

Manual de operación

Índice de Contenido

- [Introducción](#)
- [Objetivos del sistema](#)
- [Descripción de procesos](#)
- [Diagrams UML](#)
- [Modelo relacional de la base de datos](#)
- [Descripción de la plataforma](#)
- [Requerimientos para la instalación de la aplicación](#)
- [instalación de la aplicación](#)
- [instalación de la base de datos](#)
- [verificación de la instalación](#)
- [Documentación del código fuente](#)

Introducción

El manual de operación de el software Neo Library tiene como propósito ilustrar sobre la definición, diseño, organización y estructura del sistema o solución al personal encargado de mantener la prestación del servicio o servicios ofrecidos por el sistema o solución, estos lectores incluyen desarrolladores, arquitectos, ingenieros de pruebas etc.

Objetivos del sistema

Objetivo general

Diseñar y desarrollar un sistema de información que permita tener un control eficiente, confiable y accesible en las bibliotecas por medio de una interfaz clara y sencilla. con el cual se quiere llevar un monitoreo de los préstamos a través de los datos personales de los usuarios para evitar acciones o situaciones que entorpezcan el proceso.

Objetivos específicos

Complementar el sistema de seguridad de las bibliotecas, facilitando la verificación de datos de los usuarios.

Garantizar la seguridad de los datos personales de los usuarios en las bibliotecas.

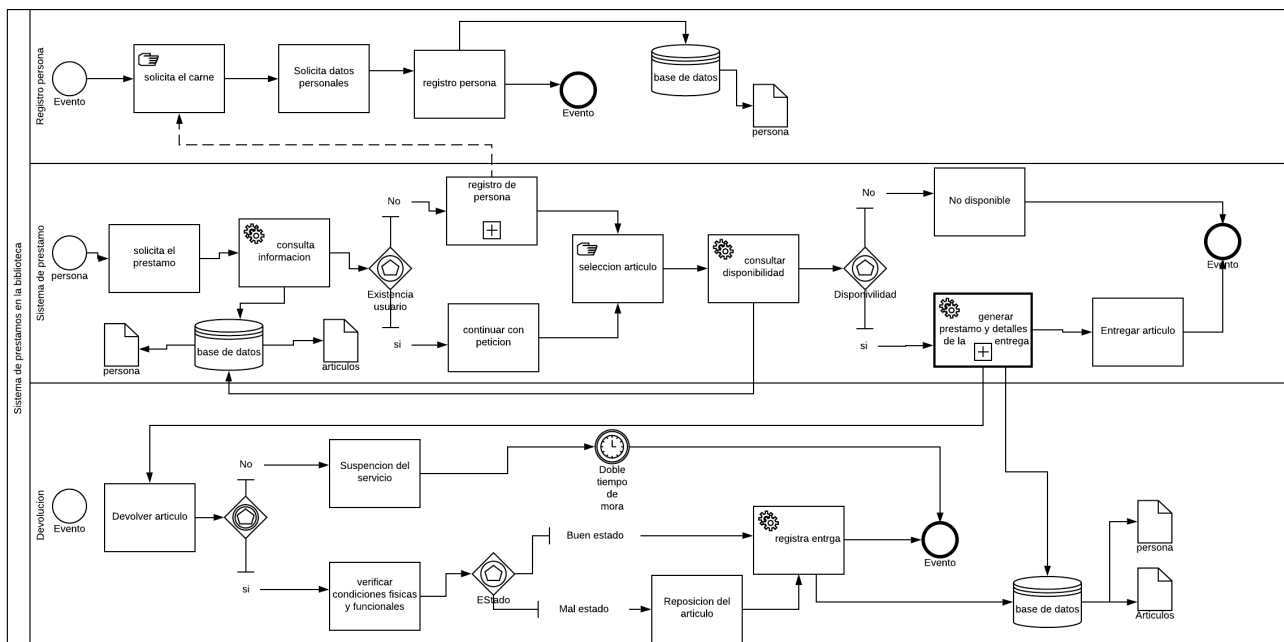
Agilizar el proceso de préstamos en las bibliotecas garantizando una mejor atención en el proceso.

Crear una interfaz grafica que sea sencilla y responsiva generando así un fácil uso del aplicativo.

Descripción de procesos

Mapa de procesos (BPMN)

Se muestra el diagrama de BPMN que describe los procesos realizados dentro de la biblioteca enfocándose en el funcionamiento del sistema.



BPMN

Se muestra el diagrama de BPMN que describe los procesos realizados dentro de la biblioteca enfocándose en el funcionamiento del sistema.

Requerimientos

Funcionales

- Registro de usuario

- Consulta de usuario
- Consulta de inventario
- Actualización de inventario
- Registro de préstamos

No Funcionales

- Interfaz del sistema
- Confiabilidad continua del sistema
- Desempeño
- Ayuda en el uso del sistema

