

Documentación

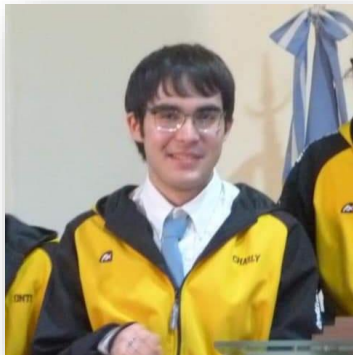
Sprint 8

Back-end

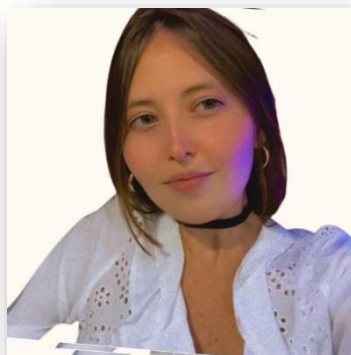
Full Stack Developer

Instituto Tecnológico de Buenos Aires

DuckBank



Carlos Castro



Celina Bono



Ignacio Iannino

views.py

Este archivo contiene las vistas y lógica para manejar las solicitudes relacionadas con la creación y gestión de usuarios, préstamos, y validación de autenticación en la aplicación.

1. UserViewSet

Tipo: viewsets.ModelViewSet

- **Descripción:**

Esta vista se encarga de gestionar las operaciones CRUD para el modelo CustomUser. Permite realizar operaciones de lectura, actualización y eliminación de los usuarios en el sistema.

- **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.

- **Métodos disponibles:**

- GET: Obtiene la lista de usuarios registrados.
 - POST: Crea un nuevo usuario.
 - PUT: Actualiza los datos de un usuario existente.
 - DELETE: Elimina un usuario.
-

2. LoanViewSet

Tipo: viewsets.ModelViewSet

- **Descripción:**

Esta vista gestiona las operaciones sobre los préstamos. Permite la creación, consulta, actualización y eliminación de préstamos.

- **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.

- **Métodos disponibles:**

- GET: Obtiene la lista de préstamos.
- POST: Solicita un nuevo préstamo.
- PUT: Actualiza un préstamo.
- DELETE: Elimina un préstamo.

3. LoginView

Tipo: APIView

- **Descripción:**

Permite que los usuarios inicien sesión proporcionando su nombre de usuario y contraseña. Si las credenciales son correctas, se genera un token JWT que el frontend puede usar para autenticar al usuario en futuras peticiones.

- **Método:**

- **POST:** Recibe el nombre de usuario y la contraseña, valida las credenciales y, si son correctas, devuelve el token de acceso y el token de actualización.

- **Respuesta:**

- **200 OK:** Si las credenciales son correctas, se devuelve el token de acceso, el token de actualización y la información del usuario.
 - **401 Unauthorized:** Si las credenciales son inválidas.
-

4. RegisterView

Tipo: APIView

- **Descripción:**

Permite registrar un nuevo usuario en el sistema. Verifica que el nombre de usuario no esté en uso y crea un nuevo usuario junto con su perfil asociado.

- **Método:**

- **POST:** Recibe los datos de usuario (nombre, contraseña y DNI), valida los datos y crea el usuario en la base de datos.

- **Respuesta:**

- **201 Created:** Si el usuario se crea correctamente.
 - **400 Bad Request:** Si el nombre de usuario ya está en uso o si falta algún dato obligatorio.
-

5. ValidateTokenView

Tipo: APIView

- **Descripción:**

Permite validar un token JWT. El token se recibe en los encabezados de la solicitud y, si es válido, se responde con un mensaje de éxito.

- **Método:**

- GET: Recibe el token en los encabezados, lo valida y devuelve una respuesta indicando si el token es válido o no.

- **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.

- **Respuesta:**

- **200 OK:** Si el token es válido.
 - **401 Unauthorized:** Si el token es inválido.
-

6. SolicitarPrestamoView

Tipo: APIView

- **Descripción:**

Permite a un usuario solicitar un préstamo, proporcionando el monto, el motivo y un comentario opcional. Valida los datos ingresados y crea un nuevo préstamo aprobado con la fecha de solicitud.

