MANUAL TÉCNICO

[PROYECTO 2]

DAVED ABSHALON EJCALON CHONAY - 202105668

INGENIERÍA EN CIENCIAS Y SISTEMAS

LABORATORIO INTRODUCCIÓN A LA PROGRAMACIÓN 1

REQUISITOS

- Sistema operativo
  Windows 7

- Procesador
  Intel Core 2 Duo E8200, AMD Phenom II X6 1075 a 3,0 GHz

- Memoria
  1 GB de RAM

- Almacenamiento
  1 GB de espacio disponible

- Tarjeta gráfica
  NVIDIA GT 630 / AMD HD 7750

- Internet
  Acceso a internet, con cualquier explorador web

- Visual Studio
  Con python instalado

DESARROLLO DEL PROGRAMA

- ## Backend

Se refiere a la parte de una aplicación web o sitio web que está encargada de procesar y gestionar la lógica del servidor. Es la capa invisible que se ocupa de manejar la lógica de negocio, la gestión de datos, la seguridad y la comunicación con la base de datos.

- ## Administrador.py

  - **Atributos**:

    - nombre (tipo: cadena de caracteres): Almacena el nombre del administrador.
    - apellido (tipo: cadena de caracteres): Almacena el apellido del administrador.
    - nombreusuario (tipo: cadena de caracteres): Almacena el nombre de usuario del administrador.
    - password (tipo: cadena de caracteres): Almacena la contraseña del administrador.

  - **Resumen**:

    - Esta clase "Administrador" proporciona una estructura básica para representar y manipular los datos de un administrador. Los métodos getter y setter permiten acceder y modificar los atributos de un objeto Administrador de manera controlada.

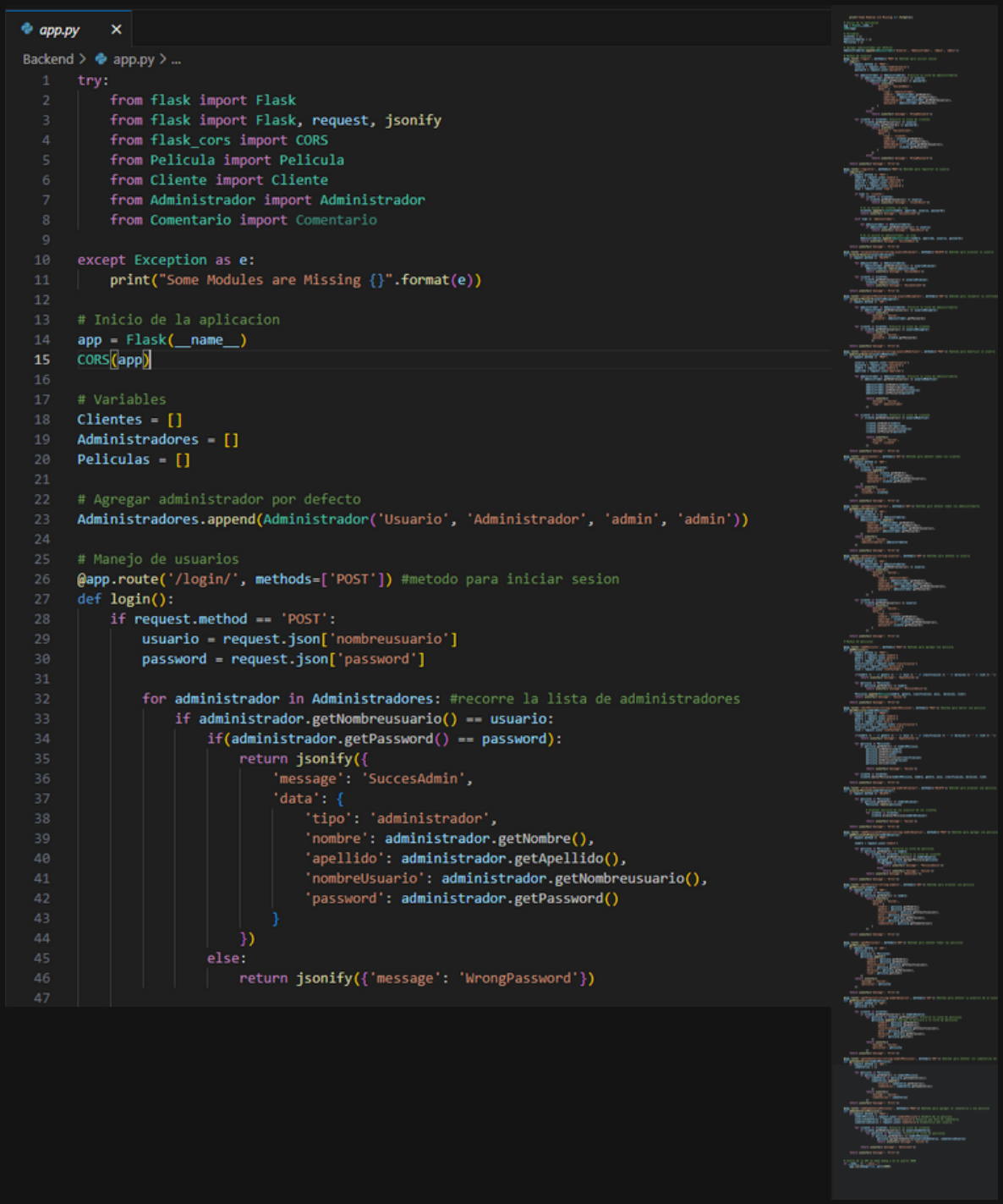
```python
class Administrador:

    def __init__(self, nombre, apellido, nombreusuario, password)
        self.nombre = nombre
        self.apellido = apellido
        self.nombreusuario = nombreusuario
        self.password = password

    #getters
    def getNombre(self):
        return self.nombre

    def getApellido(self):
        return self.apellido

    def getNombreusuario(self):
        return self.nombreusuario

    def getPassword(self):
        return self.password
```

