar el tipo apropiado.

11.2. Refactorización

IntelliJ IDEA ofrece un comprensible conjunto de refactorización de código que permite una productividad significativa. Por ejemplo cuando renombra una clase, el IDE ayuda a actualizar todas las referencias a esa clase, sin siquiera seleccionar y solo requiere confirmación.

11.3. Terminal

IntelliJ IDEA viene con terminal incorporado para trabajar con comandos de shell desde el IDE. Por ejemplo permite ejecutar comandos de Git. Soporta cmd.exe, bash, sh, etc.

11.4. Herramientas de construcción

IntelliJ IDEA viene con un completo y funcional Gradle y Maven integrado que permite automatizar el proceso de construcción, empaquetado, correr pruebas, despliegue, y otras actividades. IntelliJ IDEA detecta y descarga automáticamente todo los repositorios requeridos y plugins, y no requiere usted configurar nada.

11.5. Control de Versiones

IntelliJ IDEA está integrado con herramientas de control de versiones como, Git, Mercurial, Perforce, y Subversion.

11.6. Requerimientos del sistema e instalación

Se puede ir a la siguiente dirección <https://www.jetbrains.com/idea/download/#section=linux> En Linux

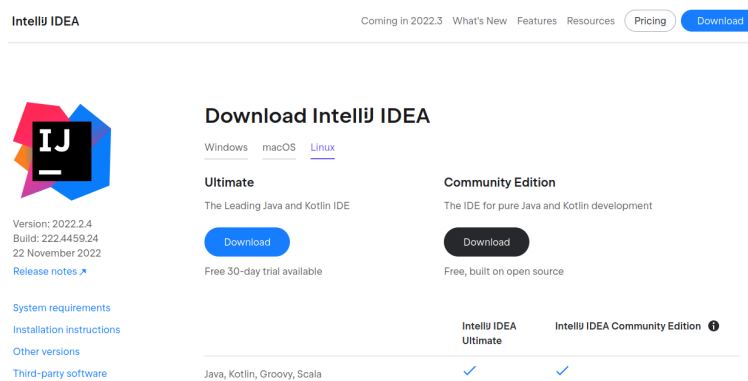


Figura 11.2: Sección de descarga para Linux

cualquier distribución que soporte Gnome, KDE, o Unity DE puede soportarlo. También se puede instalar por línea de comando:

```
sudo snap install intellij-idea-community --classic
or
```

```
sudo snap install intelliJ-idea-ultimate --classic
```

or

```
sudo snap install intelliJ-idea-educational --classic
```

En el caso para el proyecto fue solo necesario descargar el comprimido en .tar, descomprimir y ejecutarlo por un comando el archivo idea.sh en la carpeta donde se descomprimió:

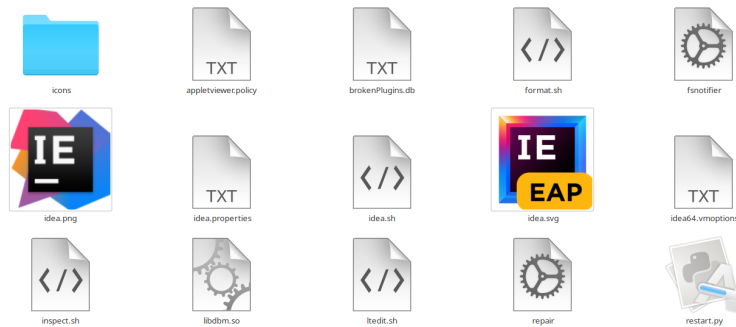


Figura 11.3: Carpeta bin de la carpeta descomprimida

Una vez descomprimido el inicializado de Spring Boot y abierto en el editor IntelliJ IDEA no muestra lo siguiente:

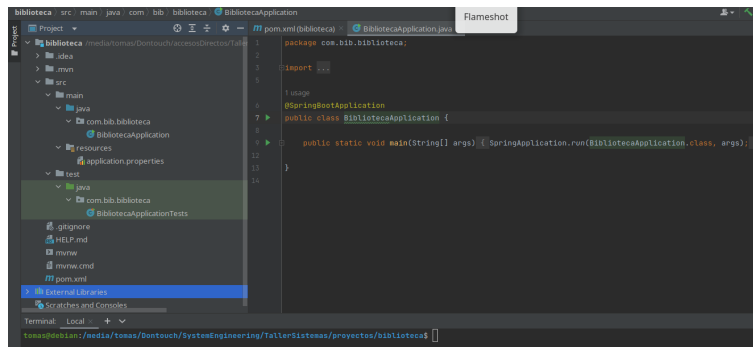


Figura 11.4: El proyecto abierto por primera vez

DBEAVER

DBeaver is un gestor de base de datos universal y profesional. Es capaz de manipular los datos, crear reportes analíticos basado en registros desde diferentes almacenes de base de datos, y exportar la información en un formato apropiado.

