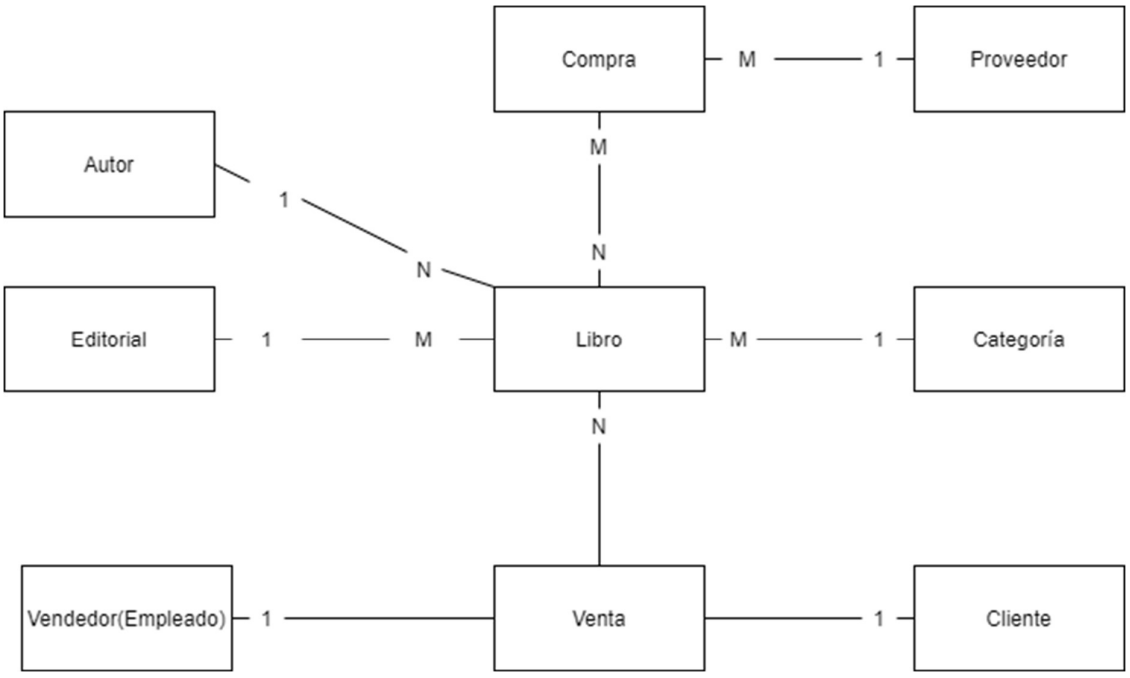
Nombre del Proyecto: Librería F&F

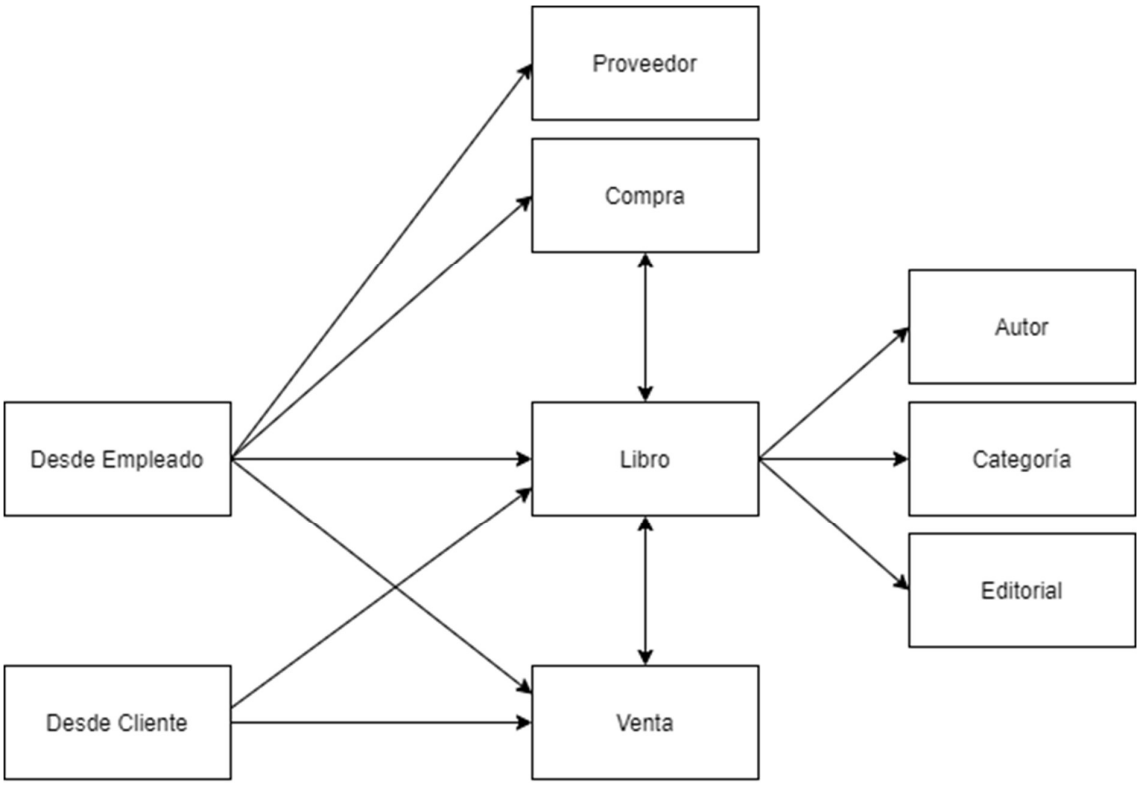
Descripción del Proyecto: nuestro proyecto constará de una aplicación web desarrollada en el framework Flask de Python el cual representará un sistema real que podría ser utilizado en una Librería en una empresa real.
Nuestro sistema tendrá un alcance multiuso dentro de la organización, tal como la gestión de venta y adquisición de los libros, registro de usuarios y empleados, cada uno con sus funcionalidades habilitadas dentro del sistema. También la búsqueda interna de los productos que están o no en stock para que el Cliente esté informado de los artículos que la empresa posee.

Modelo de Dominio:



Bosquejo de Arquitectura:

Diagramamos el bosquejo teniendo en cuenta las dos perspectivas que va a tener nuestro sistema, la del Cliente y la de los Empleados, ya que la funcionalidad de las opciones que van a tener cada uno van a ser completamente diferentes, y a los datos que van a tener acceso cada uno también lo es.



Requerimientos:

Funcionales: (el sistema debe:)

- Tener registro de todos los libros que están o no en stock y la cantidad disponible.
- Poder crear usuarios tanto para los Clientes como para los Empleados de la organización.
- Gestionar las ventas de libros a los usuarios, actualizando el Stock, calculando los costos e informándolos al Cliente para el futuro pago (fuera del alcance).
- Tener un registro apropiado de los distintos Proveedores junto con sus datos.
- Gestionar la compra de libros a los Proveedores, actualizando el Stock.
