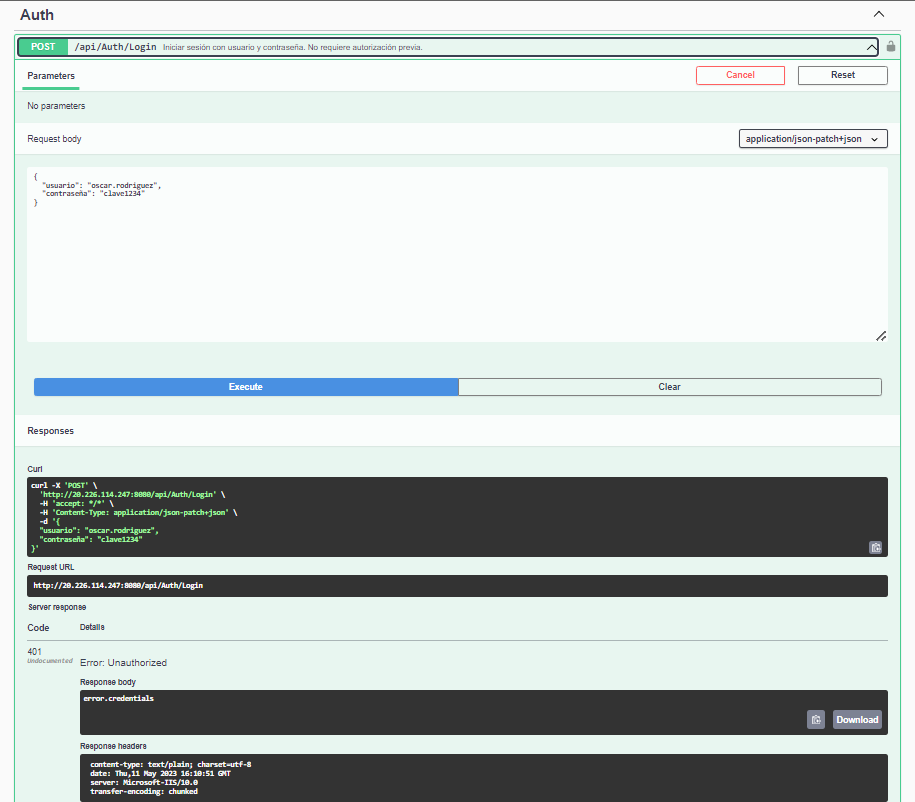
**Implementación de circuito de Login**

Ejemplos de uso en Swagger para probar el circuito. (http://20.226.114.247:8080/swagger/index.html)

**Caso de Uso**: el usuario aún no existe en el sistema.

Body JSON: Ingreso de un usuario y su contraseña:

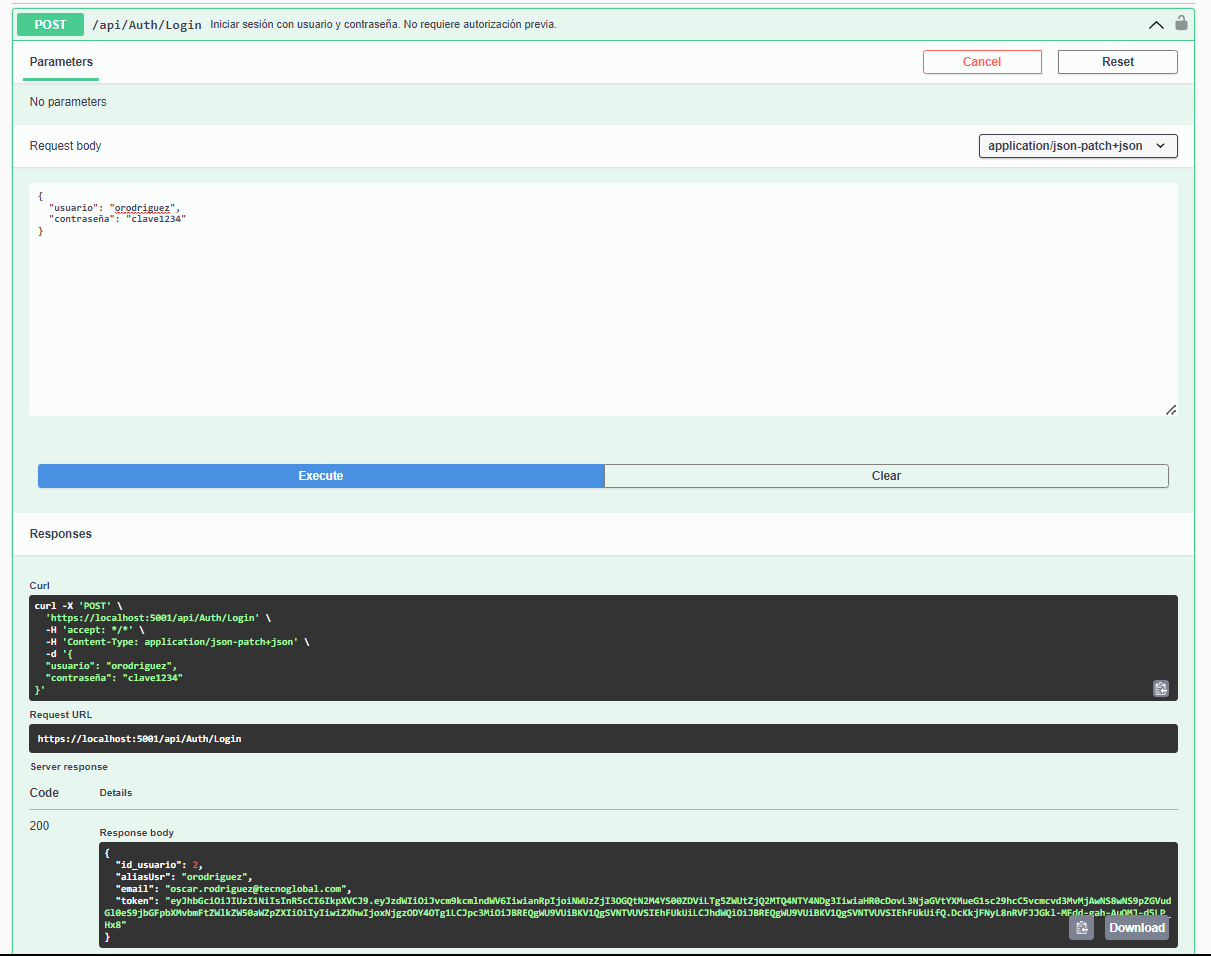
Response: error HTTP 401 (No autorizado)



