Universidad Nacional de Loja
Computación
Autores: Josue Torres, Alexis Roman.
Fecha: 19/10/2025

Dossier descriptivo de los endpoints implementados en el sistema de Gestión de Empleados, en cada microservicio.

1. **Servicio de autenticación**

Ruta	Método	Descripción	Parámetros	Código De Respuesta
/auth/login	POST	Inicia sesión y devuelve un token JWT.	email password	200 (Ok) 401 (Sin autorización)
/auth/logout	POST	Invalida el token de sesión actual.	ninguno	204
/auth/refresh	POST	Refresca un token JWT expirado.	token_caducado	200

2. **Servicio de usuarios**

Ruta	Método	Descripción	Parámetros	Código De Respuesta
/users/register	POST	Gerente crea nuevo empleado	- nombres - apellidos - email - celular - rol	201 (Creado)
/users	GET	Gerente obtiene la lista de empleados	ninguno	200
/users/{user_id}	GET	Obtiene los detalles de un empleado	user_id	200 (Ok) 404 (No encontrado)
/users/{user_id}	PUT	Actualiza la información de un usuario	- user_id - parámetros a actualizar	200 (Ok) 404 (No encontrado)

/users/me	GET	Obtiene el perfil del usuario autenticado	ninguno	200
/users/set-password	POST	Crea la contraseña para el nuevo empleado creado	nueva contraseña	200

3. **Servicios de gestión de horarios**

ruta	método	descripción	parámetros	código de respuesta
/schedules/assignments	POST	Gerente asigna un horario a un empleado	user_id hora_entrada hora_salida	201
/users/{user_id}/schedule	GET	Obtiene el horario de un empleado	user_id	200

4. **Servicio de asistencia**

Ruta	**Método**	**Descripción**	**Parámetros**	**Código De Respuesta**
/attend/in	POST	Empleado registra la marcación	- fecha - hora	201
/users/{user_id}/attendance	GET	Obtiene el historial de marcaciones	ninguno	200
/attend/status	GET	Obtiene el estado actual (trabajando o no trabajando)	ninguno	200

5. **Servicio de reporte y cálculos**

Ruta	**Método**	**Descripción**	**Parámetros**	**Código De Respuesta**
/reports/hours/{user_id}	GET	Obtiene el resumen de horas	periodo de tiempo	200

		trabajadas, extras y atrasos en un periodo		
/reports/punctuality	GET	Reporte de puntualidad	ninguno	200
/reports/attendance	GET	Personas que están trabajando en ese momento	ninguno	200

6. **Servicio de notificaciones**

Ruta	Método	Descripción	Parámetros	Código De Respuesta
/notifications	GET	Obtiene todas las notificaciones para el usuario autenticado	Ninguno	200
/notifications/mark	POST	Marca notificaciones como leídas	id_notificacion	200

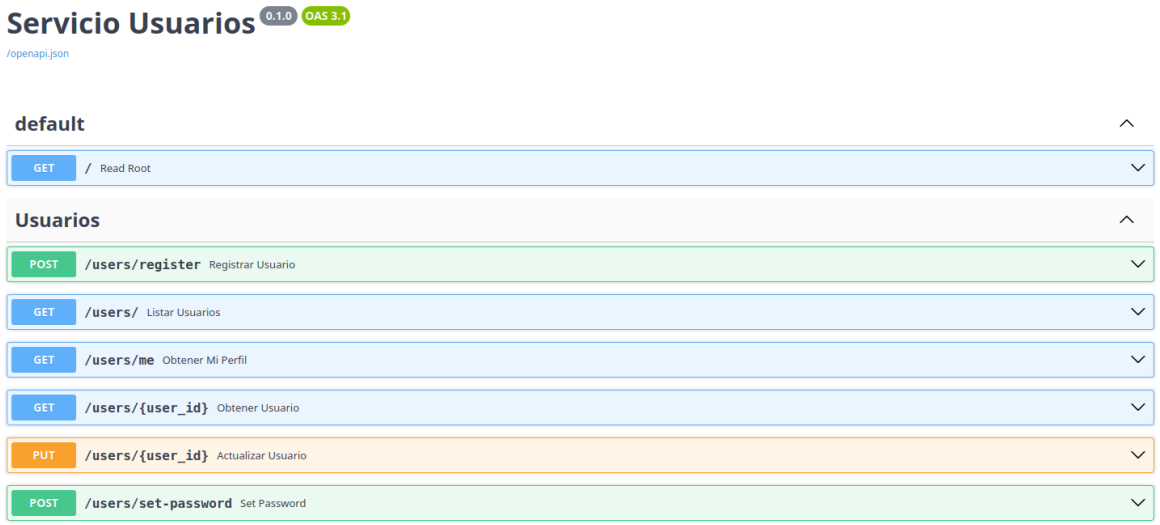
7. **Servicio de nómina**

Ruta	Método	Descripción	Parámetros	Código De Respuesta
/payrolls/generate	POST	Inicia el proceso para calcular el rol de pagos en un periodo de tiempo	- fecha inicial - fecha final	202
/users/{user_id}/payrolls	GET	Obtiene todos los roles de pago de un empleado	user_id	200
/payrolls/{payroll_id}	GET	Obtiene el detalle de un rol de pagos	payroll_id	200
/payrolls/submit	POST	Gerente envía el rol de pagos	payroll	202