DBeaver ofrece:

- Interfaz de usuario cuidadosamente diseñado e implementado.
- Soporte de datos en la nube.
- Soporte estandar para seguridad empresarial
- Capacidad de trabajar con varias extensiones para integración con Excel, Git y otros
- Soporte multiplataforma

DBeaver se utilizó como herramienta de base de datos, se descargó de su página oficial <https://dbeaver.io/> se conecto con dbeaver en puerto 3306 desde debian. La base de datos se llama DBCUERVO Para no tener error al conectar con la base de datos se realiza lo siguiente.

1. `sudo service mysqld stop`
2. `sudo netstat -nap | grep :80`
3. `sudo kill 1433`
4. `/opt/lampp/lampp stop`
5. `/opt/lampp/lampp start`

El comando `sudo service mysqld stop` es necesario para detener mysql y luego en seguida con el comando `sudo netstat -nap | grep :80` se busca que servidor está en ejecución en puerto 80 como sigue `tcp6 0 0 :::80 :::* LISTEN` 1433/apache2 en este caso el proceso 1433 se debe detener con `sudo kill 1433` y en seguida con el comando `sudo /opt/lampp/lampp start` se inicia de nuevo el servidor y tendremos el siguiente resultado:

```
Starting XAMPP for Linux 8.1.10-0...
XAMPP: Starting Apache...fail.
XAMPP: Another web server is already running.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.
```

Aquí se puede ver que Apache no se pudo iniciar porque no ejecutamos el paso 3, repitiendo correctamente es decir desocupando el puerto 80 ya no da lo siguiente:

```
Starting XAMPP for Linux 8.1.10-0...
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...already running.
XAMPP: Starting ProFTPD...already running.
```

Una vez hecho esto se puede iniciar con el gestor de base de datos DBeaver y no da error al conectar. Estos pasos siempre se los hace cada que inicia la computadora debe haber una manera de que el servidor esté corriendo siempre pero por motivos de tiempo no se pudo averiguar.

SWAGGER

Es un conjunto de herramientas para desarrolladores API (application program interface) proveido por la compañía de tecnología *SmartBear Software*.

Fue creado en 2011 por Tony Tam por necesidad

13.1. Swagger UI

La interfaz de usuario de Swagger permite que cualquier persona, visualice e interactúe con los recursos de la API . Se genera automáticamente a partir de su especificación OpenAPI (anteriormente conocida como Swagger), con la documentación visual que facilita la implementación de back-end y el consumo del lado del cliente.

Diccionario de datos

Nombre del archivo: Documento Fecha de creación 10 de diciembre de 2022
 Descripción: Cualquier objeto digital que requiera el estudiante

Campo	Tamaño	Tipo de dato	Descripción
idDoc	6	N Numérico	Clave única de registro de documento
titulo	500	C Caracter	Título del documento digital
lenguaje	20	C Caracter	Lengua en el que está el documento
ocupa	5	N Numérico	Cantidad que ocupa en un dispositivo de almacenamiento
descripcion	200	C Caracter	Breve descripción del documento
fechaReg	x	D Date	Es la fecha en que se registró el documento
Relaciones: idDocumento con idAutor		Campos clave: idDocumento	

Nombre del archivo: Académico Fecha de creación: 10 de diciembre de 2022
 Descripción: Base de datos que contiene documentos académicos o científicos

Campo	Tamaño	Tipo de dato	Descripción
idAcademico	6	N Numérico	Clave única de registro de documento académico
doi	300	C Caracter	Identificador digital de documento
paginas	x	N Numérico	Número de páginas que tiene el documentos académico
ano	4	n numérico	Es el año que se publicó el documento
Relaciones: idAcademico con idDocumento		Campos clave: idAcademico y doi	

Nombre del archivo: Libro Fecha de creación: 10 de diciembre de 2022
 Descripción: Base de datos que contiene libros

Campo	Tamaño	Tipo de dato	Descripción
idLibro	18	N Numérico	Clave única de registro de documento académico
edicion	3	N Numérico	Es el número de edición del libro
Editorial	20	C Caracter	Es el editorial publicador del libro
Relaciones: idLibro con idAcademico		Campos clave: idLibro	

Nombre del archivo: Articulo Fecha de creación: 10 de diciembre de 2022
 Descripción: Base de datos que contiene artículos

Campo	Tamaño	Tipo de dato	Descripción
idArticulo	18	N Numérico	Clave única de registro artículos
revista	200	C Caracter	Es la revista que publica el artículo
Relaciones: idLibro con idDocumento		Campos clave: idArticulo	

Nombre del archivo: Autor Fecha de creación: 10 de diciembre de 2022
 Descripción: Base de datos que contiene autores

Campo	Tamaño	Tipo de dato	Descripción
idAutor	18	N Numérico	Clave única de registro de un autor
nombreAutor	200	C Caracter	Es el nombre del autor
Relaciones: idAutor con idDocumento		Campos clave: idAutor	

Descripción de API REST

Metodo http	GET
url	http://kuerva:2295/org/biblioteca2295/almacen
Descripción	Es servicio lista todos los libros de la biblioteca digital 2295.

Bibliografia

[his,] Java SE versions history.