**Casos de Uso**: el usuario existe en el sistema.

Body JSON: Ingreso de un usuario y su contraseña:

Response: HTTP 200 (Success) Quiere decir que el usuario se ha Autenticado y responde con un JSON con el token para la Autorización.

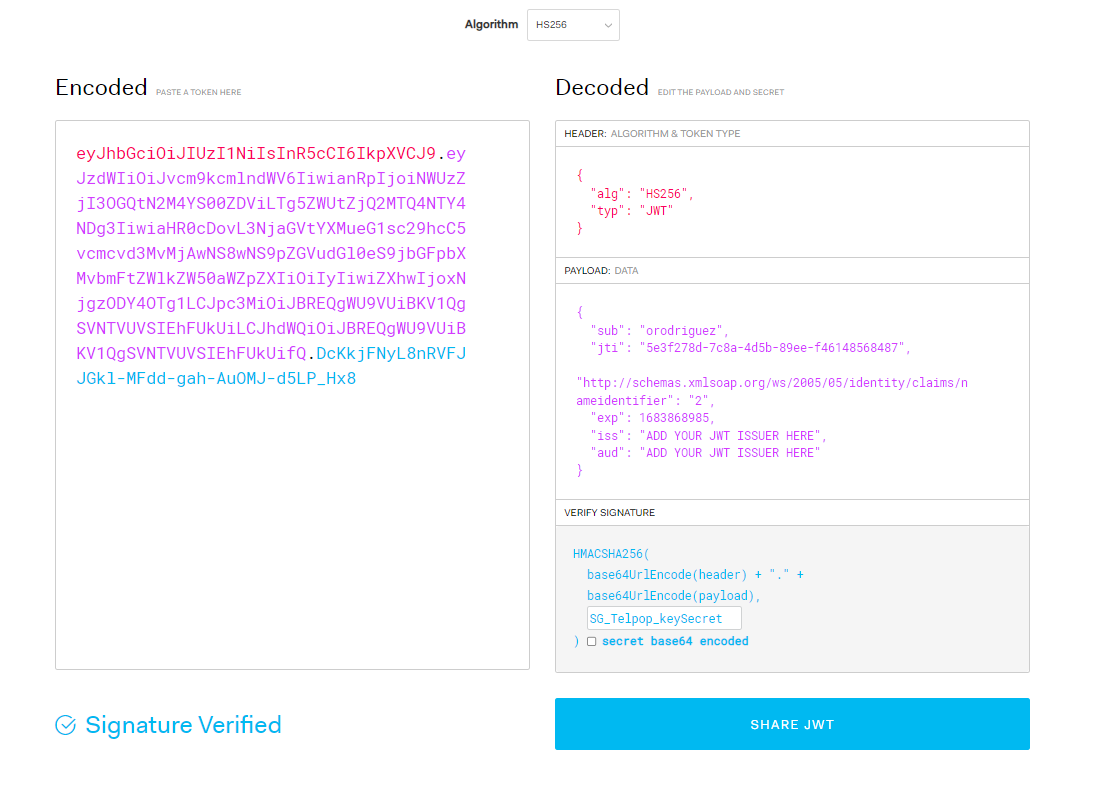


Nota importante: en esta primera versión de los endpoints, no estamos autorizando el uso mediante el token, luego en alguna charla consensuemos que utilizar….

