

# MANUAL TÉCNICO

# Introducción

El siguiente documento describe los aspectos técnicos de la aplicación NEO LIBRARY, describiendo distintos aspectos de esta para facilitar su entendimiento y posible mantenimiento.

# Alcance

## 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.

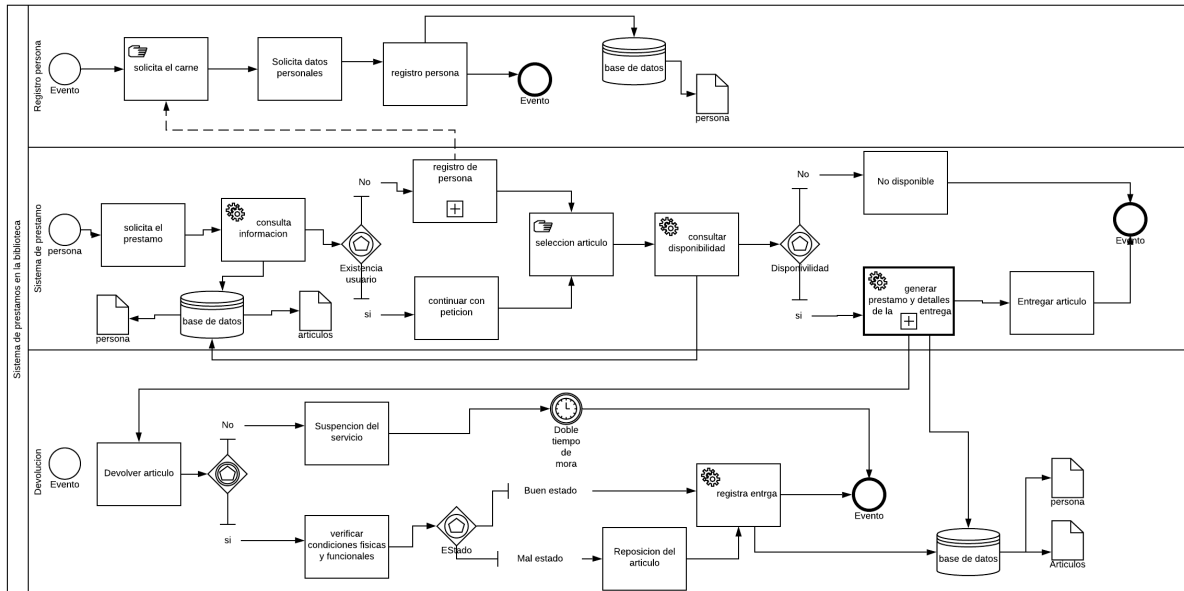
## Objetivos específicos

- 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 gráfica que sea sencilla y responsiva generando así un fácil uso del aplicativo.

En este documento no se incluyen costo y tiempo de desarrollo ni los pasos de instalación de la aplicación

