**Documentación de API Endpoints**

**1. Verificación de Conexión a la Base de Datos**

* **Endpoint**: GET /verificar\_conexion
* **Descripción**: Verifica la conexión activa con la base de datos.
* **Respuestas**:
  + 200 OK: Conexión establecida correctamente.
  + 500 Internal Server Error: Error al conectar a la base de datos.

**2. Registro de Usuarios**

* **Endpoint**: POST /usuarios
* **Descripción**: Permite el registro de un nuevo usuario en el sistema.
* **Cuerpo de la solicitud** (JSON):

json

Copiar código

{

"nombre": "Nombre",

"apellido": "Apellido",

"email": "email@example.com",

"password": "contraseña"

}

* **Respuestas**:
  + 201 Created: Usuario registrado correctamente.
  + 500 Internal Server Error: Error durante el registro del usuario.

**3. Autenticación de Usuarios (Login)**

* **Endpoint**: POST /login
* **Descripción**: Autentica a un usuario y devuelve un token JWT.
* **Cuerpo de la solicitud** (JSON):

json

Copiar código

{

"email": "email@example.com",

"password": "contraseña"

}

* **Respuestas**:
  + 200 OK: Autenticación correcta, devuelve token JWT.
  + 401 Unauthorized: Credenciales inválidas.
  + 500 Internal Server Error: Error al autenticar al usuario.

**4. Obtener Tarjetas del Usuario Autenticado**

* **Endpoint**: GET /usuario/tarjetas
* **Descripción**: Obtiene todas las tarjetas bancarias asociadas al usuario autenticado.
* **Headers**:
  + Authorization: Bearer <token>
* **Respuestas**:
  + 200 OK: Lista de tarjetas obtenidas exitosamente.
  + 401 Unauthorized: Token no proporcionado o inválido.
  + 500 Internal Server Error: Error al obtener las tarjetas.

**5. Obtener Transacciones del Usuario**

* **Endpoint**: GET /transacciones
* **Descripción**: Devuelve todas las transacciones asociadas a las tarjetas del usuario autenticado.
* **Headers**:
  + Authorization: Bearer <token>
* **Respuestas**:
  + 200 OK: Transacciones obtenidas correctamente.
  + 401 Unauthorized: Token no proporcionado o inválido.
  + 500 Internal Server Error: Error al obtener las transacciones.

**6. Registrar Nueva Transacción**

* **Endpoint**: POST /transacciones
* **Descripción**: Registra una nueva transacción en una tarjeta específica del usuario autenticado.
* **Cuerpo de la solicitud** (JSON):

json

Copiar código

{

"id\_tarjeta": 123,

"monto": 100.50,

"descripcion": "Compra en tienda"

}

* **Headers**:
  + Authorization: Bearer <token>
* **Respuestas**:
  + 201 Created: Transacción registrada correctamente.
  + 403 Forbidden: La tarjeta no pertenece al usuario autenticado.
  + 500 Internal Server Error: Error al registrar la transacción.

**7. Solicitar Nueva Tarjeta**

* **Endpoint**: POST /tarjetas/solicitar
* **Descripción**: Permite a un usuario autenticado solicitar una nueva tarjeta de crédito/debito.
* **Headers**:
  + Authorization: Bearer <token>
* **Respuestas**:
  + 201 Created: Tarjeta solicitada correctamente, devuelve ID de la nueva tarjeta.
  + 500 Internal Server Error: Error al solicitar la tarjeta.

**8. Activar Tarjeta**

* **Endpoint**: POST /tarjetas/activar
* **Descripción**: Activa una tarjeta específica del usuario autenticado.
* **Cuerpo de la solicitud** (JSON):

json

Copiar código

{

"id\_tarjeta": 123

}

* **Headers**:
  + Authorization: Bearer <token>
* **Respuestas**:
  + 200 OK: Tarjeta activada correctamente.
  + 404 Not Found: Tarjeta no encontrada o no pertenece al usuario.
  + `500 Internal Server Error