Peliculón GT

Manual Técnico

Requerimientos del Sistema:

- Procesador **a** 1.6 GHz o superior.
- 1 GB (32 bits) o 2 GB (64 bits) **de** RAM (agregue 512 MB al host si se ejecuta **en** una máquina virtual)
- 3 GB **de** espacio disponible **en** el disco duro.
- Tarjeta **de** vídeo compatible con DirectX 9 con resolución **de** pantalla **de** 1024 x 768 o más.

Descripción de la aplicación:

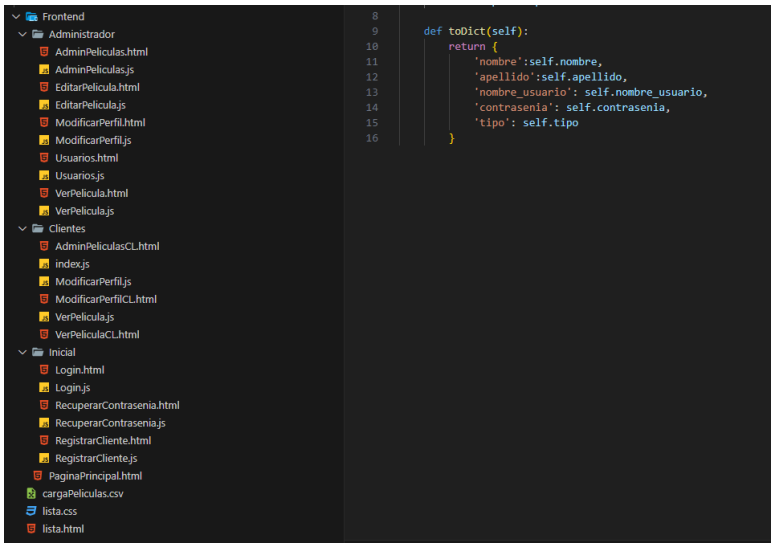
Aplicación desarrollada en el entorno de Visual Studio Code, haciendo uso de los lenguajes Java, Javascript y las herramientas de HTTP, presenta redirecciones entre páginas, imágenes, videos web y recibe datos de usuario.

Diccionario de clases:

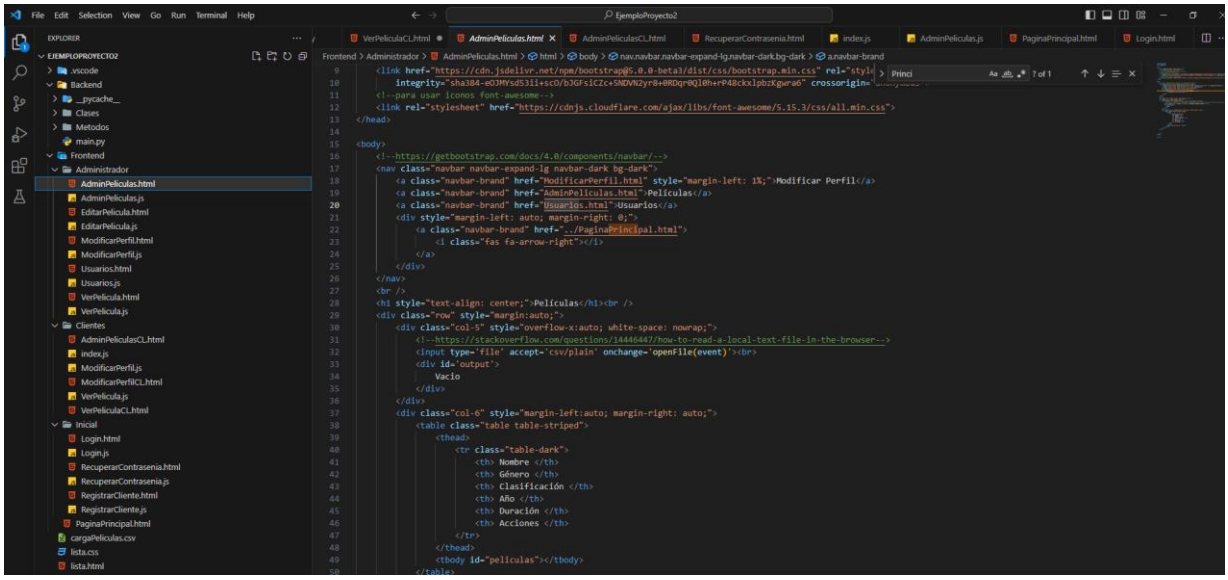
```python
class Pelicula:
    def __init__(self, nombre, genero, clasificacion, anio, duracion, link):
        self.nombre = nombre
        self.genero = genero
        self.clasificacion = clasificacion
        self.anio = anio
        self.duracion = duracion
        self.link = link

    def toDict(self):
        return {
            'nombre':self.nombre,
            'genero':self.genero,
            'clasificacion': self.clasificacion,
            'anio': self.anio,
            'duracion': self.duracion,
            'link': self.link
        }
```

Estructura de las clases para almacenamiento y reconocimiento de los datos de usuario en el backend



Fronted dividido en Administrador, clientes e inicial



Cada archivo html va conectado a un programa javascript.