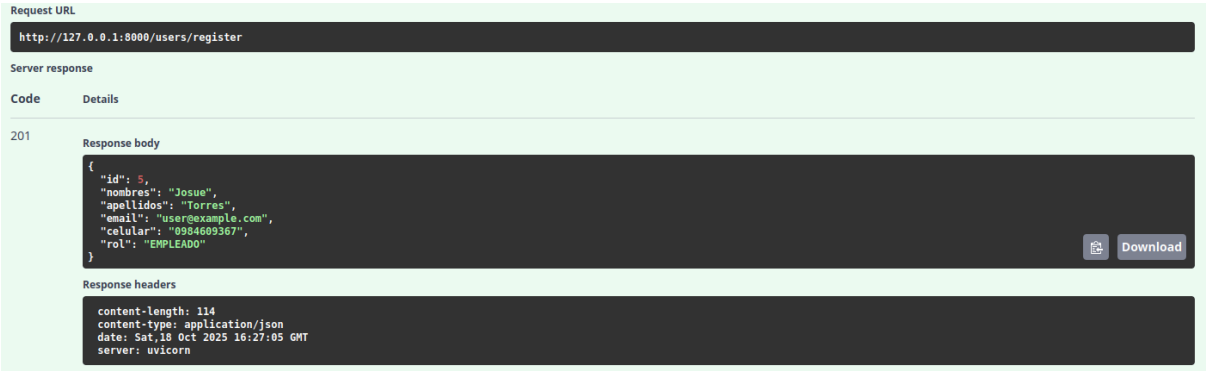
		al sistema de pagos externo		

Evidencias:

Servicio de Usuarios



/users/register:



Los parámetros enviados coinciden con los de respuesta

```python
@router.post("/register", response_model=Usuario, status_code=201)
def registrar_usuario(usuario: UsuarioCreate):
    nuevo_id = len(usuarios_db) + 1
    nuevo_usuario = Usuario(id=nuevo_id, **usuario.dict())
    usuarios_db.append(nuevo_usuario)
    return nuevo_usuario
```

/users


```python
@router.get("/", response_model=List[Usuario])
def listar_usuarios():
    return usuarios_db
```

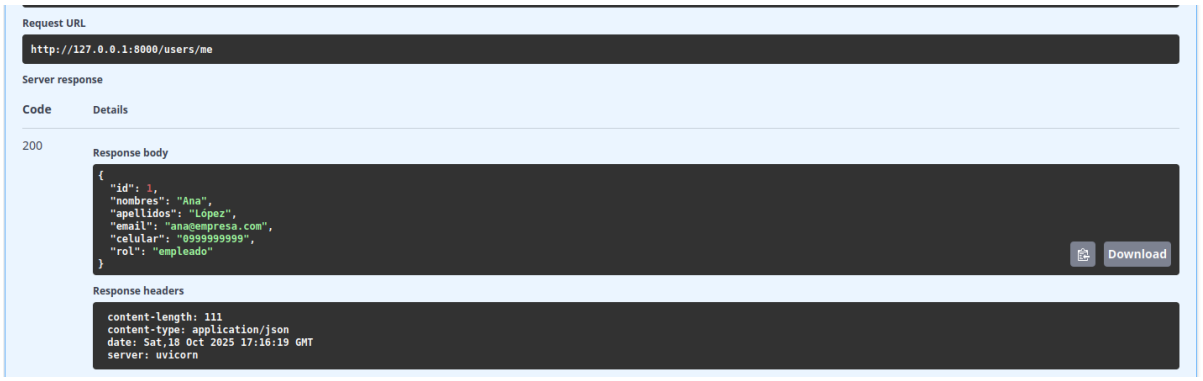
/users/{user_id} (GET)


```python
@router.get("/{user_id}", response_model=Usuario)
def obtener_usuario(user_id: int):
    for usuario in usuarios_db:
        if usuario.id == user_id:
            return usuario
    raise HTTPException(status_code=404, detail="Usuario no encontrado")
```

/users/{user_id} (PUT)


```python
@router.put("/{user_id}", response_model=Usuario)
def actualizar_usuario(user_id: int, datos: UsuarioCreate):
    for i, usuario in enumerate(usuarios_db):
        if usuario.id == user_id:
            actualizado = Usuario(id=user_id, **datos.dict())
            usuarios_db[i] = actualizado
            return actualizado
    raise HTTPException(status_code=404, detail="Usuario no encontrado")
```

/users/me



Nota: /users/me posteriormente debe extraer el usuario del token JWT o sesión actual.

```python
@router.get("/me", response_model=Usuario)
def obtener_mi_perfil():
    return usuario_actual
```

/users/set-password

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/users/set-password' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "user_id": 1,
  "password": "12345"
}'
```

Request URL

```
http://127.0.0.1:8000/users/set-password
```

Server response

Code	Details
200	Response body

```
{
  "message": "Contraseña establecida correctamente"
}
```

Download