

Documentación de API

1. Registrar Usuario

****Endpoint:**** POST /api/auth/register

****Descripción:**** Registra un nuevo usuario.

****Cuerpo de la solicitud (JSON):**

```
{  
  "email": "usuario@dominio.com",  
  "name": "Nombre Usuario",  
  "password": "contraseña",  
  "university": "Universidad"  
}
```

****Respuesta:****

- 200 OK: Usuario registrado correctamente.
- 401 Unauthorized: Ya existe un usuario con el mismo correo.

2. Iniciar Sesión

****Endpoint:**** POST /api/auth/login

****Descripción:**** Inicia sesión de un usuario con email y contraseña.

****Cuerpo de la solicitud (JSON):**

```
{  
  "email": "usuario@dominio.com",  
  "password": "contraseña"  
}
```

****Respuesta:****

- 200 OK: Devuelve un token de acceso.
- 401 Unauthorized: Credenciales incorrectas.
- 400 Bad Request: Datos mal formateados.

3. Iniciar Sesión con Token

****Endpoint:**** POST /api/auth/loginWithToken

****Descripción:**** Inicia sesión con un token de autenticación previamente emitido.

****Cuerpo de la solicitud (JSON):**

```
{  
  "email": "usuario@dominio.com",  
  "token": "token_de_acceso"  
}
```

****Respuesta:****

- 200 OK: Token válido.
- 400 Bad Request: Token inválido o expirado.

4. Cerrar Sesión

****Endpoint:**** POST /api/auth/logout

****Descripción:**** Cierra sesión del usuario.

****Cuerpo de la solicitud (JSON):**

```
{  
  "email": "usuario@dominio.com",  
  "token": "token_de_acceso"  
}
```

****Respuesta:****

- 200 OK: Cierre de sesión exitoso.
- 400 Bad Request: Token no encontrado o inválido.

5. Obtener Usuario

****Endpoint:**** GET /api/users/me

****Descripción:**** Devuelve los detalles del usuario autenticado.

****Parámetros de la solicitud (Query):**

- token: Token de autenticación.

****Respuesta:****

- 200 OK: Información del usuario.
- 400 Bad Request: Token inválido o no encontrado.

6. Recuperar Contraseña

****Endpoint:**** POST /api/auth/forgotPassword

****Descripción:**** Recupera la contraseña del usuario a través del correo electrónico.

****Cuerpo de la solicitud (JSON):**

```
{  
  "email": "usuario@dominio.com"  
}
```

****Respuesta:****

- 200 OK: Correo de recuperación enviado.
- 400 Bad Request: Error al recuperar la contraseña.

7. Crear Intercambio

****Endpoint:**** POST /api/exchanges

****Descripción:**** Crea un nuevo intercambio de idiomas.

****Cuerpo de la solicitud (JSON):**

```
{  
  "token": "token_de_acceso",  
  "nativeLanguage": "Español",  
  "targetLanguage": "Inglés",  
  "educationalLevel": "Secundaria",  
  "academicLevel": "Básico",  
  "beginDate": "01-01-2025",  
  "endDate": "01-06-2025",  
  "quantityStudents": 10,  
  "university": "Universidad Ejemplo"  
}
```

****Respuesta:****

- 201 Created: Intercambio creado con éxito.
- 400 Bad Request: Token inválido o error al procesar la solicitud.

8. Obtener Intercambios

****Endpoint:**** GET /api/exchanges

****Descripción:**** Obtiene los intercambios disponibles con parámetros de búsqueda.

****Parámetros de la solicitud (Query):**

- nativeLanguage: Idioma nativo.
- targetLanguage: Idioma objetivo.
- educationalLevel: Nivel educativo.
- academicLevel: Nivel académico.
- beginDate: Fecha de inicio.
- endDate: Fecha de fin.
- quantityStudentsMin: Mínimo de estudiantes.
- quantityStudentsMax: Máximo de estudiantes.
- university: Universidad.
- userId: ID de usuario.
- status: Estado del intercambio.

****Respuesta:****

- 200 OK: Lista de intercambios.

9. Obtener Intercambio por ID

****Endpoint:** GET /api/exchanges/**

****Descripción:** Obtiene un intercambio específico por su ID.**

****Parámetros de la solicitud (Query):**

- **exchangeId:** ID del intercambio.