- **Método:**

- POST: Recibe los datos necesarios para solicitar un préstamo y los valida. Si todo es correcto, crea el préstamo.

- **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.

- **Respuesta:**

- **201 Created:** Si la solicitud de préstamo es exitosa.
 - **400 Bad Request:** Si falta algún dato obligatorio o el monto excede el límite permitido.
-

7. ObtenerPrestamosView

Tipo: APIView

- **Descripción:**

Permite a los usuarios obtener una lista de todos los préstamos que han solicitado previamente.

- **Método:**
 - GET: Devuelve todos los préstamos asociados al usuario autenticado.
 - **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Devuelve la lista de préstamos del usuario.
-

8. mis_prestamos

Tipo: @api_view(['GET'])

- **Descripción:**

Función que permite obtener los préstamos solicitados por el usuario, similar a la vista ObtenerPrestamosView.
 - **Método:**
 - GET: Devuelve una lista de préstamos con información básica como monto, motivo, estado y fecha de solicitud.
 - **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Devuelve los datos de los préstamos solicitados.
-

9. EliminarPrestamoView

Tipo: APIView

- **Descripción:**

Permite a un usuario eliminar un préstamo previamente solicitado, si tiene los permisos necesarios.
- **Método:**
 - DELETE: Elimina un préstamo especificado por su ID, si es válido y pertenece al usuario autenticado.
- **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.
- **Respuesta:**
 - **204 No Content:** Si el préstamo es eliminado exitosamente.

- **404 Not Found:** Si el préstamo no se encuentra o el usuario no tiene permiso para eliminarlo.

10. ListDebitCardsView

Tipo: APIView

- **Descripción:**
Esta vista permite a los usuarios obtener una lista de las tarjetas de débito asociadas a su cuenta.
 - **Método:**
 - GET: Devuelve todas las tarjetas de débito asociadas al usuario autenticado.
 - **Permiso:**
Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Devuelve los datos de las tarjetas de débito asociadas al usuario.
-

11. SucursalListView

Tipo: generics.ListAPIView

- **Descripción:**
Esta vista permite obtener la lista de todas las sucursales disponibles en el sistema.
 - **Método:**
 - GET: Devuelve una lista de todas las sucursales registradas en el sistema.
 - **Respuesta:**
 - **200 OK:** Devuelve la lista de sucursales.
-

12. UpdateSucursalView

Tipo: APIView

- **Descripción:**
Permite a los usuarios elegir una sucursal y asignarla a su cuenta, siempre y cuando tengan el permiso para hacerlo. Una vez asignada la sucursal, se bloquea la posibilidad de cambiarla.

- **Método:**
 - POST: Recibe el ID de la sucursal seleccionada y actualiza la cuenta del usuario con esa sucursal. Desactiva la opción de cambiarla en el futuro.
 - **Permiso:**

Solo usuarios autenticados con el permiso adecuado (can_change_sucursal) pueden realizar esta acción.
 - **Respuesta:**
 - **200 OK:** Si la sucursal se actualiza correctamente.
 - **403 Forbidden:** Si el usuario no tiene permiso para cambiar la sucursal.
 - **404 Not Found:** Si la sucursal no existe.
-

13. ReactivateSucursalChangeView

Tipo: APIView

- **Descripción:**

Permite a un administrador reactivar el permiso para cambiar la sucursal de un usuario.
 - **Método:**
 - POST: Recibe el ID de un usuario y reactiva su permiso para cambiar la sucursal.
 - **Permiso:**

Solo los administradores pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Si el permiso se reactiva correctamente.
 - **404 Not Found:** Si el usuario no existe.
-

14. CheckSucursalPermissionView

Tipo: APIView

- **Descripción:**

Esta vista permite al usuario verificar si tiene permisos para cambiar la sucursal asociada a su cuenta.