# Descripción de procesos

## Mapa de procesos



Imágenes del mapa de procesos

## Requerimientos

### Funcionales

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

## No funcionales

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

# Diagramas UML

## Diagrama de casos de uso

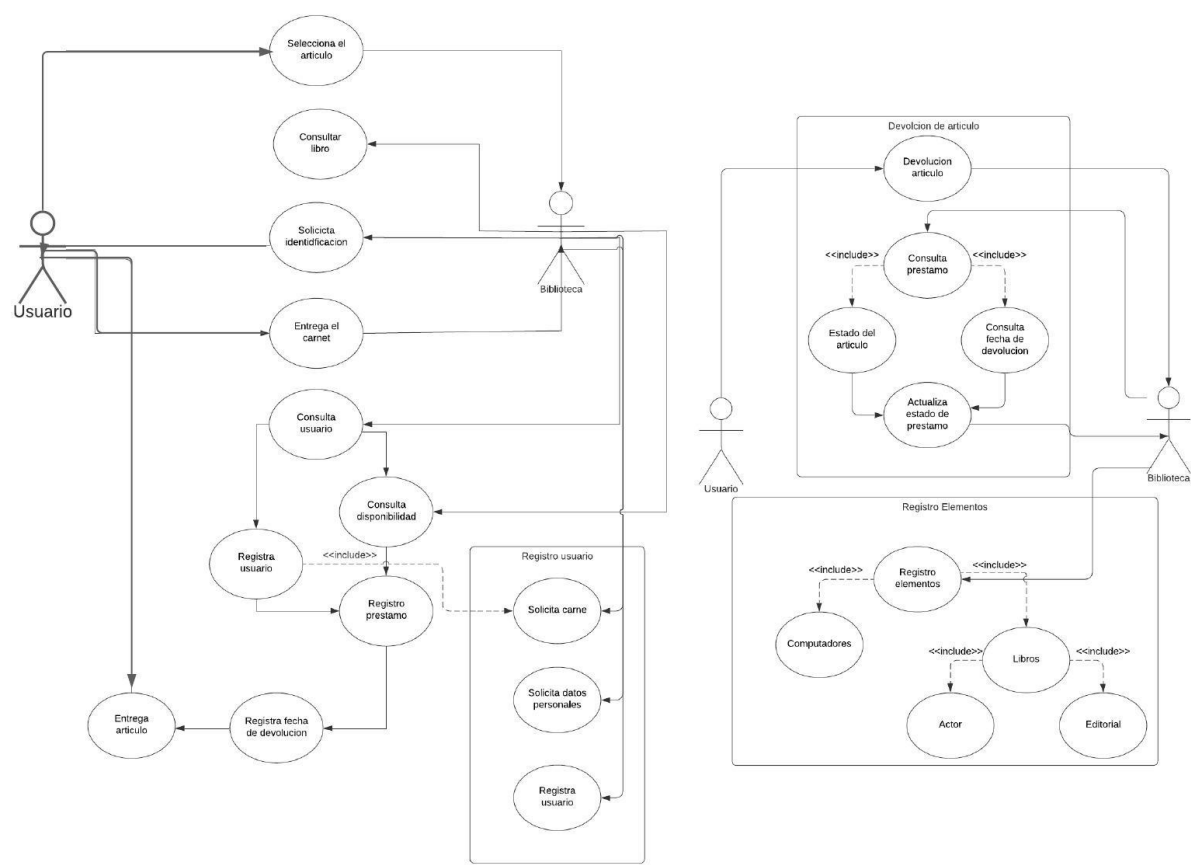


Imagen de casos de uso (debe ser cambiado)

