Sprint 3

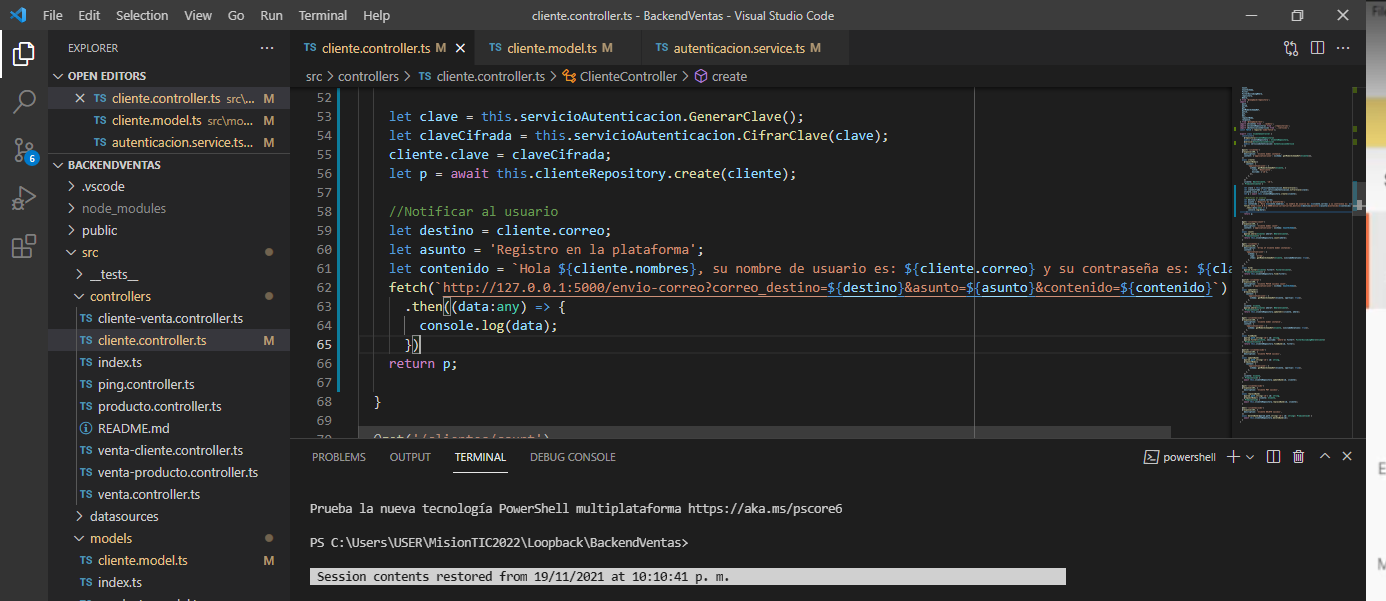
Instacion de paquetes:

$ npm i crypto-js

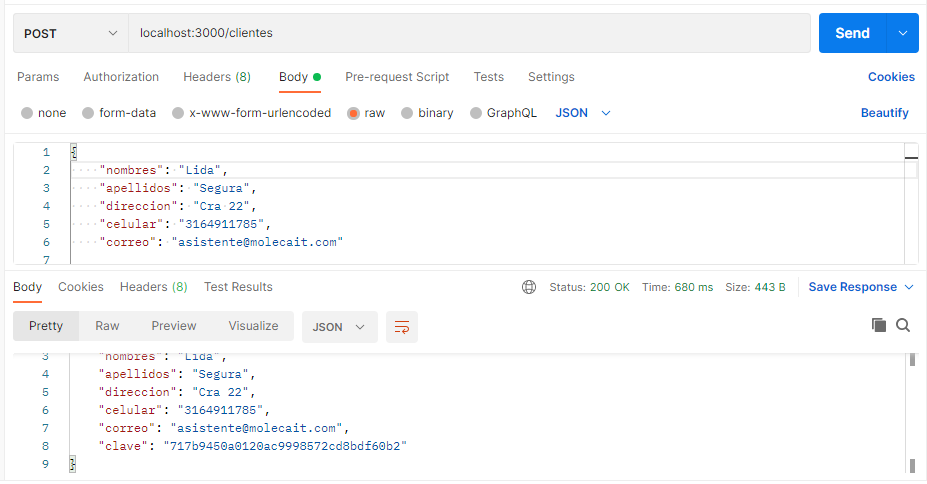
$ npm i password-generator

$ npm i [node-fetch@2.6.5](mailto:node-fetch@2.6.5)

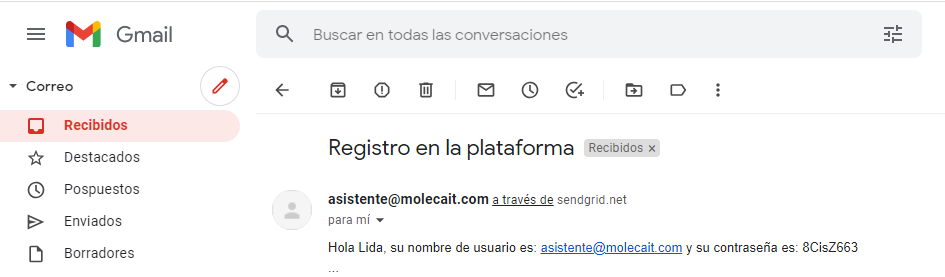
$ npm i @types/node-fetch



Pantallazo de código de email en el visual studio code.



