

# CRUD DE USUARIO (PRELIMINAR)

## PEERHIVE

Objetivo: implementar correctamente el CRUD completo de la entidad “Usuario”

Lenguaje: Python

Formato de datos: JSON

### 1. Datos que contendrá el usuario

Atributo	Tipo	Obligatorio	Nota
ID	Entero	Si	Key primaria
Email	Texto/carácter	Si	Institucional/Único
Nombre	Texto	Si	
Contraseña	Alfanumérico	Si	Encriptada
Rol	Texto	Si	Admin/usuario

### 2. Operaciones del CRUD

#### 2.1 Crear usuario

Datos que recibe: nombre, mail, contraseña y rol

Validaciones:

- Nombre no vacío
- Formato del mail institucional valido y único
- Contraseña segura

Salida esperada:

- Mensaje de éxito
- Usuario creado con credenciales correctas

## 2.2 Leer Usuarios

Listar a todos los usuarios:

- Parámetros: filtrar por rol
- Devuelve lista de usuarios

Ver usuario por ID:

- Recibe el ID
- Devuelve los datos de usuario

## 2.3 Actualizar Usuario (Por discutir)

Datos requeridos:

- ID de usuario
- Datos para modificar: nombre, email o rol

Validaciones:

- Email institucional nuevo no repetido
- Verificar que el usuario exista

Salida: Usuario actualizado correctamente o error

## 2.4 Eliminar Usuario

Entrada: ID del usuario

Salida: Mensaje de confirmación o error

Validaciones:

- Usuario existente
- Que no este relacionado con datos importantes

## 3. Formato de entrada y salida de datos (API REST)

El modulo de usuarios se implementará en base a una API REST(MICROSOFT/TEAMS), utilizando métodos de HTTP comunes (GET, POST, PUT, DELETE) para las operaciones del CRUD. Al final la comunicación se realizará en formato JSON.

### 3.1 Endpoints del CRUD de usuario

#### Crear Usuario (CREATE)

- Método: POST (se usa para enviar datos al servidor)
- Endpoint: API/usuarios
- Descripción: registra un nuevo usuario en el sistema

#### Obtener todos los usuarios (READ)

- Método: GET (se usa para recuperar información de un recurso específico)
- Endpoint: API/usuarios
- Descripción: devuelve la lista de los usuarios

#### Obtener usuario por ID (READ)

- Método: GET (se usa para recuperar información de un recurso específico)
- Endpoint: API/usuarios
- Descripción: obtiene un usuario específico

#### Actualizar usuario (UPDATE)

- Método: PUT (se usa para reemplazar completamente un recurso ya existente o lo crea si no existe)
- Endpoint: API/usuarios (ID)
- Descripción: actualiza los datos de un usuario ya existente

#### Eliminar usuario (DELETE)

- Método: DELETE (se usa para eliminar un recurso específico)
- Endpoint: API/usuarios (ID)
- Descripción: elimina un usuario del sistema

**NOTA: SOLO USUARIOS AUTORIZADOS (admins) PUEDEN ACTUALIZAR O ELIMINAR USUARIOS**