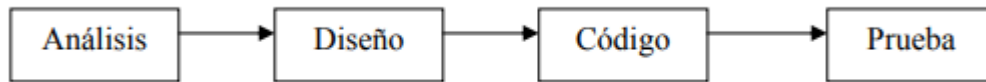


Modelo lineal secuencial.



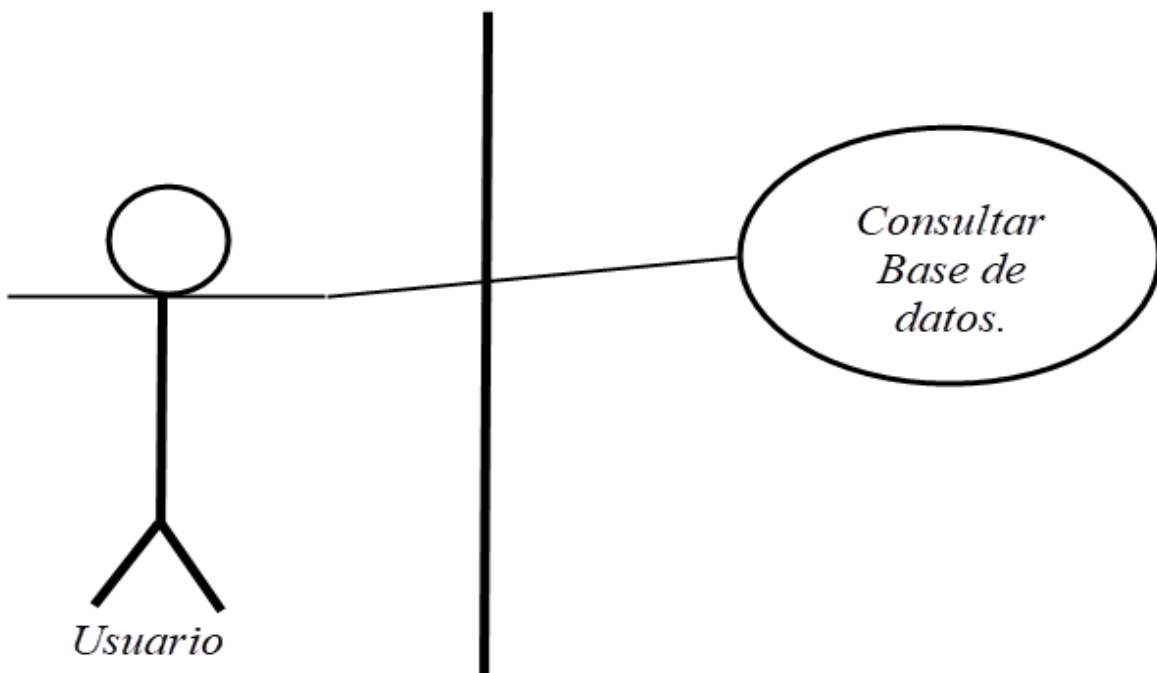
ANALISIS

Diseñar y desarrollar una pequeña aplicación en Flask para ser desplegada en un servidor web, la aplicación debe cumplir con:

1. Llenar una base de datos SQLite con la información de la lista de usuarios (firstname, lastname, username, mail, password) archivo anexo.
2. Desplegar en HTML (Vista) la información en pantalla cada vez que se solicite la URL /users/list
3. Retornar el listado en Json cada que se solicite la URL /api/v1/users/
4. Desplegar la aplicación usando Docker