DESARROLLO APP.PY

- app.py
  - **Resumen**:
    - El código implementa una API básica utilizando Flask para administrar usuarios, películas y comentarios. Sin conocer las definiciones de las clases "Pelicula", "Cliente", "Administrador" y "Comentario", no es posible proporcionar más detalles sobre su funcionalidad interna.
  - **Rutas de la aplicación para el manejo de usuarios**:
    - /login/: Verificación de inicio de sesión para administradores y clientes.
    - /registro/: Registro de nuevos usuarios (administradores o clientes).
    - /eliminarUsuario/<string:usuarioEliminar>: Eliminación de un usuario según su nombre de usuario.
    - /recuperarPassword/<string:usuarioRecuperar>: Recuperación de la contraseña de un usuario.
    - /modificarUsuario/<string:usuarioModificar>: Modificación de los datos de un usuario.
    - /getClientes/: Obtención de todos los clientes registrados.
    - /getAdministradores/: Obtención de todos los administradores registrados.
    - /getUsuario/<string:usuario>: Obtención de los datos de un usuario específico.
  - **Rutas de la aplicación para el manejo de películas**:
    - /addPelicula/: Agregado de una nueva película.
    - /editPelicula/<string:nombrePelicula>: Edición de los datos de una película existente.
    - /eliminarPelicula/<string:nombreEliminar>: Eliminación de una película según su nombre.
    - /addPeliculaToPlaylist/<string:nombreUsuario>: Agregado de una película a la lista de reproducción de un cliente.
    - /getPelicula/<string:nombre>: Obtención de los datos de una película específica.
    - /getPeliculas/: Obtención de todas las películas registradas.
    - /getPlaylist/<string:nombreUsuario>: Obtención de la lista de reproducción de un cliente.
    - /getComentarios/<string:nombrePelicula>: Obtención de los comentarios de una película.
    - /addComentarioPelicula/: Agregado de un comentario a una película.