Diagrams UML

Diagrama de casos de uso

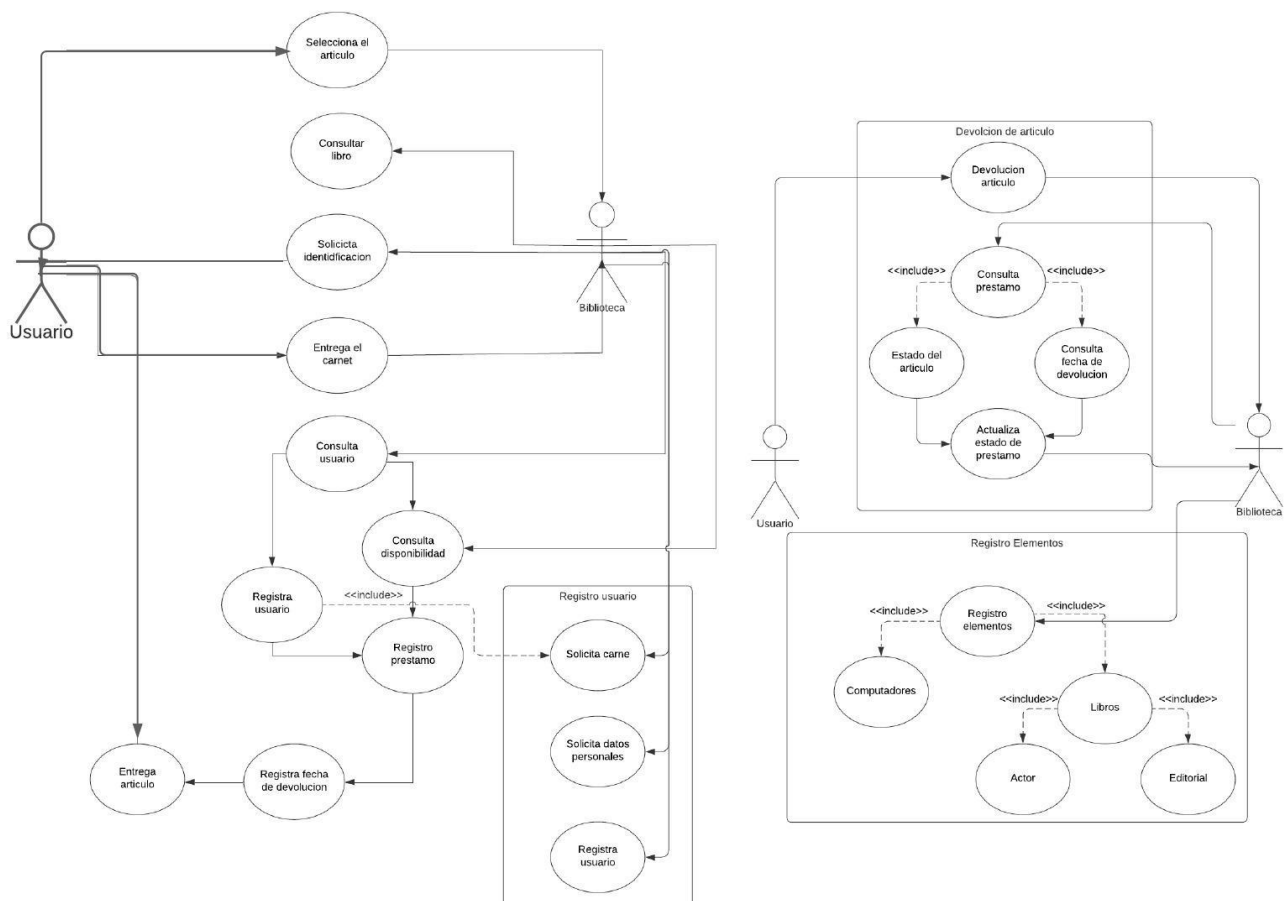


Imagen de casos de uso (debe ser cambiado)

Diagrama de clases

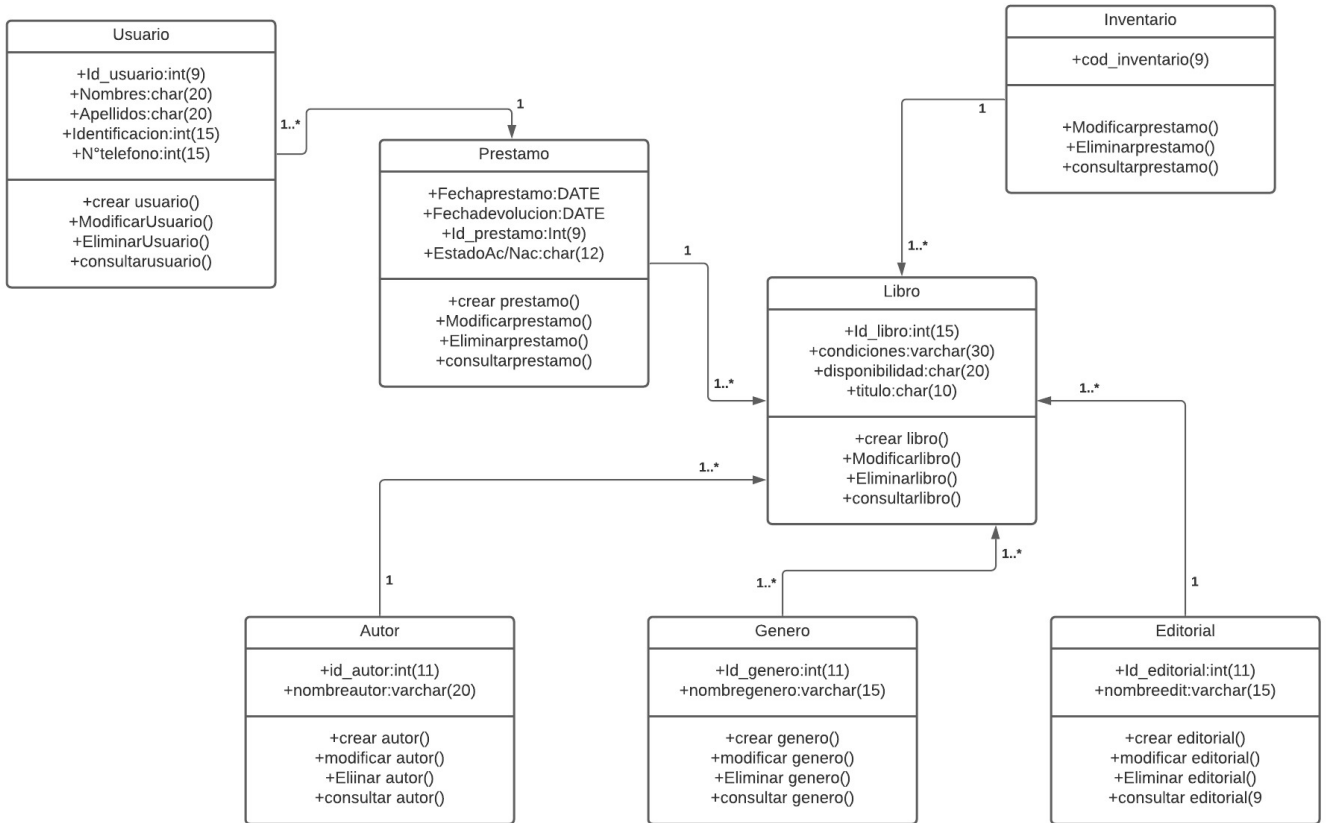
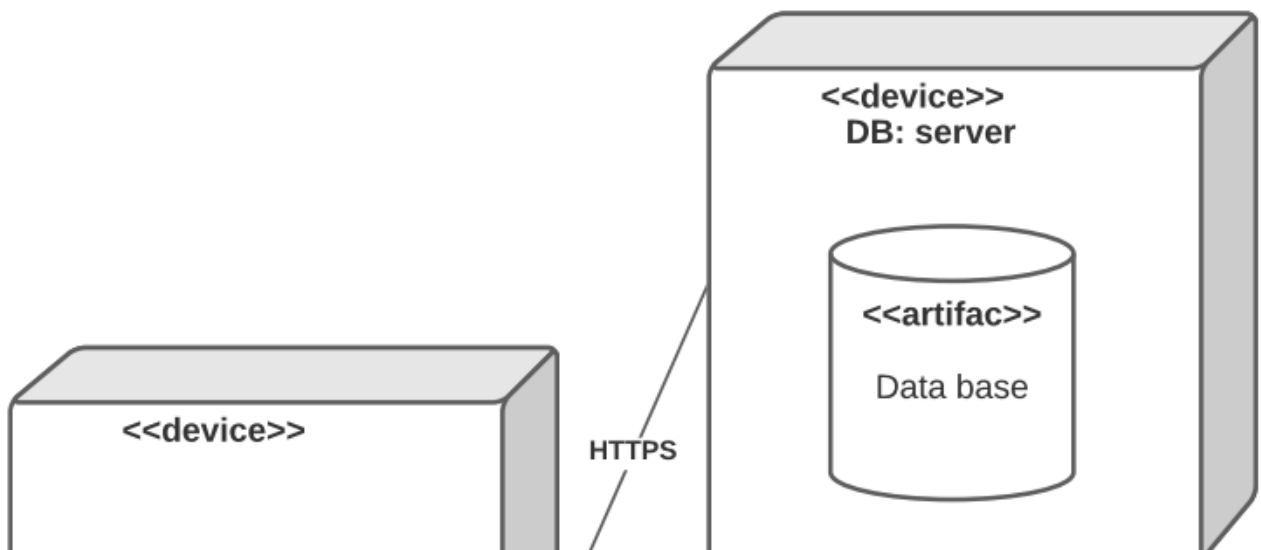