## Diagrama de clases

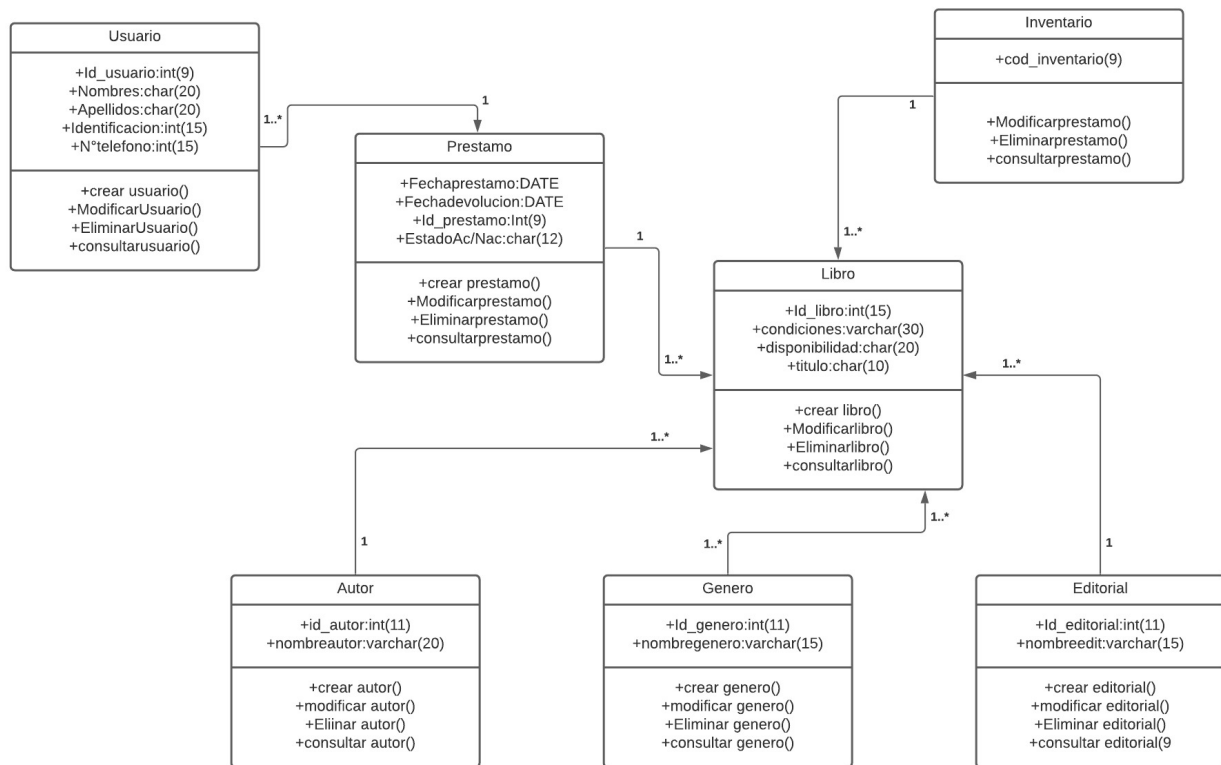


Imagen del diagrama de clases

## Diagrama de distribución

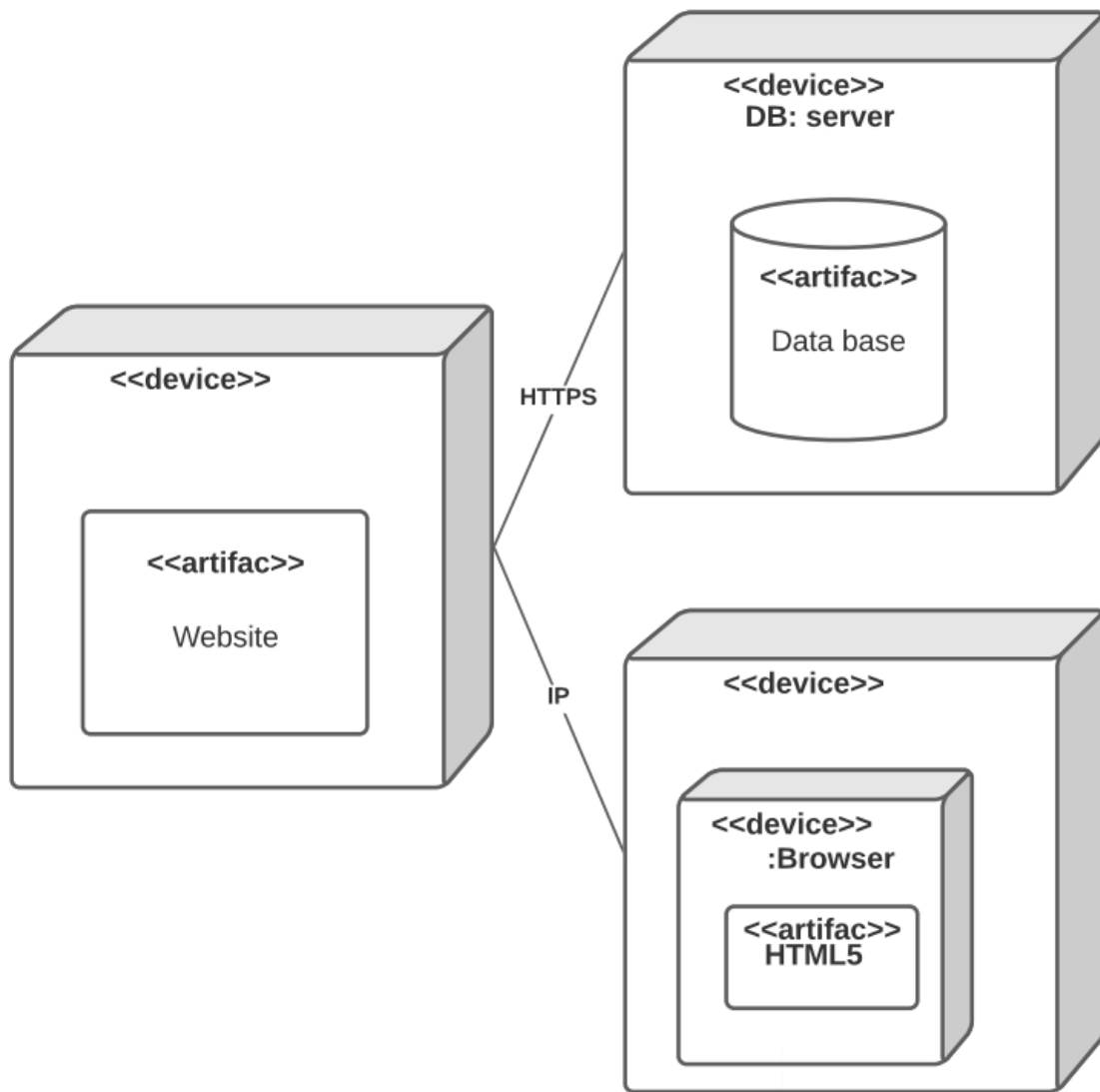