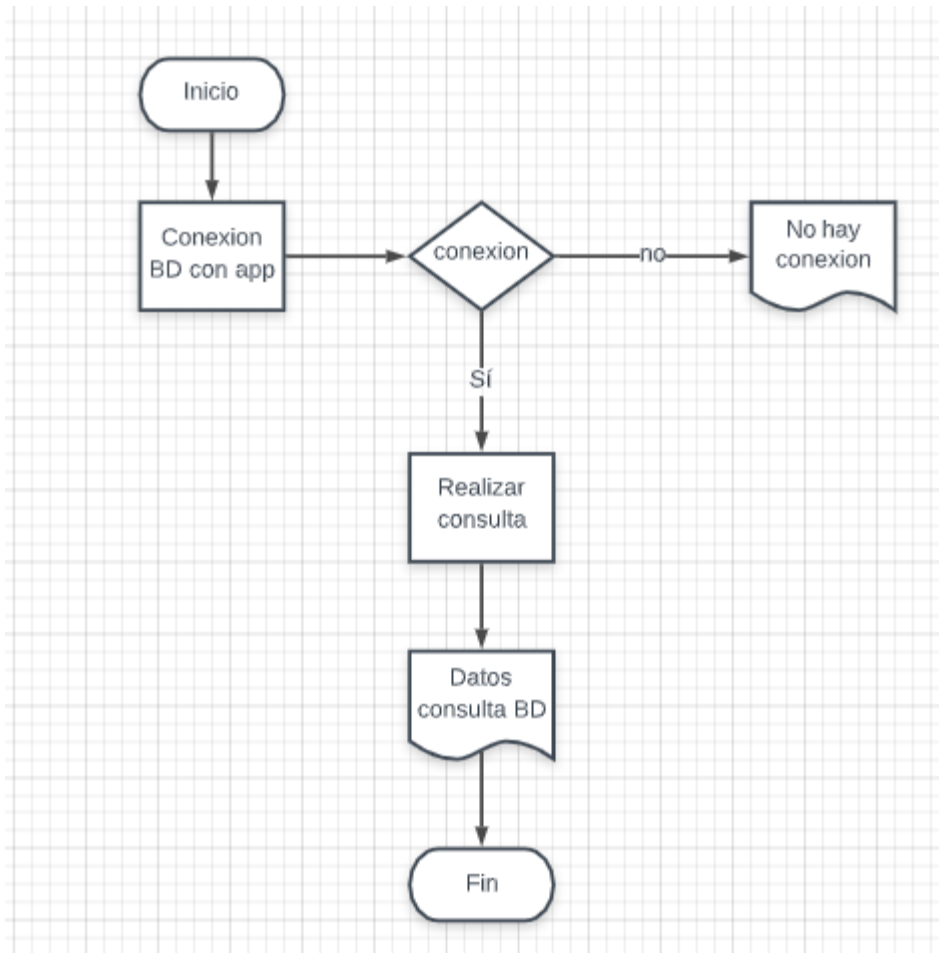
Compartir el enlace al repositorio Github del código fuente y un PDF con el diseño y comentarios sobre el diseño e implementación.

Diagrama de casos de uso



En este diagrama se muestra la acción que realiza la aplicación de realizar la consulta a nuestra base de datos para visualizar la información.

Diagrama de Flujo.



DISEÑO.

- Utilización de Github para guardar cambios y seguimiento de avance.
- Migración de CSV a base de datos Sqlite. (DB Browser for SQLite)
- CSV con información de usuarios.
- Algoritmo para conexión de base de datos.
- Algoritmo para mostrar información de la base de datos.
- Algoritmo para retorno de solicitudes con Json al usar la URL `/api/v1/users/`
- HTML para generar vista (index.html).
- Activación de servicios para visualización de información.

- Instalación y utilización de micro framework Flask.
- Creación de virtual environments para desarrollo de la aplicación
- Instalación de SQL Alchemy y otros complementos.
- Implementación y utilización de Docker.

CODIGO.

Link para repositorios y código desarrollado para la realización de aplicación solicitada con comentarios de funcionalidades.

<https://github.com/hacm97/ApiFullstack>

Modelo de datos.

users		
firstname	TEXT	"firstname" TEXT
lastname	TEXT	"lastname" TEXT
username	TEXT	"username" TEXT
mail	TEXT	"mail" TEXT
password	TEXT	"password" TEXT

En este modelo de datos, se plantea una base de datos sencilla, la cual posea 1 tabla que almacena los datos de los usuarios tales como su firstname, lastname, password, mail y username.

RF 1: Consulta

Descripción: Propiedad de la aplicación para realizar consulta de información en la base de datos.

Entrada: Usuario, Base de datos.

Prioridad: Alta

Salida: información registrada en la base de datos.

Acción: conectividad con la base de datos y realización de la consulta.

Requerimiento: Realizar consulta.

Pre-condición: Acceso a la red (Prototipo será local), base de datos.

RF 2: Mostrar consulta HTML.

Descripción: despliegue de vista con información registrada anteriormente en nuestra base de datos en vista HTML.

Entrada: Base de datos.

Prioridad: Alta

Salida: Tabla con información contenida en la base de datos.

Acción: acción correcta desplegar tabla con información de usuarios.

Requerimiento: ejecutar consulta.

Pre-condición: Acceso a la red (Prototipo será local), base de datos.

RF 3: Mostrar consulta Json.

Descripción: despliegue de vista con información registrada anteriormente en nuestra base de datos en vista Json.

Entrada: Base de datos.

Prioridad: Alta

Salida: Tabla con información contenida en la base de datos.

Acción: acción correcta desplegar tabla con información de usuarios.

Requerimiento: ejecutar consulta.

Pre-condición: Acceso a la red (Prototipo será local), base de datos.