Imagen del diagrama de clases

Diagrama de distribución



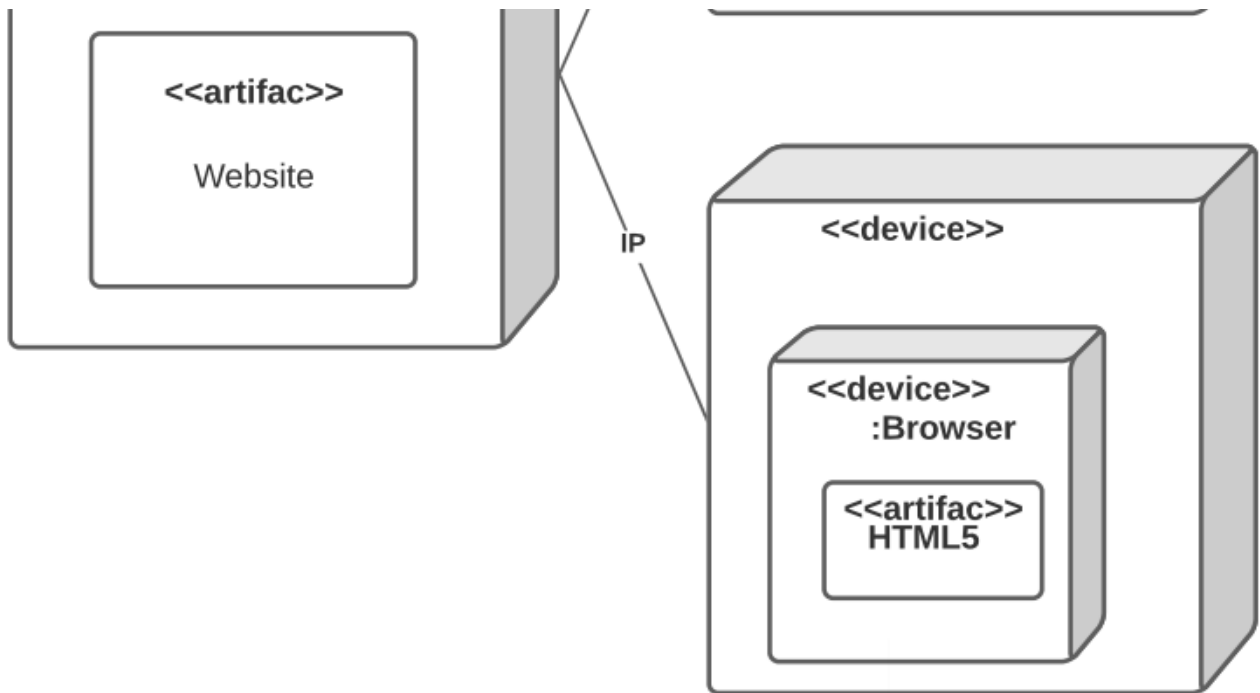


Imagen del diagrama de distribución

Administración de usuarios

Super user

Para la creación del súper usuario que es con el cual posteriormente se pueden crear usuarios alterar información sobre estos y asignar roles.

se debe ingresar el comando

```
1 python manage.py createsuperuser
```

Luego de esto se puede ingresar a al panel de administración con las credenciales del super usuario

Panel de administración

Administración Neo Library

Nombre de usuario:

Contraseña:

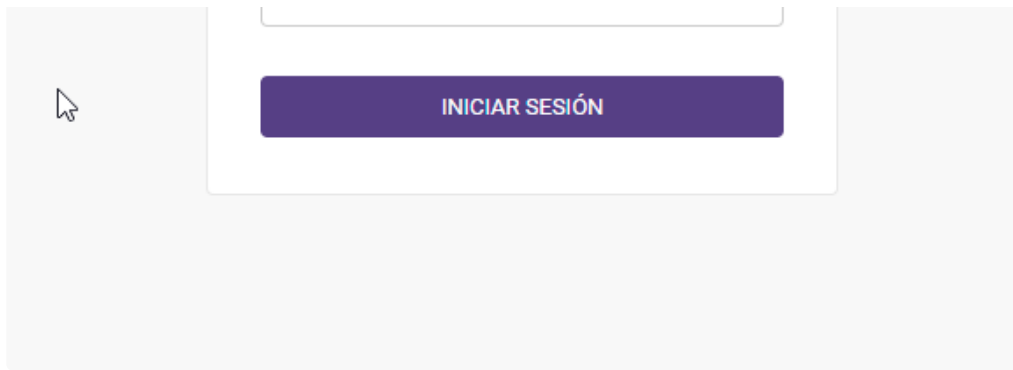


Imagen del inicio de sesión en el panel de control

Administración Neo Library

[BIENVENIDO/A, MAEST.](#) [VER EL SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [TERMINAR SESIÓN](#)

Sitio administrativo

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos

+ Añadir

✎ Modificar

Usuarios

+ Añadir

✎ Modificar

INTERFAZ DE ADMINISTRACIÓN

Temas

+ Añadir

✎ Modificar

PRESTAMOS

Autores

+ Añadir

✎ Modificar

Editoriales

+ Añadir

✎ Modificar

Géneros

+ Añadir

✎ Modificar

Libros

+ Añadir

✎ Modificar

Préstamos

+ Añadir

✎ Modificar

Tipos de Documentos

+ Añadir

✎ Modificar

Usuarios

+ Añadir

✎ Modificar

Acciones recientes

Mis acciones

+ 10025245 paco deluque

Usuario

+ pato azul

Editorial

+ ein buch

Libro

+ gnero 1

Género

+ cc

Tipo de Documento

✎ simple dark

Tema

✎ simple dark

Tema

✎ simple dark

Tema

✎ simple dark

Tema

✎ simple dark

Tema

Vista general del panel de administración

Creación de usuarios

Una vez en el panel de administración se debe dirigir a la pestaña de usuarios desde la cual se podrán llevar las acciones de añadir, modificar y eliminar usuarios

«

Escoja usuario a modificar

AÑADIR USUARIO +

Q

Buscar

Acción:

Ir

seleccionados 0 de 1

☐

NOMBRE DE USUARIO

DIRECCIÓN DE CORREO ELECTRÓNICO

NOMBRE

APELLIDOS

ES STAFF

☐

maest

1 usuario

FILTRO

Por es staff

Todo

Por es superusuario

Todo

Por Activo

Todo

Vista general de usuarios

Permisos

☒ Activo
Indica si el usuario debe ser tratado como activo. Desmarque esta opción en lugar de borrar la cuenta.

☒ Es staff
Indica si el usuario puede entrar en este sitio de administración.

☒ Es superusuario
Indica que este usuario tiene todos los permisos sin asignárselos explícitamente.

Grupos:

grupos Disponibles ?

Selecciona todos ?

grupos elegidos ?

Eliminar todos

Los grupos a los que pertenece este usuario. Un usuario tendrá todos los permisos asignados a cada uno de sus grupos. Mantenga presionado "Control" o "Comando" en una Mac, para seleccionar más de uno.

Permisos de usuario:

permisos de usuario Disponibles ?

admin | entrada de registro | Can add log entry
admin | entrada de registro | Can change log entry
admin | entrada de registro | Can delete log entry
admin | entrada de registro | Can view log entry

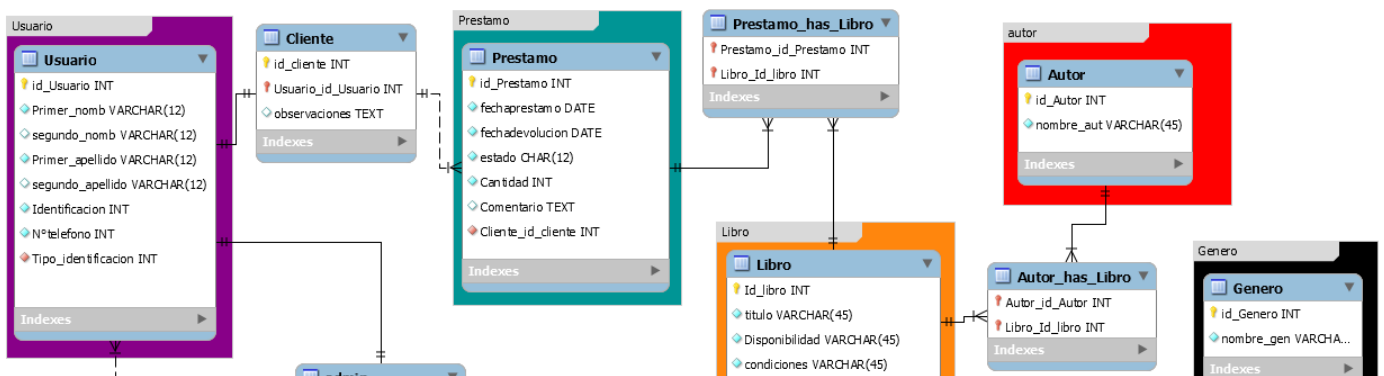
Vista de los paneles de permisos

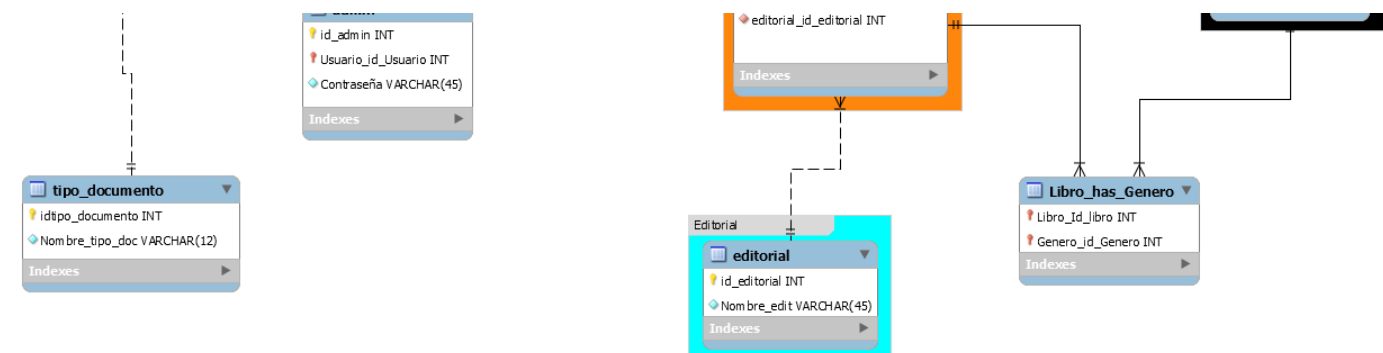
permisos de usuario elegidos ?