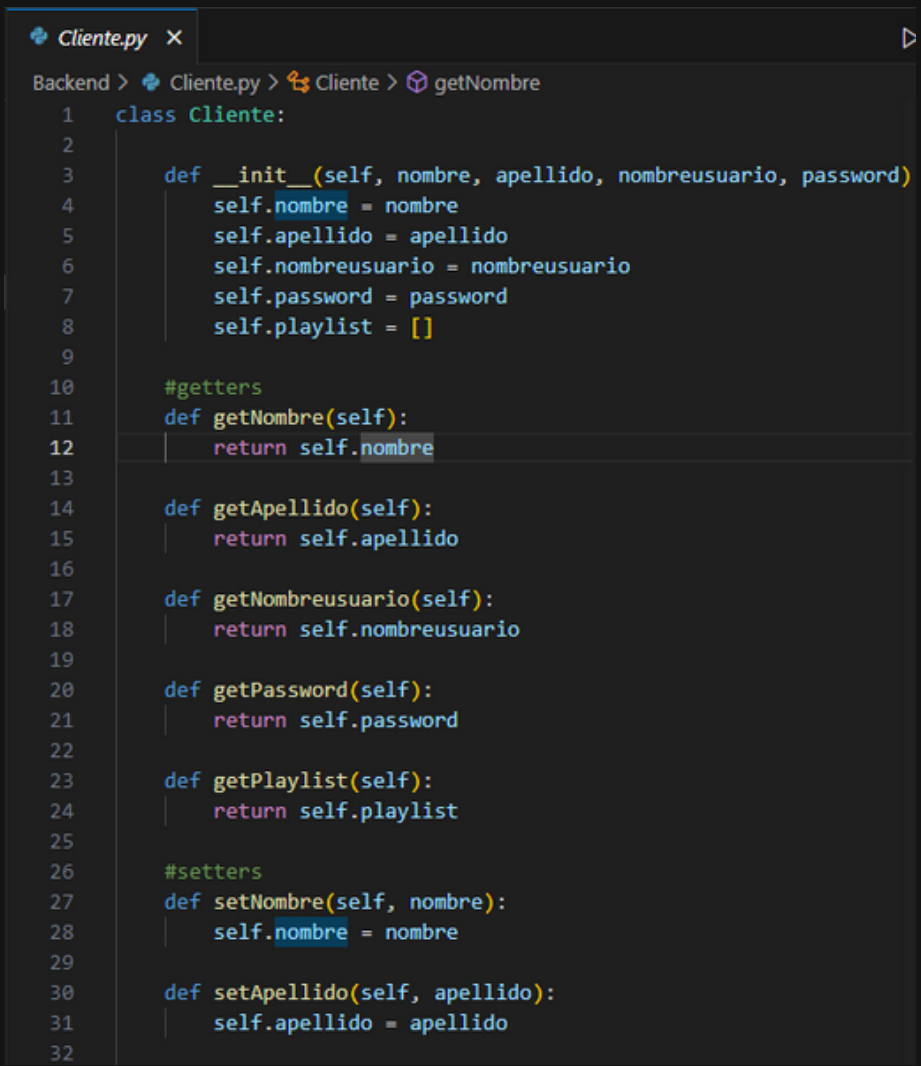
DESARROLLO APP.PY

- app.py



DESARROLLO CLIENTE.PY

- Cliente.py
  - **Atributos**:
    - nombre: Una cadena que almacena el nombre del cliente.
    - apellido: Una cadena que almacena el apellido del cliente.
    - nombreusuario: Una cadena que almacena el nombre de usuario del cliente.
    - password: Una cadena que almacena la contraseña del cliente.
    - playlist: Una lista que almacena las películas de la lista de reproducción del cliente.
  - **Resumen**:
    - El código define una clase llamada "Cliente" que representa a un cliente o usuario en una aplicación de lista de reproducción de películas. En resumen, esta clase proporciona métodos para agregar, eliminar y editar películas en la lista de reproducción de un cliente, así como para acceder y actualizar la información del cliente.



DESARROLLO COMENTARIO.PY

- Comentario.py
  - **Atributos**:
    - usuario: Almacena el nombre o identificador del usuario que hizo el comentario.
    - comentario: Almacena el texto del comentario en sí.
  - **Métodos**:
    - __init__ es el constructor de la clase y se llama cuando se crea un nuevo objeto Comentario. Toma los argumentos usuario y comentario y los asigna a los atributos correspondientes del objeto.
    - getUsuario y getComentario son getters que devuelven los valores de los atributos usuario y comentario, respectivamente.
    - setUsuario y setComentario son setters que actualizan los valores de los atributos usuario y comentario, respectivamente.
  - **Resumen**:
    - Estos métodos permiten acceder y modificar el contenido de un comentario, así como obtener información sobre el usuario que realizó el comentario. En resumen, esta clase proporciona una forma de representar y manipular comentarios de usuarios en un contexto determinado.