Imagen del diagrama de distribución



# 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

```
python manage.py createsuperuser
```

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

## Panel de administración

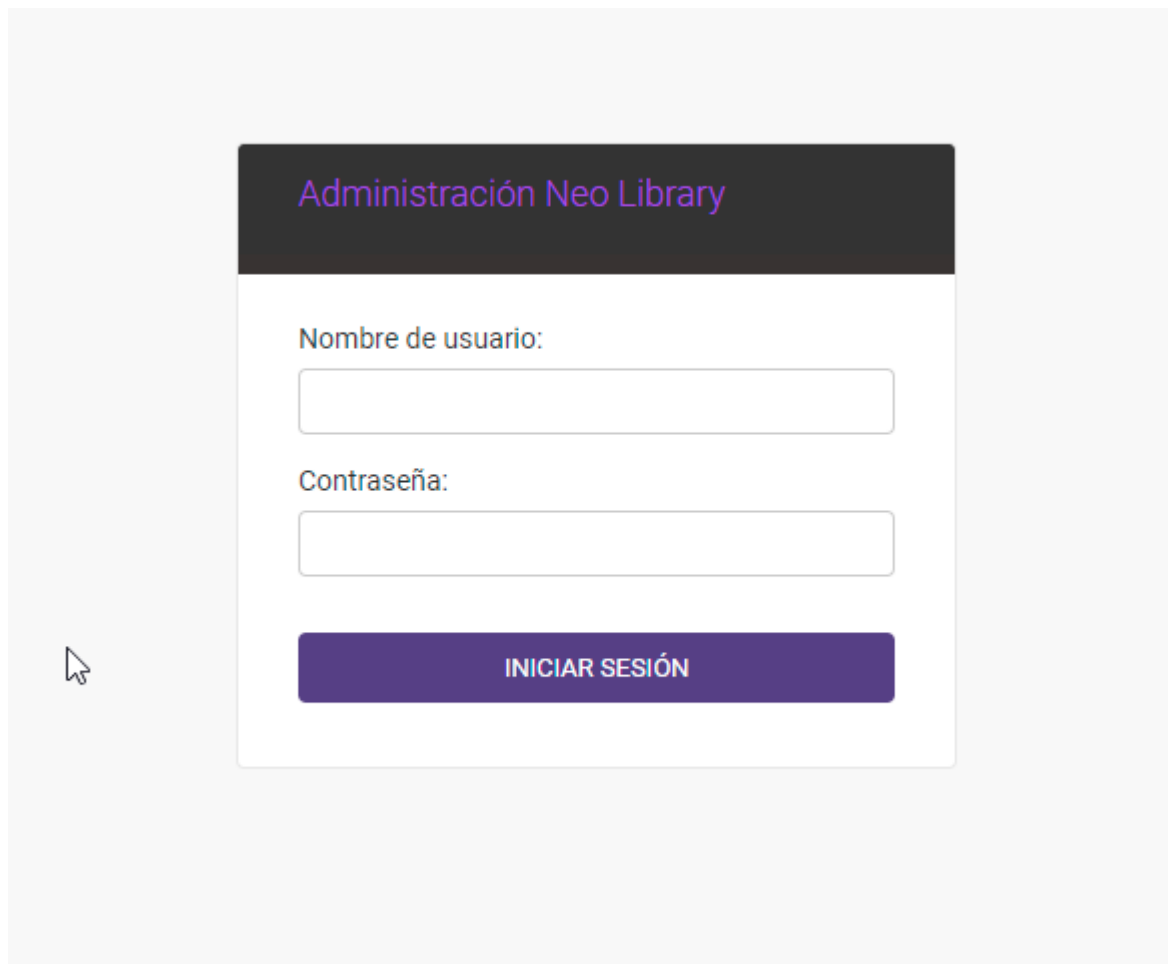
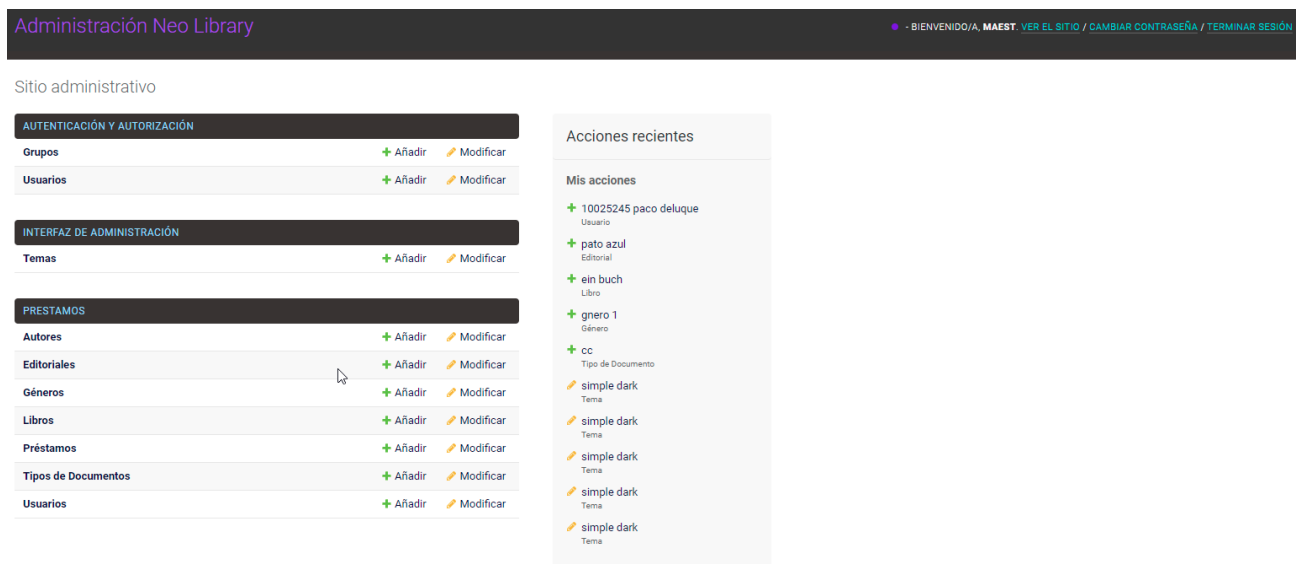