Luego de ingresar a la información de un usuario desde allí se pueden asignar los roles, perfiles y funciones del usuario

Modelo relacional de la base de datos

Modelo entidad relación





Modelo entidad-relación (crow's foot)

Diccionario de datos

Autor												
Column name	DataType	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment		
id_Autor	INT	✓	✓							Almacena el codigo que identifica al autor del libro		
nombre_aut	VARCHAR(45)		✓							almacena el nombre del autor del libro		

Genero												
Column name	DataType	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment		
id_Genero	INT	✓	✓							almacena el codigo que identifica la genero		
nombre_gen	VARCHAR(45)		✓							almacena el nombre de cada genero		

Libro												
Column name	DataType	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment		
Id_libro	INT	✓	✓							Almacena el codigo que identifica a cada libro en el sistema		
titulo	VARCHAR(45)		✓							Almacena el titulo del libro		
Disponibilidad	VARCHAR(45)		✓							almacena la informacion sobre si un libro esta disponible o no		
estado	VARCHAR(45)		✓							almacena un pequeño comentario sobre las condiciones físicas del libro		
editorial_id_editorial	INT		✓									
Autor_id_Autor	INT		✓									

Libro_has_Genero												
Column name	DataType	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment		
Libro_Id_libro	INT		✓			✓						
Genero_id_Genero	INT		✓			✓						

Prestamo												
Column name	DataType	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment		
id_Prestamo	INT	✓	✓							almacena elcodigo que identifica el prestamo		
fechaperstamo	DATE		✓							almacena la fecha en la que se realiza el prestamo		
fecha devolucion	DATE		✓							almacena la fecha en la que se realiza la la devolucion del prestamo		
estado	CHAR(12)		✓							almacena la informacion sobre si el prestamo esta en estado activo o no		
Cantidad	INT		✓							Almacena la cantidad de libros de cada prestamo		
Usuario_id_Usuario	INT		✓									

Prestamo_has_Libro												
Columna name	DataType	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment		
Prestamo_id_Prestamo	INT		✓			✓						
Libro_Id_libro	INT		✓			✓						

Usuario												
Columna name	DataType	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment		
id_Usuario	INT	✓	✓							almacena el codigo identica al usuario		
Nombres	CHAR(45)		✓							almacena los nombres del usuario		
Apellidos	CHAR(45)		✓							Almacena los apellidos del usuario		
Identificacon	INT		✓							almacena el numero de identificacion del usuario		
Nºtelefono	INT		✓							almacena el numero de telefono del usuario		

editorial												
Column name	DataType	PK	NN	UQ	BIN	UN	ZE	AI	Default	Comment		
id_editorial	INT	✓	✓							almacena el codigo que identifica a cada editorial		
Nombre_edit	VARCHAR(45)		✓							almacena el nombre de la editorial		

Imagen del diccionario de datos (Generada en MySQL workbench por medio de un plugin)

Descripción de la plataforma

El proyecto fue desarrollado en el framework Django que está basado en lenguaje de programación python, para la creación del proyecto se instaló Django en un entorno virtual creado con python al cual también se le le añadieron los siguientes paquetes:

- psycopg2
- mysqlclient
- Pillow

psycopg2 y mysqlclient para hacer posible la conexión con los gestores de bases de datos de PostgreSQL y MySQL respectivamente y pillow para el manejo de imágenes en la base de datos

Todo esto hace posible que el proyecto sea compatible con tres gestores de bases de datos, ya que por defecto Django es compatible SQLite3.

Fue escrito con la versión 3.9 32 bits de python y la versión 3.2 de django

Requerimientos para la instalación de la aplicación

Sistema operativo	Windows * Server, Linux * o cualquier sistema operativo que se pueda ejecutar como servidor Web, capaz de ofrecer contenido HTML5, incluyendo JSON y MP4.
Procesador	Procesador Intel® Celeron® 847, 1,10 GHz o equivalente.
Almacenamiento	Entre 1,3 GB-2,3 GB dependiendo de la versión del idioma.
Ram	Mínimo de 1,5GB. La cantidad recomendada puede variar dependiendo del número de usuarios conectados, el número de sitios web alojados en dispositivo, y otros factores.
Disco duro	3 GB de espacio disponible en el disco duro para la instalación; durante la instalación se requiere espacio libre adicional. No se puede instalar en dispositivos de almacenamiento flash extraíbles.