****Respuesta:****

- **200 OK:** Detalles del intercambio.

- **404 Not Found:** No se encontró el intercambio.

POST /api/messages

Descripción: Este endpoint permite enviar un mensaje entre dos usuarios. El cuerpo de la solicitud debe contener los campos idUserSender, idUserRecipient, message, typeMessage (opcional, por defecto "text") y token para autenticación.

Respuesta exitosa: 201 Created. Devuelve el mensaje creado.

Respuesta de error: 400 Bad Request si el cuerpo de la solicitud es incorrecto, 401 Unauthorized si el token es inválido o ha expirado.

PUT /api/notifications/read

Descripción: Este endpoint marca todos los mensajes de una conversación entre dos usuarios como leídos. El cuerpo de la solicitud debe contener los campos idUserSender, idUserRecipient y token para autenticación.

Respuesta exitosa: 200 OK. Devuelve un mensaje indicando que los mensajes fueron marcados como leídos.

Respuesta de error: 400 Bad Request si el formato JSON es incorrecto, 401 Unauthorized si el token es inválido o ha expirado, 204 No Content si no hay mensajes.

GET /api/messages/conversation

Descripción: Este endpoint obtiene todos los mensajes entre dos usuarios. Los parámetros idUserSender, idUserRecipient y token deben enviarse como parámetros de la URL.

Respuesta exitosa: 200 OK. Devuelve una lista de mensajes entre los usuarios solicitados.

Respuesta de error: 401 Unauthorized si el token es inválido o ha expirado, 204 No Content si no hay mensajes.

PUT /api/exchanges/

Descripción: Este endpoint permite actualizar la información de un intercambio específico. El id del intercambio debe enviarse como parámetro de la URL, y el cuerpo de la solicitud debe contener un JSON con los campos a actualizar. Se valida un token de autenticación en la solicitud.

Respuesta exitosa: 200 OK. Devuelve el intercambio actualizado.

Respuesta de error: 400 Bad Request si la solicitud tiene un error, 401 Unauthorized si el token es inválido o ha expirado, 404 Not Found si no se encuentra el intercambio.

DELETE /api/exchanges/

Descripción: Este endpoint elimina un "exchange" dado un ID y un token de autenticación.

Parámetros:

- `id` (String): El ID del "exchange" a eliminar.
- `token` (String): El token para autenticación.

Respuesta exitosa: 200 OK. Devuelve un mensaje confirmando la eliminación del "exchange".

Respuesta de error:

- 400 Bad Request si el token es inválido o ha expirado.
 - 404 Not Found si el "exchange" no se encuentra.
 - 500 Internal Server Error si ocurre un error en el proceso de eliminación.
-

POST /api/groups

Descripción: Este endpoint crea un nuevo grupo. Requiere un token de autenticación y los datos del grupo.

Cuerpo de la solicitud:

- `token` (String): El token para autenticación.
- `name` (String): El nombre del grupo.
- `languageLevel` (int): El nivel de idioma del grupo.
- `quantity` (int): La cantidad de estudiantes en el grupo.
- `language` (String): El idioma del grupo.

Respuesta exitosa: 201 Created. Devuelve un mensaje indicando que el grupo fue creado correctamente.

Respuesta de error:

- 400 Bad Request si el token es inválido o ha expirado.
 - 500 Internal Server Error si ocurre un error al crear el grupo.
-

GET /api/groups

Descripción: Este endpoint obtiene los grupos asociados al ID del profesor, usando un token de autenticación.

Parámetros:

- `token` (String): El token para autenticación.
Respuesta exitosa: 200 OK. Devuelve el grupo encontrado asociado al ID del profesor.
Respuesta de error:
 - 400 Bad Request si el token es inválido o ha expirado.
 - 404 Not Found si no se encuentra el grupo.
-

GET /api/groups/

Descripción: Este endpoint obtiene un grupo por su ID, usando un token de autenticación.

Parámetros:

- `groupId` (String): El ID del grupo a obtener.
 - `token` (String): El token para autenticación.
Respuesta exitosa: 200 OK. Devuelve el grupo encontrado.
Respuesta de error:
 - 400 Bad Request si el token es inválido o ha expirado.
 - 404 Not Found si no se encuentra el grupo con el ID proporcionado.
-

PUT /api/groups/

Descripción: Este endpoint actualiza un grupo existente con los nuevos valores proporcionados en la solicitud.