Imagen del inicio de sesión en el panel de control



Vista general del panel de administracion

## 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

Eliminar

Grabar y añadir otro

Grabar y continuar editando

GRABAR

Opciones del usuario(dentro del perfil de cada usuario)

«

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				<div></div>

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 ?

Q Filtro

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 ?

Q Filtro

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

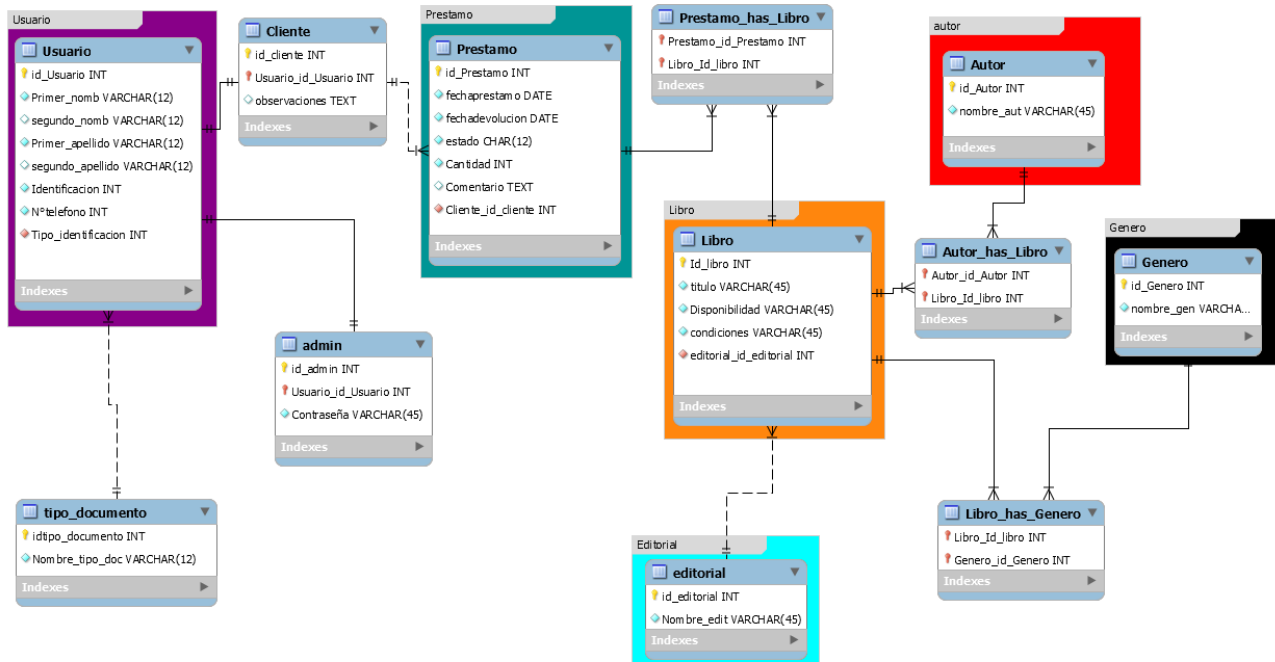
permisos de usuario elegidos ?

Vista de los paneles de permisos

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	ZF	AI	Default	Comment	
id_Autor	INT	✓	✓							Almacena el codigo que identifica al autor del labor	
nombre_aut	VARCHAR(45)		✓							almacena el nombre del autor del libro	

Genero											
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	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	ZF	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	ZF	AI	Default	Comment	
Libro_Id_libro	INT		✓	✓							
Genero_id_Genero	INT		✓	✓							

Prestamo											
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
id_Prestamo	INT	✓	✓							almacena elcodigo que identifica el prestamo	
fechaprestamo	DATE		✓							almacena la fecha en la que se realiza el prestamo	
fechadevolucion	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											
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
Prestamo_id_Prestamo	INT		✓	✓							
Libro_Id_libro	INT		✓	✓							

Usuario											
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	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	
Identificacion	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	ZF	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.

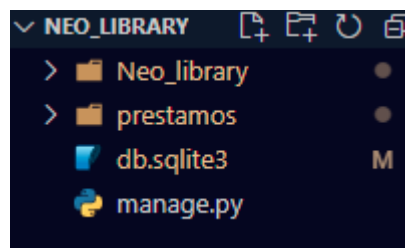
Fue escrito con la versión 3.9 32 bits de python y la versión 3.2 de django

# Documentación del código fuente

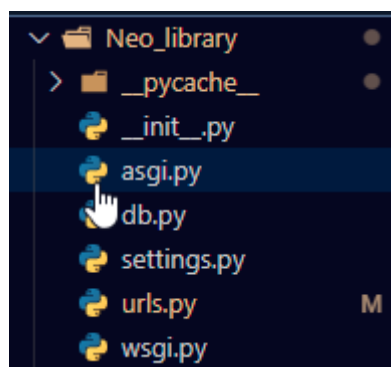
## Estructura de la aplicación

Una vez creado el proyecto se crea una carpeta que dentro contiene una carpeta con el mismo nombre y el archivo manage.py y asu 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
```



Carpeta principal del proyecto



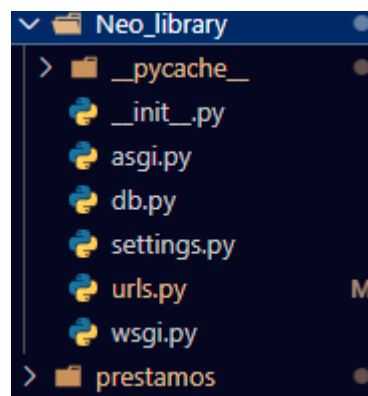
Archivos dentro de la carpeta del mismo nombre

Luego de creada una aplicación se crea una carpeta con la siguiente organización de archivos

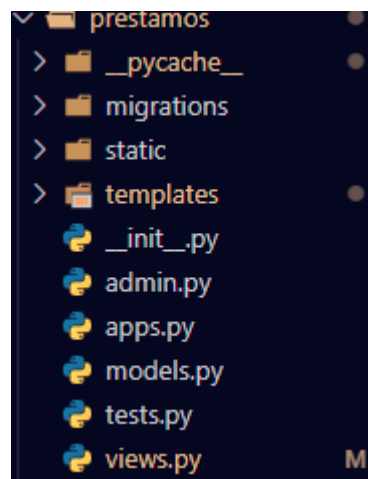
---



