

Tabla: users

HTTP	URL/ PARAM	DESCRIPCIÓN
GET	/users	Obtener la lista de todos los usuarios en la base de datos. (op)
GET	/users/:id	Obtener un solo usuario dado un id.
POST	/register	Crear un nuevo usuario. Se debe proporcionar name, email, password.
POST	/login	Login de usuario. Debe generar token

Tabla: security_box

HTTP	URL/ PARAM	DESCRIPCIÓN
GET	/listcategories	Listar por categorías, permitir ordenar por ALFABÉTICOS, FECHA CREACIÓN, CANTIDAD REGISTROS, y filtrar por favoritos o todos los registros. Verificar token
POST	/create	Crear un nuevo security_box. Se debe proporcionar name, icon, user_id, favorito. Las claves deben ser encriptadas utilizando JSON WEB TOKENS. Verificar token
GET	/OneBox/:id	Obtener un solo security_box dado un id. Verificar token
GET	/detbox/:id /detail	Ver el detalle de un security_box, con descriptación de contraseña. La contraseña debe ser descriptada del JWT, luego encriptada en base64 para enviar al frontend. Esta ruta requiere dos factores de protección: usuario autenticado y código verificado. Verificar token
PATCH	/update/:id	Actualizar los datos de un security_box dado un id solo los campos , nombre, favoritos, y status. Verificar token
DELETE	/del/:id	Cambiar el status de un security_box a DELETED. Verificar token (opt)

Tabla: credential_storage

HTTP	URL / PARAM	DESCRIPCIÓN
GET	/list	Obtener la lista de todos los credential_storages en la base de datos, ordenados según el orden dado por el usuario, solo enviar id y account. Los registros deberán ser paginados desde el backend. Verificar token
GET	/OneCredential/:id	Obtener un solo credential_storage dado un id. Verificar token
POST	/create	Crear un nuevo credential_storage. Se debe proporcionar account, password, security_box_id, pin_id. Verificar token
PATCH	/update/:id	Actualizar los datos de un credential_storage dado un id. Verificar token
DELETE	/del/:id	Eliminar un credential_storage dado un id. Verificar token
POST	/generate_password	Generar una contraseña aleatoria segura. La contraseña debe tener mínimo una minúscula, una mayúscula, números, caracteres especiales y mínimo 12 caracteres. Verificar token
PATCH	/position_cred	Actualizar la posición de un registro. Verificar token

Tabla: pin

HTTP	URL / PARAM	DESCRIPCIÓN
GET	/list	Obtener la lista de todos los pins en la base de datos. Verificar token (op)
GET	/pin/:id	Obtener un solo pin dado un id. Verificar token como login
POST	/create	Crear un nuevo pin. Se debe proporcionar code. Verificar token
POST	/save_pin	Guardar el PIN encriptado. Verificar token
POST	/validate_pin	Validar el PIN. Verificar token
PATCH	/update/:id	Actualizar el code de un pin dado un id. Verificar token
DELETE	/delete/:id	Eliminar un pin dado un id. Verificar token (op)

Endpoint para login de usuario

Tener en cuenta roles, trayectoria

Endpoint para crear usuario

Endpoint para listar categorías, el endpoint deberá permitir a los usuarios ordenar por las opciones ya dadas en el diseño ALFABETICO, FECHA CREACION, CANTIDAD RESPUESTAS, valores default

Endpoint para ver el detalle de un todo de contraseña, se deberá descryptar y por web token para enviar la contraseña, cuando se descrypte la contraseña deberá encriptar nuevamente en base64 y enviar al frontend y al frontend antes de mostrarle al usuario la contraseña se deberá usar una función de protección (el usuario solo recibirá y el de código verificado)

Endpoint para ver el detalle de un todo de contraseña

Endpoint para guardar el PIN (contraseña debe estar encriptada)

Endpoint para guardar una contraseña, se deberá validar que la contraseña tenga al menos una mayúscula, un número y una longitud mínima de 12 caracteres

Endpoint para guardar los datos de claves

Endpoint para guardar la clave de acceso al sistema

Endpoint para crear un folder

Endpoint para listar todos los registros guardados ordenados según el orden dado por el usuario, esto enviar el account, los registros deberán ser paginados desde el backend

Endpoint para actualizar la posición en la que guardó un registro, se deberá enviar la posición del elemento a actualizar, se deberá enviar también la posición del elemento con el que se desea cambiar

users	
id	uuid
name	varchar(100) NN
surname	varchar(100) NN
email	varchar(150) NN
cellphone	varchar(20) NN
password	varchar(255) NN
status	boolean

security_box	
id	uuid
name	varchar(100) NN
favorite	boolean
icon	varchar(20) NN
status	sec_status E
user_id	uuid NN

credential_storage	
id	uuid
account	varchar(100) NN
password	varchar(255) NN
description	text
code_1	varchar(20)
code_2	varchar(20)
security_box_id	uuid NN
pin_id	uuid NN

pin	
id	uuid
code	varchar(6)