Diccionario de clases:

- Comentario.__init__(self, pelicula, autor, mensaje): El método constructor de la clase Comentario. Inicializa los atributos pelicula, autor y mensaje de la instancia.
- Comentario.toDict(self): Devuelve un diccionario que contiene los atributos de la instancia de Comentario.
- Pelicula.__init__(self, nombre, genero, clasificacion, anio, duracion, link): El método constructor de la clase Pelicula. Inicializa los atributos nombre, genero, clasificacion, anio, duracion y link de la instancia.
- Pelicula.toDict(self): Devuelve un diccionario que contiene los atributos de la instancia de Pelicula.
- Usuario.__init__(self, nombre, apellido, nombre_usuario, contrasenia, tipo): El método constructor de la clase Usuario. Inicializa los atributos nombre, apellido, nombre_usuario, contrasenia y tipo de la instancia.
- Usuario.toDict(self): Devuelve un diccionario que contiene los atributos de la instancia de Usuario.
- CargarPeliculas(datos): Toma un diccionario datos que contiene un texto con información de películas, y devuelve una lista de diccionarios que representan cada película.
- GetPelicula(datos, peliculas): Toma un diccionario datos que contiene el nombre de una película, y una lista de diccionarios peliculas que representan las películas cargadas. Devuelve un diccionario que contiene la información de la película correspondiente al nombre proporcionado y un código de estado HTTP (200 si se encuentra la película, 400 en caso contrario).
- EditarPelicula(datos, peliculas): Toma un diccionario datos que contiene información de una película a editar, y una lista de diccionarios peliculas que representan las películas cargadas. Devuelve un diccionario que contiene la lista actualizada de películas y un código de estado HTTP (200 si se edita la película, 400 en caso contrario).
- EliminarPelicula(datos, peliculas): Toma un diccionario datos que contiene el nombre de una película, y una lista de diccionarios peliculas que representan las películas cargadas. Devuelve un diccionario que contiene la lista actualizada de películas y un código de estado HTTP (200 si se elimina la película, 400 en caso contrario).
- GetComentarios(datos, comentarios): Toma un diccionario datos que contiene el nombre de una película, y una lista de diccionarios comentarios que representan los comentarios cargados. Devuelve un diccionario que contiene la lista de comentarios correspondientes a la película proporcionada y un código de estado HTTP (200 si se encuentran comentarios, 400 en caso contrario).
- EliminarUsuario(datos, usuarios): Toma un diccionario datos que contiene el nombre de un usuario, y una lista de diccionarios usuarios que representan los usuarios registrados. Devuelve un diccionario que contiene la lista actualizada de usuarios y un código de estado HTTP (200 si se elimina el usuario, 400 en caso contrario).
- RegistrarUsuario(datos, usuarios): Toma un diccionario datos que contiene información de un nuevo usuario a registrar, y una lista de diccionarios usuarios que representan los usuarios registrados. Devuelve un diccionario que indica si se ha registrado el usuario correctamente o no, y un código de estado HTTP (200 si se registra el usuario, 400 en caso contrario).
- RecuperarContrasenia(datos, usuarios): Toma un diccionario datos que contiene el nombre de un usuario, y una lista de diccionarios usuarios que representan los usuarios registrados. Devuelve un diccionario que contiene la contraseña del usuario correspondiente al nombre proporcionado y un código de estado HTTP (200 si se encuentra la contraseña, 400 en caso contrario).
- IniciarSesion(datos, usuarios): Toma un diccionario datos que contiene información de un usuario que quiere iniciar sesión, y una lista de diccionarios usuarios que representan los usuarios registrados. Devuelve un diccionario que contiene información del usuario que ha iniciado sesión, si existe, y un código de estado HTTP (200 si se inicia sesión, 400 en caso contrario)

Diccionario de funciones:

```python
def rutaInicial():
    return("Si funciona")

# ------------------------------------------------------------ INICIAL
@app.route('/registrarUsuario', methods=['POST'])
def registrarUsuario():
    respuesta = RegistrarUsuario(request.json, usuarios)
    return respuesta

@app.route('/recuperarContrasenia', methods=['POST'])
def recuperarContrasenia():
    respuesta = RecuperarContrasenia(request.json, usuarios)
    return respuesta

@app.route('/iniciarSesion', methods=['POST'])
def iniciarSesion():
    respuesta = IniciarSesion(request.json, usuarios)
    global usuarioEnSesion

    usuarioEnSesion = respuesta['usuarioEnSesion']
    print(usuarioEnSesion)
    return respuesta
```