- **Método:**
 - GET: Devuelve un booleano indicando si el usuario tiene permiso para cambiar la sucursal.
 - **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Si se puede cambiar la sucursal, se devuelve `can_change_sucursal: true`. Si no, se devuelve `false`.
-

15. AssignedSucursalView

Tipo: APIView

- **Descripción:**

Permite a los usuarios obtener la sucursal que tienen asignada.
 - **Método:**
 - GET: Devuelve los detalles de la sucursal asignada al usuario, incluyendo la provincia y la dirección.
 - **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Si el usuario tiene una sucursal asignada.
 - **404 Not Found:** Si el usuario no tiene una sucursal asignada.
-

16. UpdateUserProfileView

Tipo: APIView

- **Descripción:**

Permite a los usuarios actualizar su perfil, incluyendo nombre, apellido, correo electrónico, DNI y la imagen de perfil.
- **Método:**
 - PUT: Recibe los datos del perfil y actualiza la información del usuario. Los datos pueden ser opcionales, y solo los campos enviados serán actualizados.
- **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.

- **Respuesta:**
 - **200 OK:** Si el perfil se actualiza correctamente.
 - **400 Bad Request:** Si los datos enviados no son válidos.
-

17. GetUserProfileView

Tipo: APIView

- **Descripción:**

Permite a los usuarios obtener los datos de su perfil, como nombre, apellido, correo electrónico, DNI e imagen de perfil.
 - **Método:**
 - GET: Devuelve los datos del perfil del usuario autenticado.
 - **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Si el perfil del usuario se encuentra y se devuelve correctamente.
 - **404 Not Found:** Si no se encuentra el perfil del usuario.
-

18. GetUserBalanceView

Tipo: APIView

- **Descripción:**

Permite a los usuarios consultar su saldo disponible.
 - **Método:**
 - GET: Devuelve el saldo disponible del usuario autenticado.
 - **Permiso:**

Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Si el saldo se obtiene correctamente.
 - **404 Not Found:** Si el usuario no es encontrado.
-

19. GetUserBalanceAndCBUView

Tipo: APIView

- **Descripción:**
Permite a los usuarios consultar su saldo y CBU (Código Bancario Único).
 - **Método:**
 - GET: Devuelve el saldo y el CBU del usuario autenticado.
 - **Permiso:**
Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Si los datos se obtienen correctamente, se devuelve el saldo y el CBU.
-

20. ConsultarCuentaView

Tipo: APIView

- **Descripción:**
Permite a los usuarios consultar su alias o CBU. La consulta se determina por un parámetro (tipo) pasado en la solicitud.
- **Método:**
 - GET: Devuelve el CBU o alias del usuario según el parámetro tipo en la solicitud.
- **Permiso:**
Solo usuarios autenticados pueden acceder a esta vista.
- **Respuesta:**
 - **200 OK:** Si se consulta correctamente el alias o el CBU del usuario.

Aquí tienes la documentación detallada para las últimas vistas de tu backend Django:

21. GetFullUserDetailsView

Tipo: APIView

- **Descripción:**
Esta vista permite mostrar la información completa de un usuario autenticado, incluyendo datos del perfil, saldo, sucursal asignada, alias, CBU y la imagen de perfil.
- **Método:**

- GET: Devuelve todos los detalles del usuario, como su nombre, apellido, email, dirección, saldo, alias, CBU y más.
 - **Permiso:**
Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Si los datos del usuario se obtienen correctamente.
 - **404 Not Found:** Si no se encuentra el perfil del usuario.
 - **500 Internal Server Error:** Si ocurre un error inesperado.
-

22. TransferirAPIView

Tipo: APIView

- **Descripción:**
Permite a los usuarios realizar una transferencia de dinero a otro usuario registrado. La cantidad a transferir se resta del saldo del remitente y se suma al saldo del receptor.
 - **Método:**
 - POST: Recibe el alias o CBU del receptor, el monto a transferir y una descripción opcional. Si el saldo del usuario es suficiente, realiza la transferencia.
 - **Permiso:**
Solo usuarios autenticados pueden acceder a esta vista.
 - **Respuesta:**
 - **200 OK:** Si la transferencia se realiza con éxito.
 - **400 Bad Request:** Si el monto es inválido, insuficiente o si faltan datos.
 - **404 Not Found:** Si no se encuentra al receptor.
 - **400 Bad Request (Saldo insuficiente):** Si el saldo del usuario no es suficiente para completar la transferencia.
-

23. TransferenciasAPIView

Tipo: APIView

- **Descripción:**
Permite a los usuarios consultar todas las transferencias que han

realizado o recibido. Las transferencias se ordenan por la fecha más reciente.