Parámetros:

- `id` (String): El ID del grupo a actualizar.
 - `groupJson` (String): El cuerpo de la solicitud con los nuevos datos del grupo (name, languageLevel, quantity, language, userId).
Respuesta exitosa: 200 OK. Devuelve el grupo actualizado.
Respuesta de error:
 - 400 Bad Request si los datos proporcionados son incorrectos o si ocurre un error al procesar la solicitud.
 - 404 Not Found si el grupo no se encuentra.
-

DELETE /api/groups/

Descripción: Este endpoint elimina un grupo dado un ID y un token de autenticación.

Parámetros:

- `id` (String): El ID del grupo a eliminar.
- `token` (String): El token para autenticación.

Respuesta exitosa: 200 OK. Devuelve un mensaje confirmando la eliminación del grupo.

Respuesta de error:

- 400 Bad Request si el token es inválido o ha expirado.
 - 404 Not Found si el grupo no se encuentra.
 - 500 Internal Server Error si ocurre un error en el proceso de eliminación.
-

POST /api/upload

Descripción: Este endpoint permite cargar un archivo. El archivo será subido a Cloudinary.

Parámetros:

- `file` (MultipartFile): El archivo que se desea cargar.

Respuesta exitosa: 200 OK. Devuelve la URL del archivo cargado.

Respuesta de error:

- 500 Internal Server Error si ocurre un error al subir el archivo.
-

POST /api/preferences

Descripción: Este endpoint crea las preferencias del usuario, como el modo oscuro, notificaciones y el idioma, usando un token de autenticación.

Cuerpo de la solicitud:

- `token` (String): El token para autenticación.
- `darkMode` (boolean): Estado del modo oscuro.
- `notifications` (boolean): Estado de las notificaciones.
- `language` (String): El idioma preferido.

Respuesta exitosa: 201 Created. Devuelve un mensaje indicando que las preferencias fueron creadas.

Respuesta de error:

- 400 Bad Request si el token es inválido o ha expirado.
 - 500 Internal Server Error si ocurre un error al crear las preferencias.
-

GET /api/users/me/preferences

Descripción: Este endpoint obtiene las preferencias del usuario autenticado mediante su

token.

Parámetros:

- `token` (String): El token para autenticación.
Respuesta exitosa: 200 OK. Devuelve las preferencias del usuario.
Respuesta de error:
 - 400 Bad Request si el token es inválido o ha expirado.
 - 404 Not Found si no se encuentran las preferencias para el usuario.
-

PUT /api/users/me/preferences

Descripción: Este endpoint actualiza las preferencias del usuario.

Cuerpo de la solicitud:

- `token` (String): El token para autenticación.
 - `darkMode` (boolean): Estado del modo oscuro.
 - `notifications` (boolean): Estado de las notificaciones.
 - `language` (String): El idioma preferido.
Respuesta exitosa: 200 OK. Devuelve un mensaje indicando que las preferencias fueron actualizadas correctamente.
Respuesta de error:
 - 400 Bad Request si el token es inválido o ha expirado.
 - 404 Not Found si no se encuentran las preferencias para el usuario.
-

PUT /api/auth/update

Descripción: Este endpoint actualiza los datos del usuario, como su nombre, universidad y fotos de perfil.

Cuerpo de la solicitud:

- `token` (String): El token para autenticación.
 - `name` (String): El nombre del usuario (opcional).
 - `university` (String): La universidad del usuario (opcional).
 - `urlProfilePicture` (String): URL de la foto de perfil (opcional).
 - `urlHeaderPicture` (String): URL de la foto de cabecera (opcional).
Respuesta exitosa: 200 OK. Devuelve un mensaje indicando que el usuario fue actualizado correctamente.
Respuesta de error:
 - 400 Bad Request si los datos proporcionados son incorrectos o si ocurre un error al procesar la solicitud.
 - 404 Not Found si no se encuentra el usuario.
-

GET /api/users/image

Descripción: Este endpoint obtiene la imagen de perfil del usuario basado en su email.

Parámetros:

- `email` (String): El email del usuario.

Respuesta exitosa: 200 OK. Devuelve la URL de la imagen de perfil del usuario.

Respuesta de error:

- 404 Not Found si el usuario no se encuentra.