

Mini-dossier técnico – Modulo Persona API

Proyecto: Codium

Desarrolladores: Eberson Guayllas – Wilmer Pardo – Santiago Tamayo

1. Introducción

El modulo persona da los endpoints utilizados en la gestión de usuarios dentro del sistema Codium. Se realizan operaciones CRUD (crear, leer, actualiza, eliminar) sobre los registros de los usuarios que utilizan la plataforma, del mismo modo se puede obtener información paginada o individual.

Los endpoints están implementados mediante Flask Blueprints y se comunican con el modelo PersonaModel, este realiza consulta a la base de datos MySQL con la conexión db.py.

Este modulo resulta importante en el backend ya que se encarga de administrar la información base de los usuarios, esto para su autenticación y otras funcionalidades.

2. Endpoints RESTful

GET /personas

Obtiene una lista paginada de todas las personas activas en la base de datos. La ruta para obtener el listado es: <http://127.0.0.1:5000/api/personas>.

Los códigos de respuesta pueden ser 200 (lista de personas obtenidas correctamente) o 500 (error interno del servidor)

Ejemplo de solicitud: GET <http://127.0.0.1:5000/api/personas>.

Ejemplo de respuesta:

```
[  
 {  
   "apellidos": "Tamayo",  
   "correo": "santiago@example.com",  
   "id_persona": 2,  
   "id_rol": 2,  
   "nombre": "Santiago",  
   "nombre_usuario": "stamayo",  
   "num_retos_resueltos": 0,  
   "puntaje_total": 0  
 }]  
 ]
```

GET /personas/<id_persona>

Este método obtiene la información de una persona específica mediante su ID. La ruta requerida es: http://127.0.0.1:5000/api/personas/<id_persona>. Y esta nos puede generar los códigos de respuesta 200, 404 (persona no encontrada) y 500 (error interno del servidor)

El parámetro a utilizar sería el id_persona (int)

Ejemplo de solicitud: GET http://127.0.0.1:5000/api/personas/5/

Ejemplo de respuesta:

```
{  
    "apellidos": "Pardo",  
    "correo": "wilmer.pardo @correo.com",  
    "id_persona": 1,  
    "id_rol": 2,  
    "nombre": "Wilmer",  
    "nombre_usuario": "wpardo",  
    "num_retos_resueltos": 0,  
    "puntaje_total": 0  
}
```

POST /personas

Crea un registro de usuario en la base de datos, asegurando que la contraseña se encripta antes de ser almacenada en la base de datos para proteger la información del usuario. La ruta para registrar una persona es: <http://127.0.0.1:5000/api/personas>.

Los parámetros a utilizar son los presentes en el siguiente body, donde todos los campos son strings. Los códigos que se pueden generar son 201, 400 y 500.

Ejemplo de solicitud:

POST http://127.0.0.1:5000/api/personas

```
{  
    "nombre": "Wilmer",  
    "apellidos": "Pardo",  
    "correo": "wilmer.pardo@correo.com ",  
    "contrasena_plana": "passWilner123",  
    "nombre_usuario": "wpardo"  
}
```

Ejemplo de respuesta:

```
{  
    "id_persona": 1,  
    "message": "Persona creada exitosamente"  
}
```

PUT /personas/<id_persona>

Actualiza los datos permitidos de una persona (nombre, apellidos, nombre_usuario, token_refresco). Su ruta para actualizar una persona es: <http://127.0.0.1:5000/api/personas/<idPersona>>

Ejemplo de solicitud:

PUT http://127.0.0.1:5000/api/personas/3

```
{  
    nombre: "daniel"  
}
```

Ejemplo de respuesta

```
{  
    message: "Persona actualizada"  
}
```

DELETE /personas/<id_persona>

Este método no elimina la persona de forma definitiva, sino la desactiva realizando un “**borrado lógico**”, cambiando su estado a inactivo (esta_activo = FALSE). La ruta para eliminar una persona es: <http://127.0.0.1:5000/api/personas/<idPersona>>. Y el parámetro es id_persona (int), esta solicitud puede generar los códigos 200 y 500.