Comentario.py

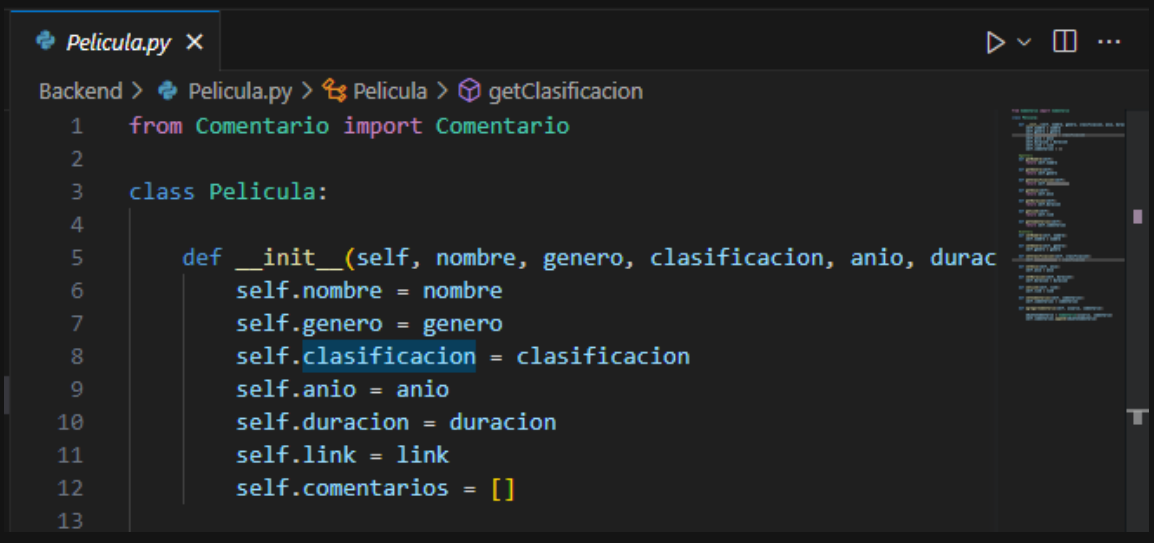
Backend > Comentario.py > ...

```python
class Comentario:

    def __init__(self, usuario, comentario):
        self.usuario = usuario
        self.comentario = comentario

    #getters
    def getUsuario(self):
        return self.usuario

    def getComentario(self):
        return self.comentario

    #setters
    def setUsuario(self, usuario):
        self.usuario = usuario

    def setComentario(self, comentario):
        self.comentario = comentario

```

DESARROLLO PELICULA.PY

- Pelicula.py
  - **Métodos**:
    - __init__ es el constructor de la clase y se llama cuando se crea un nuevo objeto Pelicula. Toma los argumentos correspondientes a los atributos de la película y los asigna a los atributos del objeto.
    - getNombre, getGenero, getClasificacion, getAnio, getDuracion y getLink son getters que devuelven los valores de los atributos correspondientes de la película.
    - setNombre, setGenero, setClasificacion, setAnio, setDuracion y setLink son setters que actualizan los valores de los atributos correspondientes de la película.
    - getComentarios es un getter que devuelve la lista de comentarios asociados a la película.
    - setComentarios es un setter que actualiza la lista de comentarios asociados a la película.
    - agregarComentario permite agregar un nuevo comentario a la lista de comentarios de la película. Recibe dos argumentos: usuario y comentario, que se utilizan para crear un nuevo objeto de la clase "Comentario" y luego se agrega a la lista de comentarios de la película.

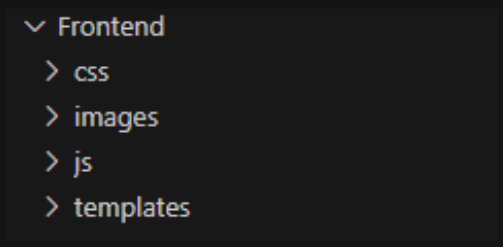
  - **Resumen**:
    - En resumen, esta clase proporciona una forma de representar y manipular películas, incluyendo información sobre la película y la capacidad de agregar comentarios a ellas.



DESARROLLO DEL PROGRAMA

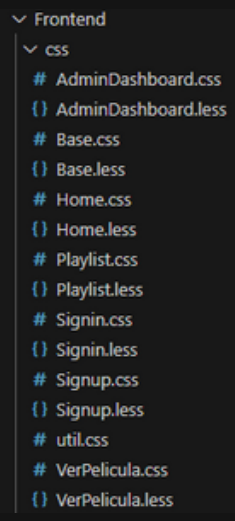
- ## Frontend

El término "Frontend" se refiere a la parte de una página web o aplicación que se muestra y con la que interactúan los usuarios. Es la interfaz visible y accesible al público. El desarrollo del Frontend implica la creación de la estructura, diseño y funcionalidad de la interfaz de usuario.