Hay tres métodos de Autorización:

* ROLES
* CLAIMS
* POLICY

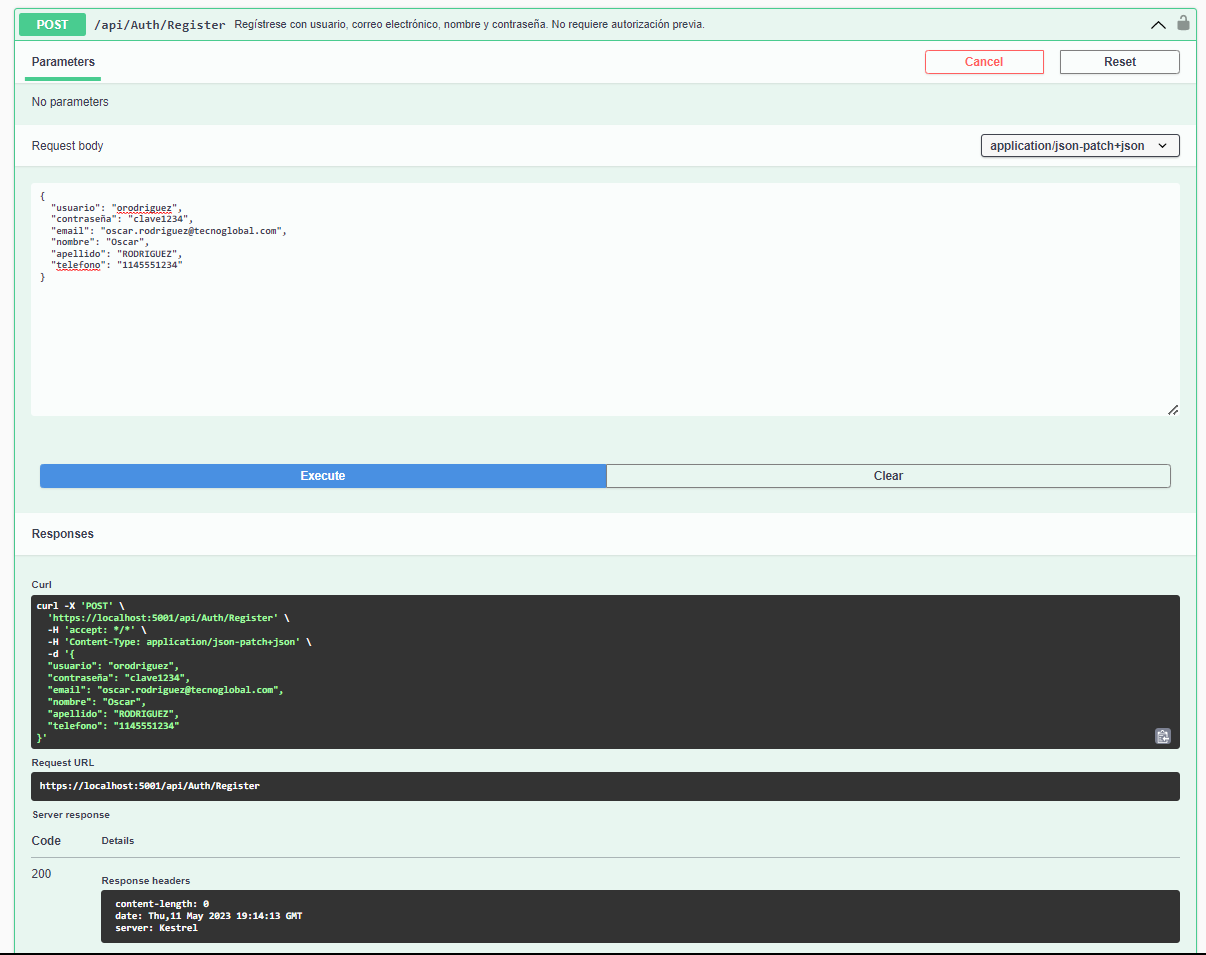
Esto significa que desacoplamos la parte visual de los permisos que tiene cada usuario, siempre ejecutara los endpoints para cada acción CRUD (Crear [POST], Leer [GET], Actualizar [PUT] y Eliminar [DELETE]) sin importar si el usuario autenticado puede o no hacerlas.

Este paso no es necesario, pero si validamos el Token en alguna página online de decodificación, nos muestra los datos que estamos compartiendo entre endpoints:  


**Casos de Uso**: Registrar el usuario en el sistema.

Body JSON: Ingreso de los datos del usuario y su contraseña

Response: HTTP 200 (Success) o algún HTTP de error (401, 403, 500, etc)



**Casos de Uso**: Activar el usuario en el sistema. Por seguridad el usuario se crea deshabilitado, y no tiene posibilidad de ingresar hasta que el mismo sea autorizado a operar con el sistema (puede ser el Administrador del sistema por ejemplo)

Body JSON: email del usuario

Response: HTTP 200 (Success) o algún HTTP de error (401, 403, 500, etc)