Instalación de la aplicación

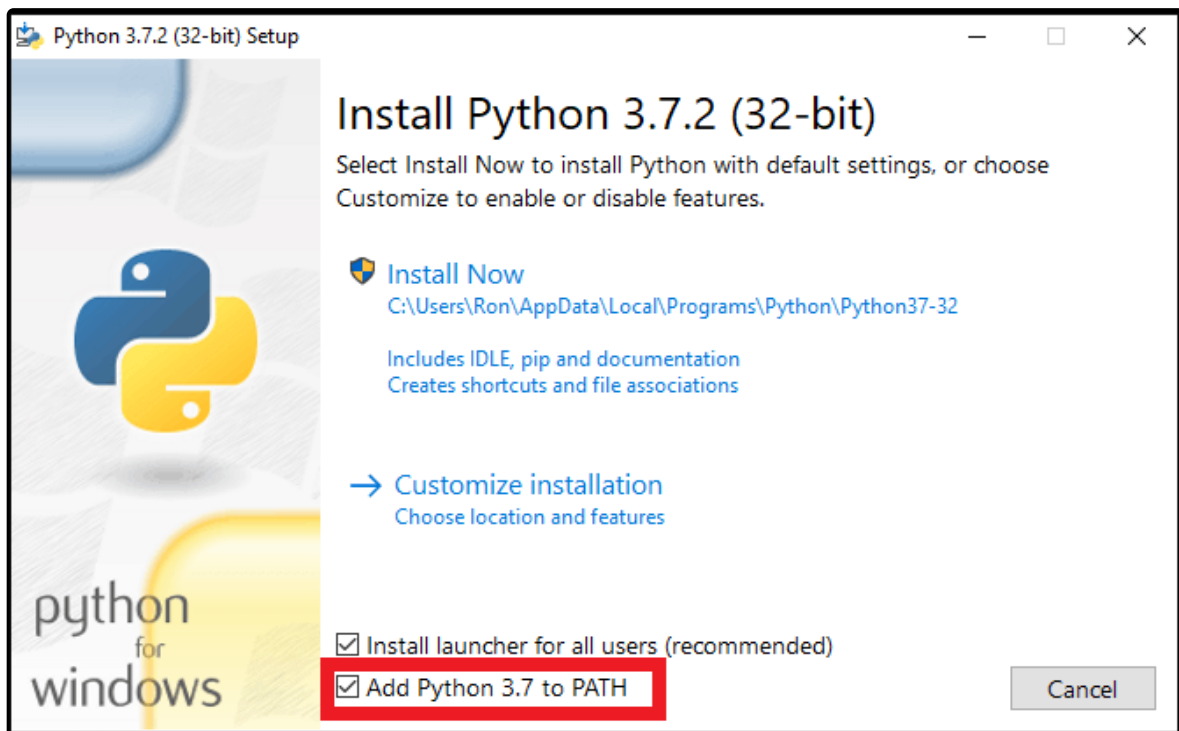
Instalación de python

lo primero que se debe hacer es instalar python (3.9)



Link de descarga en la página oficial de python

Instalar python como administrador y añadir a el PATH



Creación del entorno virtual

Desde la consola de comandos dirigirse a la consola a la carpeta donde se quiere crear el entorno virtual o abrir la consola desde esa carpeta (si se instaló python con el path) se ingresa el siguiente código:

```
1 c:\>python -m venv venv
```

Donde el primer venv es la declaración para crear el entorno virtual y el segundo venv es el nombre que se le da al entorno virtual y podría ser cualquier otro

Iniciar el entorno virtual

Una vez creado el entorno virtual procedemos a activarlo para poder usarlo desde la consola de comandos nos dirigimos al interior de la carpeta venv con el comando cd luego a scripts.

```
1 cd venv
2
3 cd scripts
```

Si estamos en powerShell usamos .\activate para activar el entorno virtual pero si estamos en el "símbolo de sistema" o "default" se usa activate.bat

```
1 .\activate
2
3 activate.bat
```

Sabemos que el entorno virtual esta activado cuando aparece (venv) antes de la dirección de la carpeta donde estamos

```
1 (venv)c:\>
```

y para desactivarlo solo hay que escribir deactivate y veremos que ya no aparece (venv) antes de la dirección de donde nos encontramos

```
1 (venv)c:\>deactivate
2
3 c:\>
```

Instalación de Django y paquetes

pip install

Una vez instalado python la manera más rápida, sencilla y confiable de instalar las versiones adecuadas es hacer uso del archivo requirements.txt que se encuentra en la carpeta del proyecto.

Usando la consola de comandos del sistema

```
1 pip install requirements.txt
```

de esta manera ya se habrá instalado Django y todos los paquetes necesarios del proyecto.

paso a paso

Si, no se encuentra el archivo de requirements.txt o se desea hacer la instalación de cada componente por separada se hace de la siguiente manera:

Todos se instalan desde la consola

Django:

```
1 pip install Django==3.2.3
```

paquetes adicionales para el manejo de base de datos y de imágenes:

```
1 pip install psycopg2
```

```
1 pip install pillow
```

```
1 pip install mysqlclient
```

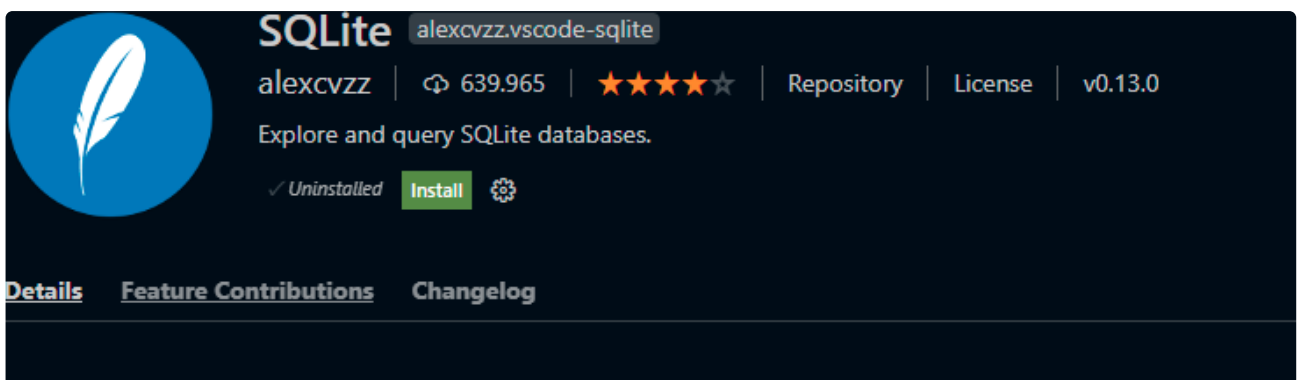
Instalación de la base de datos

Instalación gestores de bases de datos

SQLite

Como este proyecto ha sido desarrollado a través de visual code se instaló SQLite que es una extensión para este y permite gestionar la base de datos desde visual code.

Su instalación es sencilla solamente hay que dirigirse a la página de la extensión desde visual code y pulsar en instalar



captura desde visual code

MySQL workbench

Para instalar MySQL workbench se descarga desde su página oficial

General Availability (GA) Releases
Archives

MySQL Workbench 8.0.25

Select Operating System:

Microsoft Windows

Recommended Download:


MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.