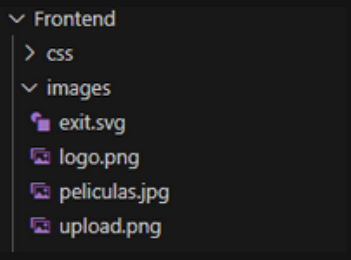


- ## Desarrollo general

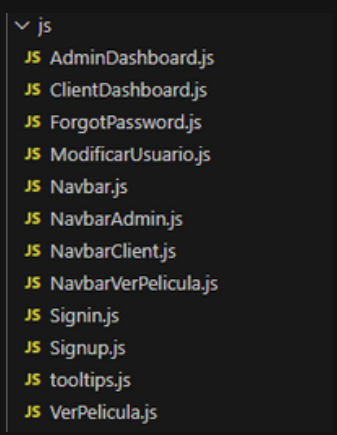
  - **CSS**:
    - Es el lenguaje utilizado para dar estilo y presentación al contenido HTML. CSS define la apariencia visual de los elementos HTML, como colores, fuentes, tamaños, márgenes, diseños, efectos visuales, entre otros. Permite controlar la disposición y el aspecto estético de los elementos en la página.



  - **Images**:
    - El manejo de imágenes en el desarrollo Frontend es esencial para enriquecer visualmente las páginas web y proporcionar una experiencia atractiva para los usuarios. El apartado de imágenes en Frontend implica la inclusión, presentación y optimización de imágenes dentro de una página web, en este caso se utilizaron estas 4.



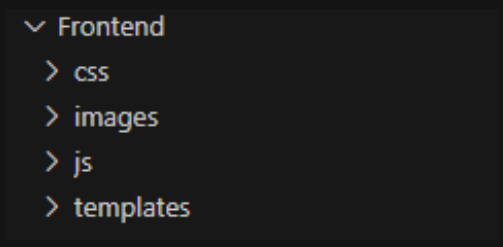
  - **JS**:
    - Se utiliza para agregar interactividad y funcionalidad dinámica a una página web. Con JavaScript, se pueden realizar acciones y manipular elementos HTML en respuesta a eventos del usuario, como hacer clic en un botón, enviar formularios, mostrar u ocultar contenido, realizar validaciones, entre otros.



DESARROLLO DEL PROGRAMA

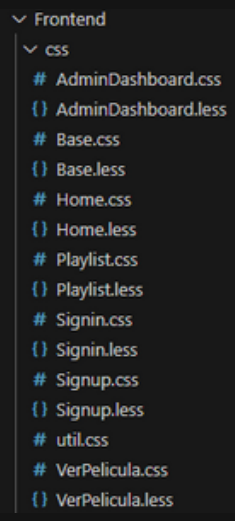
- ## Frontend

El término "Frontend" se refiere a la parte de una página web o aplicación que se muestra y con la que interactúan los usuarios. Es la interfaz visible y accesible al público. El desarrollo del Frontend implica la creación de la estructura, diseño y funcionalidad de la interfaz de usuario.

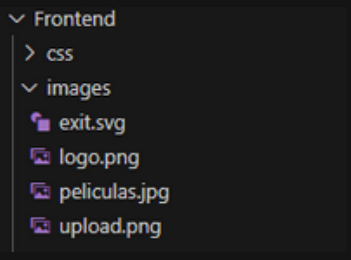


- ## Desarrollo general

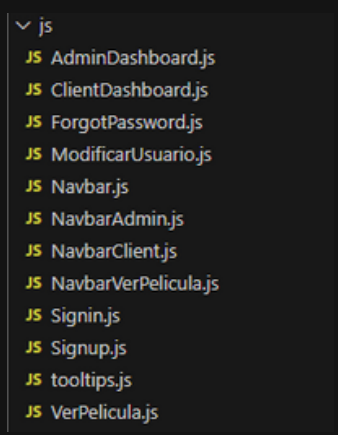
  - **CSS**:
    - Es el lenguaje utilizado para dar estilo y presentación al contenido HTML. CSS define la apariencia visual de los elementos HTML, como colores, fuentes, tamaños, márgenes, diseños, efectos visuales, entre otros. Permite controlar la disposición y el aspecto estético de los elementos en la página.



  - **Images**:
    - El manejo de imágenes en el desarrollo Frontend es esencial para enriquecer visualmente las páginas web y proporcionar una experiencia atractiva para los usuarios. El apartado de imágenes en Frontend implica la inclusión, presentación y optimización de imágenes dentro de una página web, en este caso se utilizaron estas 4.



  - **JS**:
    - Se utiliza para agregar interactividad y funcionalidad dinámica a una página web. Con JavaScript, se pueden realizar acciones y manipular elementos HTML en respuesta a eventos del usuario, como hacer clic en un botón, enviar formularios, mostrar u ocultar contenido, realizar validaciones, entre otros.



DESARROLLO DEL HTML

- HTML

(HyperText Markup Language): Es el lenguaje de marcado utilizado para estructurar y organizar el contenido de la página web. HTML define los elementos y etiquetas que representan los diferentes componentes, como encabezados, párrafos, imágenes, formularios, enlaces, etc.