Ejemplo de solicitud:

DELETE http://127.0.0.1:5000/api/personas/3

Ejemplo de respuesta:

```
{  
    "message": "Persona desactivada"  
}
```

3. Capturas de Postman/ Swagger

Get /personas

```

[{"id": 1, "apellido": "Pardo", "nombre": "Wilmer", "correo": "wilmer.pardo@correo.com", "id_persona": 1, "id_rol": 2, "puntaje_total": 8}, {"id": 2, "apellido": "Tamayo", "nombre": "Wilmer", "correo": "wilmer.tamayo@correo.com", "id_persona": 2, "id_rol": 2, "puntaje_total": 8}, {"id": 3, "apellido": "Tamayo", "nombre": "Emerson", "correo": "emerson.guaylana@correo.com", "id_persona": 3, "id_rol": 2, "puntaje_total": 8}, {"id": 4, "apellido": "Tamayo", "nombre": "Santiago", "correo": "santiago.tamayo@correo.com", "id_persona": 4, "id_rol": 2, "puntaje_total": 8}, {"id": 5, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 5, "id_rol": 1, "puntaje_total": 8}, {"id": 6, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 6, "id_rol": 1, "puntaje_total": 8}, {"id": 7, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 7, "id_rol": 1, "puntaje_total": 8}, {"id": 8, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 8, "id_rol": 1, "puntaje_total": 8}, {"id": 9, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 9, "id_rol": 1, "puntaje_total": 8}, {"id": 10, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 10, "id_rol": 1, "puntaje_total": 8}, {"id": 11, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 11, "id_rol": 1, "puntaje_total": 8}, {"id": 12, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 12, "id_rol": 1, "puntaje_total": 8}, {"id": 13, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 13, "id_rol": 1, "puntaje_total": 8}, {"id": 14, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 14, "id_rol": 1, "puntaje_total": 8}, {"id": 15, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 15, "id_rol": 1, "puntaje_total": 8}, {"id": 16, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 16, "id_rol": 1, "puntaje_total": 8}, {"id": 17, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 17, "id_rol": 1, "puntaje_total": 8}, {"id": 18, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 18, "id_rol": 1, "puntaje_total": 8}, {"id": 19, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 19, "id_rol": 1, "puntaje_total": 8}, {"id": 20, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 20, "id_rol": 1, "puntaje_total": 8}, {"id": 21, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 21, "id_rol": 1, "puntaje_total": 8}, {"id": 22, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 22, "id_rol": 1, "puntaje_total": 8}, {"id": 23, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 23, "id_rol": 1, "puntaje_total": 8}, {"id": 24, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 24, "id_rol": 1, "puntaje_total": 8}, {"id": 25, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 25, "id_rol": 1, "puntaje_total": 8}, {"id": 26, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 26, "id_rol": 1, "puntaje_total": 8}, {"id": 27, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 27, "id_rol": 1, "puntaje_total": 8}, {"id": 28, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 28, "id_rol": 1, "puntaje_total": 8}, {"id": 29, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 29, "id_rol": 1, "puntaje_total": 8}, {"id": 30, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 30, "id_rol": 1, "puntaje_total": 8}, {"id": 31, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 31, "id_rol": 1, "puntaje_total": 8}, {"id": 32, "apellido": "Guzman", "nombre": "Daniela", "correo": "daniela.guzman@correo.com", "id_persona": 32, "id_rol": 1, "puntaje_total": 8}

```

GET /personas/<id_persona>

Código 200 al encontrar una persona:

```

PUT http://127.0.0.1:5000/api/personas/3
{
  "nombre": "daniel"
}
200 OK
{
  "message": "Persona actualizada"
}

```

Código 404 cuando se ingrese un id incorrecto:

```

GET http://127.0.0.1:5000/api/personas/5
404 NOT FOUND
{
  "error": "Persona no encontrada"
}

```

POST /personas

Código 201 al ingresar un usuario a la BD:

```

POST http://127.0.0.1:5000/api/personas
{
  "apellido": "Cale", "correo": "estefania.cale@correo.com", "nombre": "Estefania", "nombre_usuario": "Dific", "password": "123456789", "num_reto_resuelto": 8, "puntaje_total": 0
}
201 CREATED
{
  "id_persona": 4,
  "message": "Persona creada exitosamente"
}

```

Código 409 al violar la integridad de los datos:

```

POST http://127.0.0.1:5000/api/personas
{
  "apellido": "Tamayo", "correo": "santiago.tamayo@correo.com", "nombre": "Santiago", "nombre_usuario": "Santiago", "password": "123456789", "num_reto_resuelto": 8, "puntaje_total": 0, "contraseña_plana": "posseSantiago466"
}
409 CONFLICT
{
  "error": "El correo electrónico ya está registrado."
}

```

Datos erróneos al ingresar un usuario:

The screenshot shows a Postman request to `http://127.0.0.1:5000/api/personas` using the `POST` method. The body contains the following JSON:

```
1 {
2     "contraseña_plana": "123",
3     "nombre": "jorge",
4     "apellidos": "lopez",
5     "correo": 21,
6     "nombra_usuario": "si"
7 }
```

The response status is `400 BAD REQUEST` with a duration of `8 ms` and a size of `202 B`. The error message is:

```
1 {
2     "error": "Campo 'correo' debe ser texto (string)"
3 }
```

`PUT /personas/<id_persona>`

Código 200 al actualizar el dato de un usuario:

The screenshot shows a Postman request to `http://127.0.0.1:5000/api/personas/3` using the `PUT` method. The body contains the following JSON:

```
1 {
2     "nombre": "Daniel"
3 }
```

The response status is `200 OK` with a duration of `63 ms` and a size of `238 B`. The message is:

```
1 {
2     "Message": "Personas actualizadas"
3 }
```

Código 400 al ingresar un dato erróneo:

The screenshot shows a Postman request to `http://127.0.0.1:5000/api/personas/2` using the `PUT` method. The body contains the following JSON:

```
1 {
2     "correo": 23
3 }
```

The response status is `400 BAD REQUEST` with a duration of `18 ms` and a size of `279 B`. The error message is:

```
1 {
2     "error": "No se proporcionaron datos válidos para actualizar"
3 }
```

`DELETE /personas/<id_persona>`

Desactivación de un usuario activo:

The screenshot shows a Postman request to `http://127.0.0.1:5000/api/personas/4` using the `DELETE` method. The response status is `200 OK` with a duration of `48 ms` and a size of `236 B`. The message is:

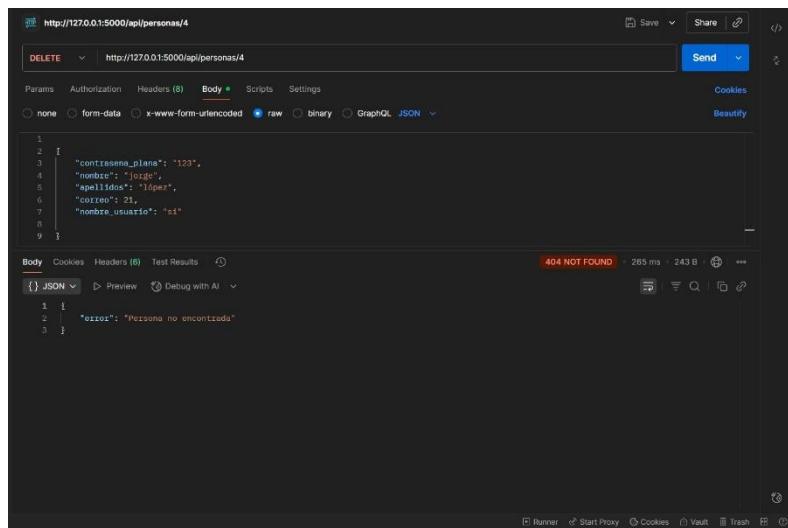
```
1 {
2     "message": "Personas desactivadas"
3 }
```

Intento de desactivar un usuario inexistente:

The screenshot shows a Postman request to `http://127.0.0.1:5000/api/personas/5` using the `DELETE` method. The response status is `404 NOT FOUND` with a duration of `16 ms` and a size of `243 B`. The error message is:

```
1 {
2     "error": "Personas no encontrada"
3 }
```

Error al desactivar una cuenta inactiva:



The screenshot shows a Postman interface with a DELETE request to `http://127.0.0.1:5000/api/personas/4`. The request body is a JSON object with fields: `contraseña_plana`, `nombre`, `apellidos`, `correo`, and `nombre_usuario`. The response status is `404 NOT FOUND` with a timestamp of `265 ms` and a size of `243 B`. The response body is a JSON object with a single key `error` containing the value `"Persona no encontrada"`.

```
1 [ { 2 "contraseña_plana": "123", 3 "nombre": "jorge", 4 "apellidos": "lopez", 5 "correo": "21", 6 "nombre_usuario": "el" 7 } ] 8 9 }
```

Body Cookies Headers (6) Test Results ⚙️

{ } JSON ▾ ▶ Preview ⚙️ Debug with AI ⚙️

```
1 [ { 2 "error": "Persona no encontrada" 3 } ]
```

404 NOT FOUND 265 ms 243 B ⚙️ 🔍 ⚙️ ⚙️

Runner Start Proxy Cookies Vault Trash ⚙️ ⚙️

4. Referencias

Repositorio: <https://github.com/SantiagoTamayo2002/codium>