- Permitir a los usuarios consultas de libros, junto con sus precios, descripción, stock, etc.

No funcionales

Portabilidad:

- El sistema debe funcionar correctamente en múltiples navegadores (Sólo Web).

- El sistema debe ejecutarse desde un único archivo .py llamado app.py (Sólo Escritorio).

Seguridad:

- Todas las contraseñas deben guardarse con encriptado criptográfico (SHA o equivalente).
- Todas los Tokens / API Keys o similares no deben exponerse de manera pública.
- Todos los usuarios deben poder registrarse al sistema de forma segura y sus datos deben tener un grado de confidencialidad acorde a los requisitos solicitados en la actualidad.
- Poder diferenciar a qué puede acceder cada usuario, ya sea por parte de los Clientes o los Empleados.

Mantenibilidad:

- El sistema debe diseñarse con la arquitectura en 3 capas.
- El sistema debe utilizar control de versiones mediante GIT.
- El sistema debe estar programado en Python 3.8 o superior.
- El sistema debe estar desarrollado en el framework Flask.

Confiabilidad:

Escalabilidad:

- El sistema debe funcionar desde una ventana normal y una de incógnito de manera independiente (Sólo Web). Aclaración: No se debe guardar el usuario en una variable local, deben usarse Tokens, Cookies o similares.

Rendimiento:

- El sistema debe funcionar en un equipo hogareño estándar.

Reusabilidad:

Flexibilidad:

- El sistema debe utilizar una base de datos SQL.

Stack tecnológico

Capa de datos:

- Base de datos Sql: Lo utilizaremos para gestionar la base de datos y conectarla al proyecto, aún no tenemos definido cuál hosting utilizaremos.

Capa de negocios:

Capa de presentación: