**API reciclaDUOC**

**Endpoint registroUsuario**

Descripción: Registra un nuevo usuario en el sistema.

**Información del Endpoint:**

Método: POST

URL: /api/usuario/registroUsuario

**Parámetros de la solicitud:**

**Body:**

| {  "email": “string”,  “nombre”: “string”,  "password": “string”  } |
| --- |

**Respuestas:**

| **200 Éxito**  {  "mensaje": "Usuario registrado correctamente",  "usuario": {  "id": number,  "email": string  }  }  **400 Bad Request**  {  "error": "Email, password y nombre son obligatorios"  }  **409 Conflict**  {  "error": "El usuario ya existe"  }  **500 Internal Server Error**  {  "error": "Error en el servidor"  } |
| --- |

**Seguridad:**

- La contraseña se encripta usando bcrypt con un factor de costo de 10

- Se verifica la existencia previa del email para evitar duplicados

**Notas:**

- El id del usuario es retornado como número

- Solo se requiere email y password para el registro

- La contraseña se almacena de forma segura (hasheada)

**Endpoint de loginUsuario**

Descripción: Permite a los usuarios autenticarse en el sistema mediante sus credenciales (email y contraseña).

**Información del Endpoint:**

Método: POST

URL: /api/usuario/loginUsuario

Autenticación: No requerida

**Parámetros de la solicitud:**

**Body:**

| {  "email": "string",  "password": "string"  } |
| --- |

**Respuestas:**

| **200 Exito**  {  "mensaje": "Login exitoso",  "usuario": {  "id": "number",  "email": "string"  },  "token": "string"  }  **400 Bad Request**  {  "error": "Email y password son obligatorios"  }  **401 Unauthorized**  {  "error": "Credenciales inválidas"  }  **500 Internal Server Error**  {  "error": "Error en el servidor"  } |
| --- |

**Seguridad:**

Utiliza bcrypt para la comparación segura de contraseñas

Genera un token JWT con expiración de 5 horas

No devuelve la contraseña en la respuesta

**Endpoint de getPerfil**

Este endpoint permite obtener el perfil completo de un usuario autenticado, incluyendo sus datos personales y el historial de reciclajes realizados.

**Información del Endpoint:**

URL: /api/usuario/getPerfil

Método: GET

Autenticación: Requerida (JWT Token)

**Parámetros de la solicitud:**

**Headers:**

| Authorization: Bearer <token> |
| --- |

**Respuestas:**

| **200 Exito**  {  "mensaje": "Perfil obtenido correctamente",  "perfil": {  "id\_usuario": "number",  "email": "string",  "nombre": "string",  "puntos": "number"  },  "reciclajes": [  {  "id\_reciclaje": "number",  "fecha": "string",  "foto": "string",  "productos": [  {  "id\_producto": "number",  "nombre": "string",  "descripcion": "string",  "valor\_punto": "number",  "cantidad": "number"  }  ]  }  ]  }  **401 Unauthorized**  {  "error": "Token requerido"  }  **404 Not Found**  {  "error": "Usuario no encontrado"  }  **500 Internal Server Error**  {  "error": "Error en el servidor"  } |
| --- |

**Características:**

Obtiene información básica del usuario (id, email, nombre, puntos)

Incluye historial completo de reciclajes.

Los reciclajes están ordenados por fecha descendente.

Cada reciclaje incluye los productos reciclados con sus detalles.

**Endpoint de updatePerfil**

Este endpoint permite actualizar los datos del perfil de un usuario autenticado. Se puede actualizar el nombre.

**Información del Endpoint:**

URL: /api/usuario/updatePerfil

Método: PUT

Autenticación: Requerida (JWT Token)

**Parámetros de la solicitud:**

**Headers:**

| Authorization: Bearer <token> |
| --- |

**Body:**

Se debe enviar al menos uno de los campos para realizar la actualización.

| {  "nombre": "string"  } |
| --- |

**Respuestas:**

| **200 Exito**  {  "mensaje": "Perfil actualizado correctamente",  "perfil": {  "id\_usuario": "number",  "email": "string",  "nombre": "string",  "puntos": "number"  }  }  **400 Bad Request**  {  "error": "El nombre es obligatorio"  }  **401 Unauthorized**  {  "error": "Token requerido"  }  **403 Forbidden**  {  “error”: "Token inválido o expirado"  }  **500 Internal Server Error**  {  "error": "Error en el servidor"  } |
| --- |

**Endpoint de registroReciclaje**

Este endpoint permite registrar un nuevo reciclaje con uno o varios productos, calculando y actualizando los puntos del usuario.

**Información del Endpoint:**

URL: /api/reciclaje/registroReciclaje

Método: POST

Autenticación: Requerida

**Parámetros de la solicitud:**

**Headers:**

| Authorization: Bearer <token> |
| --- |

**Body:**

Se debe enviar al menos uno de los campos para realizar la actualización.

| {  "id\_deposito": "number",  "foto": "string", // Opcional  "productos": [  {  "id\_producto": "number",  "cantidad": "number"  },  {  "id\_producto": "number",  "cantidad": "number"  }  ]  } |
| --- |

**Respuestas:**

| **200 Exito**  {  "mensaje": "Reciclaje registrado correctamente",  "reciclaje": {  "id\_reciclaje": “number”,  "fecha": "string",  "foto": "string",  "productos": [  {  "id\_producto": “number”,  "nombre": "string",  "descripcion": "string",  "valor\_punto": “number”,  "cantidad": “number”  }  ],  "puntos\_ganados": “number”  }  }  **400 Bad Request**  {  "error": "Datos incompletos. Se requiere id\_deposito y al menos un producto"  }  {  "error": "Estructura de productos inválida. Cada producto debe tener id\_producto y cantidad mayor a 0"  }  **401 Unauthorized**  {  "error": "Token requerido"  }  **403 Forbidden**  {  "error": "Token inválido o expirado"  }  **500 Internal Server Error**  {  "error": "Error en el servidor"  } |
| --- |

**Funcionalidad:**

Valida el token JWT del usuario

Registra el nuevo reciclaje en la base de datos

Registra los productos asociados al reciclaje

Calcula los puntos ganados basándose en el valor de cada producto

Actualiza los puntos del usuario

Retorna los detalles del reciclaje creado

**Notas Adicionales:**

Los puntos se calculan multiplicando el valor\_punto de cada producto por la cantidad reciclada

La fecha se genera automáticamente al momento del registro

Se pueden registrar múltiples productos en un solo reciclaje

La foto es opcional y puede ser null