- **Método:**

- GET: Devuelve una lista de todas las transferencias realizadas o recibidas por el usuario autenticado, incluyendo detalles como el nombre del remitente y receptor, la cantidad transferida, la fecha y la descripción.

- **Permiso:**

No se especifica un permiso, ya que el usuario solo puede acceder a sus propias transferencias. Sin embargo, debe estar autenticado.

- **Respuesta:**

- **200 OK:** Si las transferencias se obtienen correctamente, se devuelve la lista de transferencias.

models.py

1. Sucursal

Descripción:

Este modelo almacena la información sobre las sucursales disponibles. Cada sucursal tiene un nombre, provincia y dirección.

- **Campos:**

- nombre: Nombre de la sucursal (opcional).
- provincia: Provincia donde se encuentra la sucursal.
- direccion: Dirección de la sucursal.

- **Métodos:**

- `__str__`: Devuelve una representación en cadena de la sucursal con la provincia y dirección.
-

2. CustomUser (Extiende AbstractUser)

Descripción:

Este modelo extiende `AbstractUser` para incluir campos adicionales específicos de la aplicación, como dirección, saldo, sucursal asignada, alias, CBU y otros.

- **Campos:**

- `address`: Dirección del usuario.
- `balance`: Saldo del usuario (decimal).
- `sucursal`: Relación con la Sucursal asignada.
- `can_change_sucursal`: Booleano que indica si el usuario puede cambiar su sucursal.
- `profile_image`: Imagen de perfil del usuario.
- `dni`: Documento Nacional de Identidad del usuario.
- `alias`: Alias único asignado al usuario.
- `cbu`: CBU único del usuario.

- **Métodos:**

- `generate_alias`: Genera un alias aleatorio.
- `generate_cbu`: Genera un CBU aleatorio de 22 dígitos.
- `save`: Sobrescribe el método `save` para generar alias y CBU si no existen al momento de guardar el usuario.
- `__str__`: Representación en cadena del nombre de usuario.

3. Loan

Descripción:

Este modelo gestiona los préstamos solicitados por los usuarios, incluyendo el monto, motivo, comentario, fecha de solicitud y estado de aprobación.

- **Campos:**

- `user`: Relación con el usuario que solicita el préstamo.
- `monto`: Monto del préstamo solicitado.
- `motivo`: Motivo del préstamo.
- `comentario`: Comentario opcional del usuario.
- `fecha_solicitud`: Fecha de la solicitud del préstamo (automática).
- `aprobado`: Booleano que indica si el préstamo ha sido aprobado.

- fecha_aprobacion: Fecha de aprobación (si es aplicable).
 - **Métodos:**
 - __str__: Representación en cadena con el nombre de usuario, monto y estado de aprobación.
-

4. DebitCard

Descripción:

Este modelo almacena información sobre las tarjetas de débito asociadas a los usuarios.

- **Campos:**
 - user: Relación con el usuario propietario de la tarjeta.
 - numero: Número único de la tarjeta.
 - tipo: Tipo de tarjeta (por defecto "Débito").
 - fecha_emision: Fecha en la que se emite la tarjeta (automática).
 - fecha_vencimiento: Fecha de vencimiento de la tarjeta.
 - cvv: Código CVV de la tarjeta (opcional).
 - **Métodos:**
 - save: Sobrescribe el método save para generar automáticamente un número de tarjeta y CVV si no se proporcionan.
 - __str__: Representación en cadena con el tipo de tarjeta y número.
-

5. UserProfile

Descripción:

Este modelo almacena información adicional del perfil de usuario, como nombres, email, DNI y la imagen de perfil.

- **Campos:**
 - user: Relación uno a uno con el usuario (CustomUser).
 - first_name: Primer nombre del usuario.
 - last_name: Apellido del usuario.
 - email: Dirección de correo electrónico del usuario.
 - dni: Documento de identidad.