```
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
```

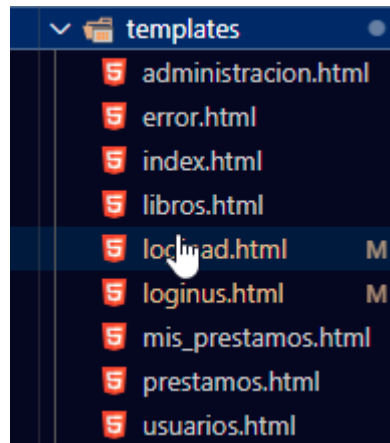


carpeta de neo library con la carpeta de prestamos



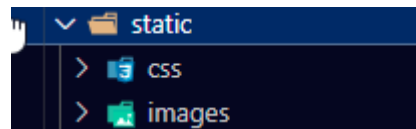
carpeta de prestamos(la aplicacion)

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



carpeta de templates

En la carpeta static están los archivos estáticos como las imagenes archivos css y json



Carpeta static

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 configuracion para distintas bases de datos

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

# Descripción de los acuerdos de niveles de servicios o ANS

## I. partes

A continuación se identifican las partes que suscriben el presente acuerdo: De una parte, CLIENTE, con C.I.F. \_CIF CLIENTE\_ y domicilio social en \_DOMICILIO SOCIAL CLIENTE\_, representada por D. \_NOMBRE DEL REPRESENTANTE CLIENTE\_ actuando en nombre y representación de esta entidad en virtud de su condición de \_CARGO REPRESENTANTE CLIENTE\_. De otra parte, PROVEEDOR como prestadora del servicio \_SERVICIO\_, con C.I.F. \_CIF PROVEEDOR\_ y domicilio social en \_DOMICILIO SOCIAL PROVEEDOR\_, representada por D. \_NOMBRE DEL REPRESENTANTE PROVEEDOR\_ actuando en nombre y representación de esta entidad en virtud de su condición de \_CARGO REPRESENTANTE PROVEEDOR\_.

## II. condiciones de servicio

El desollador de este sistema se compromete a cumplir con los requerimientos funcionales y no funcionales, es decir hacer que sea posible hacer un manejo adecuado de los libros de la biblioteca, de los usuarios de la biblioteca y llevar un buen control sobre los préstamos de la biblioteca el sistema se entregará a el administrador de la biblioteca o quien cumpla sus funciones

## III. rendimiento del sistema

El éxito del sistema se medirá en base a los requerimientos funcionales y/ no funcionales que se cumplan exitosamente

## IV. tiempo de respuesta

La prestación del servicio puede estar sujeta a incidentes que pueden comprometer el mantenimiento de unos niveles de servicio adecuados. En este sentido y para evitar que estos incidentes impacten en la menor medida posible en la prestación del servicio, se establecen unos criterios de priorización de incidentes que permitan ofrecer unos tiempos

de respuesta y resolución correctos. Estos criterios de priorización quedan recogidos en 2 tipos: Normal y críticos. Normales: Incidentes que no implican la detención total del servicio o que o comprometen la seguridad del mismo en cualquiera de sus parámetros. Críticos: Incidentes que implican la detención total del servicio o que pueden comprometer la seguridad del mismo

Por lo que respecta, a los tiempos de respuesta y resolución se han establecido los siguientes niveles de servicio:

Tarea	T.respuesta	T.solucion
Peticiones	24h	48h
Incidencia normal	12h	16h
Incidencia crítica	4h	6h

**V.finalización**

El acuerdo de nivel de servicio tendrá validez durante todo el periodo de tiempo que dure la prestación del servicio.