Windows (x86, 32 & 64-bit), MySQL Installer MSI

Go to Download Page >



Other Downloads:

Windows (x86, 64-bit), MSI Installer	8.0.25	42.2M	Download
<small>(mysql-workbench-community-8.0.25-winx64.msi)</small>		<small>MD5: 4220a115ad93e4caa7e9bcb7b08906e Signature</small>	



We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

Captura pagina de descarga MySQL

Una vez descargado se ejecuta el programa de instalación y se instala con la configuración deseada

Si necesita más ayuda sobre la instalación y configuración se puede dirigir a

<https://dev.mysql.com/doc/workbench/en/wb-installing-windows.html#wb-windows-standalone>

PostgreSQL

Para instalar PostgreSQL workbench se descarga desde su página oficial

Downloads

PostgreSQL Downloads

PostgreSQL is available for download as ready-to-use packages or installers for various platforms, as well as a source code archive if you want to build it yourself.

Packages and Installers

Select your operating system family:

Linux

macOS

Windows

BSD

Solaris

Una vez descargado se ejecuta el programa de instalación y se instala con la configuración deseada

Si necesita más ayuda sobre la instalación y configuración se puede dirigir a

<https://www.postgresql.org/docs/12/tutorial-install.html>

configuración de la base de datos

Para la configuración de la base de datos son necesarios los paquetes de :

```
1 psycopg2
```

```
1 mysqlclient
```

que son los que hacen posible la conexión a las bases de datos de PostgreSQL y mysql luego en la carpeta del proyecto se crea un archivo llamado db

```
1 import os
2
3 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
4
5 SQLITE = {
6     'default': {
7         'ENGINE': 'django.db.backends.sqlite3',
8         'NAME': os.path.join(BASE_DIR, 'db.sqlite3')
9     }
10 }
11
12 POSTGRESQL = {
13     'default': {
14         'ENGINE': 'django.db.backends.postgresql_psycopg2',
15         'NAME': 'Neo_library',
16         'USER': 'postgres',
17         'PASSWORD': '12345',
18         'HOST': 'localhost',
19         'PORT': '5432'
20     }
21 }
22
23 MYSQL = {
24     'default': {
25         'ENGINE': 'django.db.backends.mysql',
26         'NAME': 'NEO_LIBRARY',
27         'USER': 'root',
28         'PASSWORD': '12345',
29         'HOST': 'localhost',
30         'PORT': '3306'
31     }
32 }
```

El cual contiene la configuración y credenciales para conectarse a las distintas bases de datos

luego en el archivo settings.py cambiar la base de datos

```

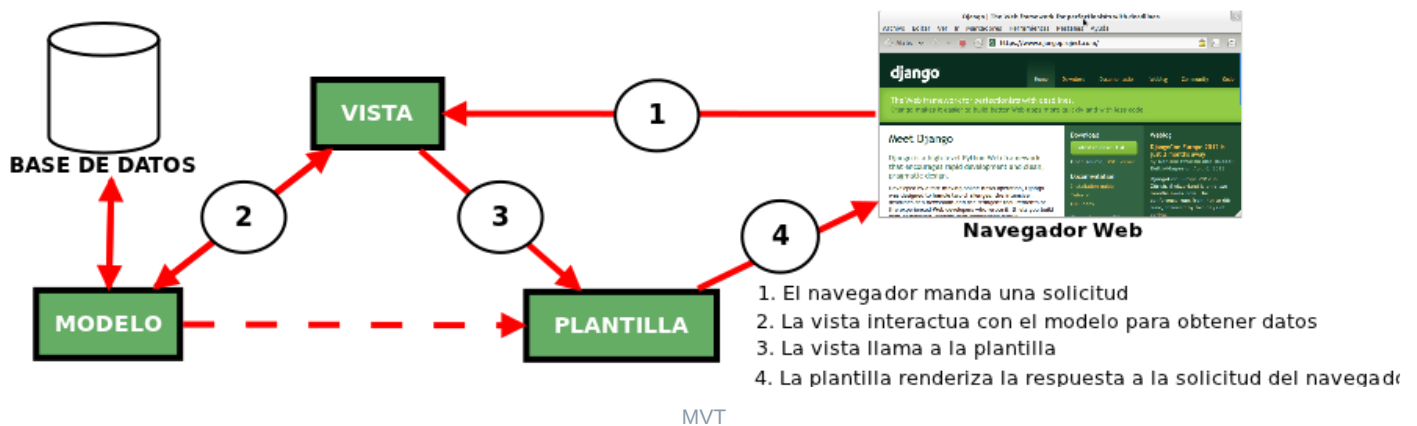
1 # Database
2 # https://docs.djangoproject.com/en/3.2/ref/settings/#databases
3
4 DATABASES = {
5     'default': {
6         'ENGINE': 'django.db.backends.sqlite3',
7         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
8     }
9 }

```

Esquema de trabajo

El esquema de trabajo de Django es MVT model view template

1. El navegador manda una solicitud
2. La vista interactúa con el modelo para obtener datos
3. La vista llama a la plantilla
4. La plantilla renderiza la respuesta a la solicitud del navegador



Usuarios

Una vista sobre la tabla de usuarios en el sistema, donde también se puede ver algunas acciones disponibles en esta.

AUTENTICACIÓN Y AUTORIZACIÓN	
Grupos	+ Añadir
Usuarios	+ Añadir

tabla de usuarios

«

Escoja usuario a modificar

+

AÑADIR USUARIO

Q

Buscar

Acción:

Ir

seleccionados 0 de 1

<input type="checkbox"/>	NOMBRE DE USUARIO	DIRECCIÓN DE CORREO ELECTRÓNICO	NOMBRE	APELLIDOS	ES STAFF
<input type="checkbox"/>	maest				✓

1 usuario

FILTRO

Por es staff

Todo

Por es superusuario

Todo

Por Activo

Todo

verificación de la instalación

pip freeze

Para verificar la instalación adecuada se puede hacer uso de pip freeze desde la consola de comandos

```
1 py -m pip freeze
```

que desplegara una lista de todos los programas y paquetes instalados en python

Aparecerán programas que están instalados por defecto, pero lo que nos importa es que muestra los programas que se han instalado previamente

Runserver

Para comprobar que se ha creado correctamente el proyecto se puede hacer lo siguiente:

1. Ingresar con cd a la carpeta del proyecto
2. y desde allí se ingresa el comando:

```
1 python manage.py runserver
```

Lo cual inicia un server para poder trabajar en nuestro proyecto y ver como va quedando, la primera vez que se abre y no se ha hecho ningún cambio en el proyecto aparece una pagina con el aviso de que Django fue instalado correctamente.

django

View [release notes](#) for Django 3.2



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.