- `profile_image`: Imagen de perfil del usuario.
 - **Métodos:**
 - `__str__`: Representación en cadena con el nombre de usuario.
-

6. Transfer

Descripción:

Este modelo gestiona las transferencias de dinero entre usuarios, incluyendo detalles como el remitente, receptor, monto, descripción y estado.

- **Campos:**
 - `sender`: Relación con el usuario remitente.
 - `receiver`: Relación con el usuario receptor.
 - `amount`: Monto de la transferencia.
 - `timestamp`: Fecha y hora de la transferencia (automática).
 - `status`: Estado de la transferencia (por defecto "Pendiente").
 - `description`: Descripción de la transferencia (opcional).
- **Métodos:**
 - `__str__`: Representación en cadena que incluye el nombre del remitente y receptor, el monto y el estado.
 - `to_dict`: Devuelve un diccionario con los detalles de la transferencia.

serializers.py

1. SucursalSerializer

Descripción:

Convierte los datos del modelo Sucursal en un formato JSON y viceversa. Solo incluye los campos relevantes: provincia y dirección.

- **Campos:**
 - `provincia`: Provincia de la sucursal.
 - `direccion`: Dirección de la sucursal.
-

2. UserSerializer

Descripción:

Este serializer se utiliza para representar los datos básicos del usuario, incluyendo la sucursal a la que pertenece y el balance.

- **Campos:**
 - id: ID del usuario.
 - username: Nombre de usuario.
 - email: Correo electrónico del usuario.
 - address: Dirección del usuario (opcional).
 - balance: Saldo del usuario.
 - sucursal: Sucursal del usuario (representada mediante SucursalSerializer).
 - profile_image: Imagen de perfil del usuario (opcional).
-

3. LoanSerializer

Descripción:

Este serializer maneja la conversión de un préstamo en un formato JSON. Incluye una representación del estado del préstamo (Aprobado/Rechazado) mediante un método personalizado.

- **Campos:**
 - id: ID del préstamo.
 - monto: Monto del préstamo.
 - motivo: Motivo del préstamo.
 - comentario: Comentario adicional sobre el préstamo.
 - status: Estado del préstamo, calculado por el método `get_status`.
 - fecha_solicitud: Fecha en la que se realizó la solicitud.
 - **Método:**
 - `get_status`: Devuelve el estado del préstamo como "Aprobado" o "Rechazado" según el campo aprobado.
-

4. DebitCardSerializer

Descripción:

Este serializer convierte los datos de las tarjetas de débito en un formato JSON.

Los campos incluyen el número de tarjeta, tipo, fechas de emisión y vencimiento, y el código CVV.

- **Campos:**
 - id: ID de la tarjeta.
 - numero: Número de la tarjeta (único).
 - tipo: Tipo de tarjeta (por defecto "Débito").
 - fecha_emision: Fecha de emisión de la tarjeta.
 - fecha_vencimiento: Fecha de vencimiento de la tarjeta.
 - cvv: Código CVV de la tarjeta (opcional).
-

5. UserProfileSerializer

Descripción:

Este serializer convierte los datos del perfil completo del usuario en un formato JSON. Incluye detalles del usuario como su nombre, dirección de email, DNI, alias, CBU, imagen de perfil, sucursal y balance.

- **Campos:**
 - username: Nombre de usuario.
 - email: Correo electrónico del usuario.
 - first_name: Primer nombre.
 - last_name: Apellido.
 - dni: Documento de identidad.
 - profile_image: Imagen de perfil.
 - sucursal: Nombre de la sucursal donde trabaja el usuario.
 - balance: Saldo del usuario.
 - alias: Alias del usuario.
 - cbu: CBU del usuario.
-

6. TransferSerializer

Descripción:

Este serializer representa las transferencias de dinero entre usuarios. Incluye el nombre del remitente y el receptor, así como detalles de la transferencia como el monto, la descripción y el estado.

- **Campos:**

- id: ID de la transferencia.
- sender_name: Nombre del remitente (calculado por el método get_sender_name).
- receiver_name: Nombre del receptor (calculado por el método get_receiver_name).
- amount: Monto de la transferencia.
- timestamp: Fecha y hora de la transferencia.
- description: Descripción de la transferencia.
- status: Estado de la transferencia.
- **Métodos:**
 - get_sender_name: Devuelve el nombre completo del remitente.
 - get_receiver_name: Devuelve el nombre completo del receptor.

urls.py

Rutas Definidas:

1. Rutas de los ViewSets

- **users/:**

Ruta para gestionar los usuarios a través del UserViewSet. Este ViewSet normalmente manejaría operaciones como listar, crear, actualizar y eliminar usuarios. Está registrado en el router.
- **loans/:**

Ruta para gestionar los préstamos a través del LoanViewSet. Similar al ViewSet de usuarios, este manejaría las operaciones CRUD para los préstamos.
- **Nota:** Comentada la línea de cards que parece estar relacionada con las tarjetas, pero no está en uso actualmente.

2. Rutas de Vistas Basadas en Clases (Class-Based Views)

- **register/:**

Ruta para el registro de un usuario, mapeada a la vista RegisterView.
- **login/:**

Ruta para el inicio de sesión de un usuario, mapeada a la vista LoginView.

- **api/validate-token/:**
Ruta para validar un token de autenticación, mapeada a la vista ValidateTokenView.
- **solicitar-prestamo/:**
Ruta para solicitar un préstamo, mapeada a la vista SolicitarPrestamoView.
- **mis-prestamos/:**
Ruta para obtener los préstamos de un usuario, mapeada a la vista ObtenerPrestamosView.
- **eliminar-prestamo/<int:pk>/:**
Ruta para eliminar un préstamo específico, mapeada a la vista EliminarPrestamoView. Utiliza un parámetro de tipo pk (ID del préstamo).
- **tarjetas/:**
Ruta para listar las tarjetas de débito de un usuario, mapeada a la vista ListDebitCardsView.
- **sucursales/:**
Ruta para listar todas las sucursales, mapeada a la vista SucursalListView.
- **update-sucursal/:**
Ruta para actualizar la sucursal del usuario, mapeada a la vista UpdateSucursalView.
- **update-sucursal/<int:id>/:**
Ruta para actualizar la sucursal de una sucursal específica (mediante su ID), mapeada a la vista UpdateSucursalView.
- **reactivate-sucursal-change/<int:user_id>/:**
Ruta para reactivar el cambio de sucursal para un usuario, mapeada a la vista ReactivateSucursalChangeView.
- **check-sucursal-permission/:**
Ruta para verificar si un usuario tiene permisos para cambiar de sucursal, mapeada a la vista CheckSucursalPermissionView.
- **update-profile/:**
Ruta para actualizar el perfil del usuario, mapeada a la vista UpdateUserProfileView.

- **profile/:**
Ruta para obtener el perfil básico del usuario, mapeada a la vista GetUserProfileView.
- **user-balance/:**
Ruta para obtener el balance de un usuario, mapeada a la vista GetUserBalanceView.
- **user-cuenta/:**
Ruta para consultar la cuenta del usuario, mapeada a la vista ConsultarCuentaView.
- **perfil-completo/:**
Ruta para obtener la información completa del usuario, mapeada a la vista GetFullUserDetailsView.
- **transferir/:**
Ruta para realizar una transferencia de dinero entre usuarios, mapeada a la vista TransferirAPIView.
- **transferencias/:**
Ruta para obtener la lista de transferencias realizadas o recibidas por el usuario, mapeada a la vista TransferenciasAPIView.
- **mi-sucursal/:**
Ruta para obtener la sucursal asignada al usuario, mapeada a la vista AssignedSucursalView.

Archivos Estáticos y Medios:

El archivo también incluye configuraciones para servir archivos estáticos y archivos de medios (como imágenes de perfil), especialmente en el entorno de desarrollo (DEBUG):

```
+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Esto asegura que los archivos de medios (por ejemplo, imágenes subidas por el usuario) sean accesibles durante el desarrollo.

Aclaración: Hay algunas funcionalidades que no están respondiendo de forma correcta, por lo tanto, es probable que a veces funcionen o no. Por otro lado, el Registro sí funciona, aunque diga “Error al registrarse”. Más

allá de este mensaje, la funcionalidad se completa y registra un nuevo usuario.