Django Documentation
Topics, references, & how-to's



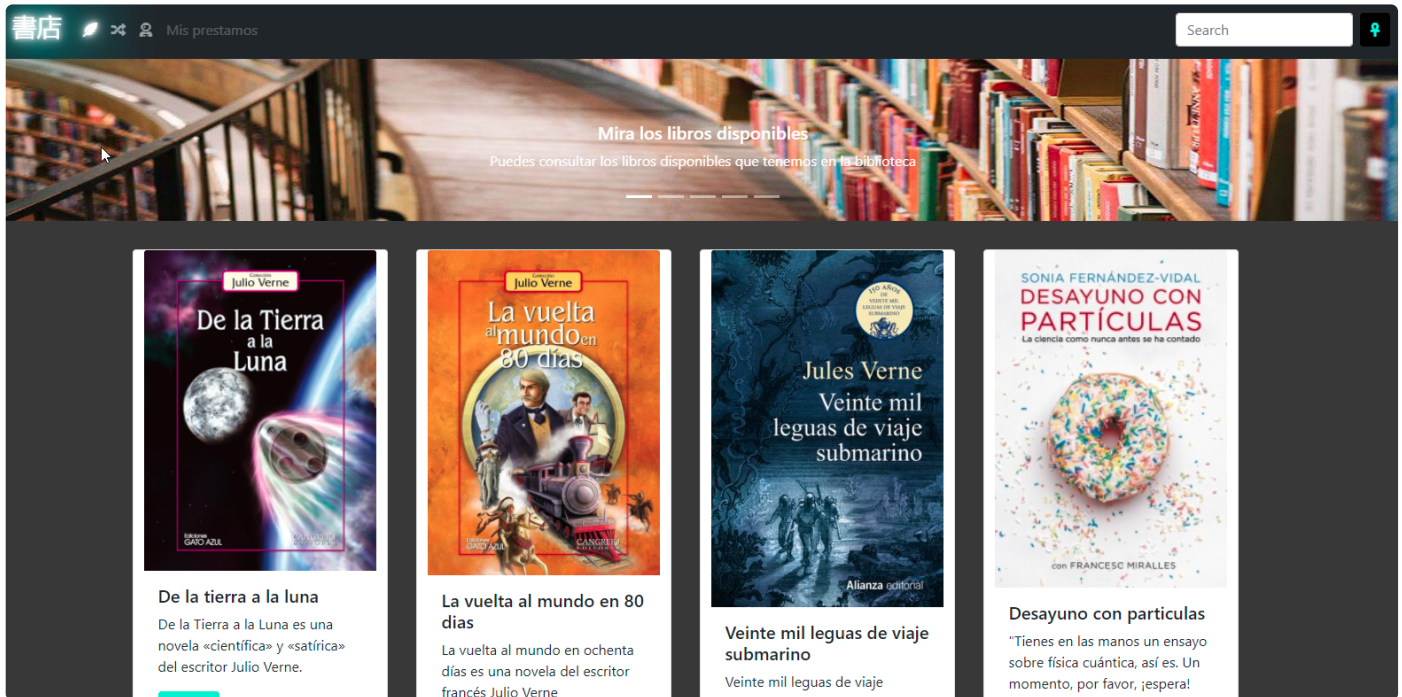
Tutorial: A Polling App
Get started with Django



Django Community
Connect, get help, or contribute

Django welcome page

y en nuestro proyecto ya configurado aparecerá la página index



index del proyecto

Documentación del código fuente

Estructura de la aplicación

Una vez creado el proyecto se crea una carpeta que dentro contiene una carpeta con el mismo nombre y el archivo `manage.py` y a su vez se crean archivos dentro de esa carpeta

```
1 Neo_library/
2   manage.py
3   Neo_library/
4     __init__.py
5     settings.py
6     urls.py
7     asgi.py
8     wsgi.py
```

Luego de creada una aplicación se crea una carpeta con el nombre que le hayamos puesto

```
1 Neo_library/
```

```
2    manage.py
3    Neo_library/
4        __init__.py
5        settings.py
6        urls.py
7        asgi.py
8        wsgi.py
9        prestamos
```

y dentro de esta estarán los siguientes archivos

```
1 prestamos/
2     __init__.py
3     admin.py
4     apps.py
5     migrations/
6         __init__.py
7     models.py
8     tests.py
9     views.py
```

luego dentro la carpeta de la aplicación se crearán dos carpetas templates y static

```
1 prestamos/
2     __init__.py
3     admin.py
4     apps.py
5     migrations/
6     templates
7     static
8         __init__.py
9     models.py
10    tests.py
11    views.py
```

templates y static son carpetas creadas por nosotros para almacenar y hacer posible la renderización de las páginas, dentro de templates estarán los archivos HTML

En la carpeta static están los archivos estáticos como las imágenes archivos CSS y json

Los archivos más importantes son:

settings.py:

Que es archivo donde se añadirá la configuración para la conexión en la base de datos, se incluirá la aplicación sección de installed apps, se cambiara la zona horaria el idioma y algunas configuraciones extra.

views.py:

El archivo donde se configurará la vista de las templates y algunas funciones

Urls.py:

Es el archivo en el cual se crean las URLs que maneja el proyecto

Models.py:

En models.py es donde el archivo se crean los esquemas para la base de datos el equivalente a la definición de la base de datos

Admin.py:

El archivo donde se registraran las tablas que se quieren mostrar en el panel de administración

db.py:

Un archivo creado para hacer la configuración para distintas bases de datos

Estos son los archivos mas importantes y que mas cambios han tenido durante el desarrollo