Se realiza proceso en el Postman para el envío de la clave cifrada al correo electrónico



Confirmación que el correo con la clave cifrada llego efectivamente.

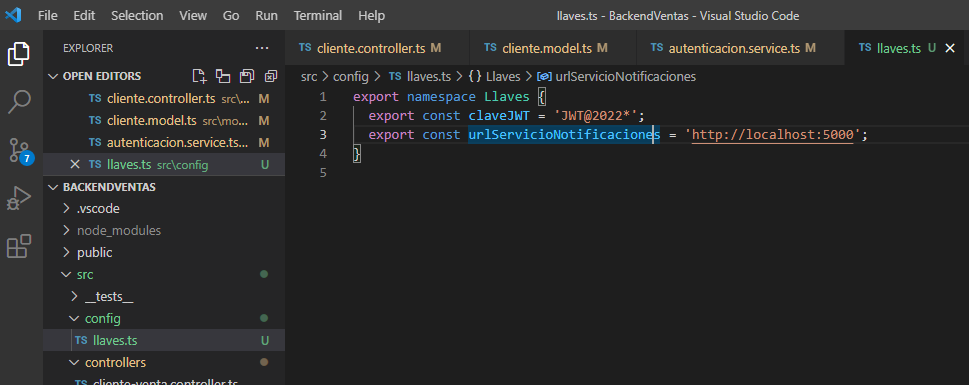
Video 4.2

Primera parte del video:

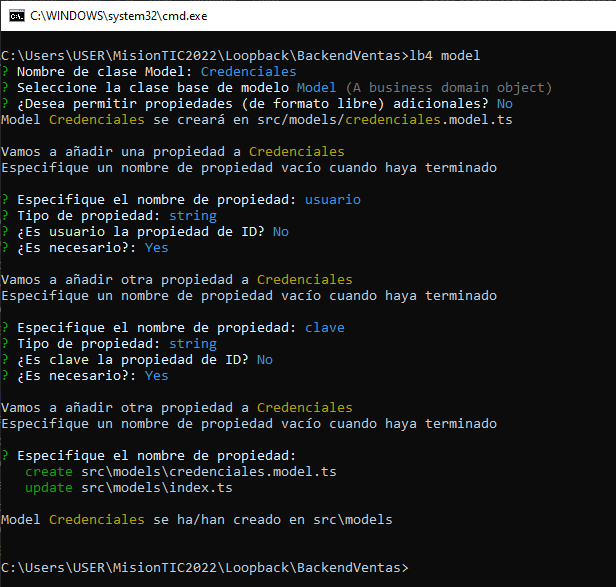
Instalar paquete: npm i jsonwebtoken

Importar cliente: import {Cliente} from ‘../models’;

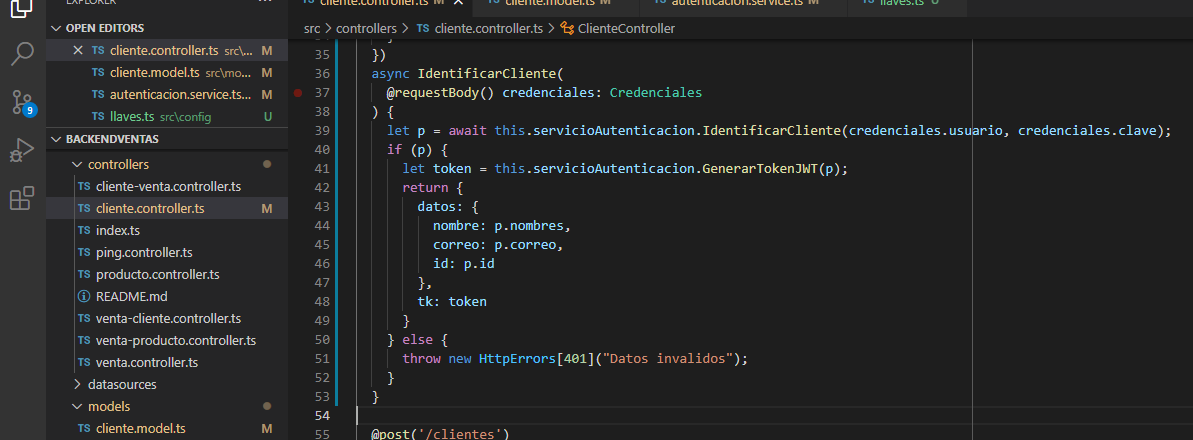
Se crea nueva carpeta: src/config/llaves.ts

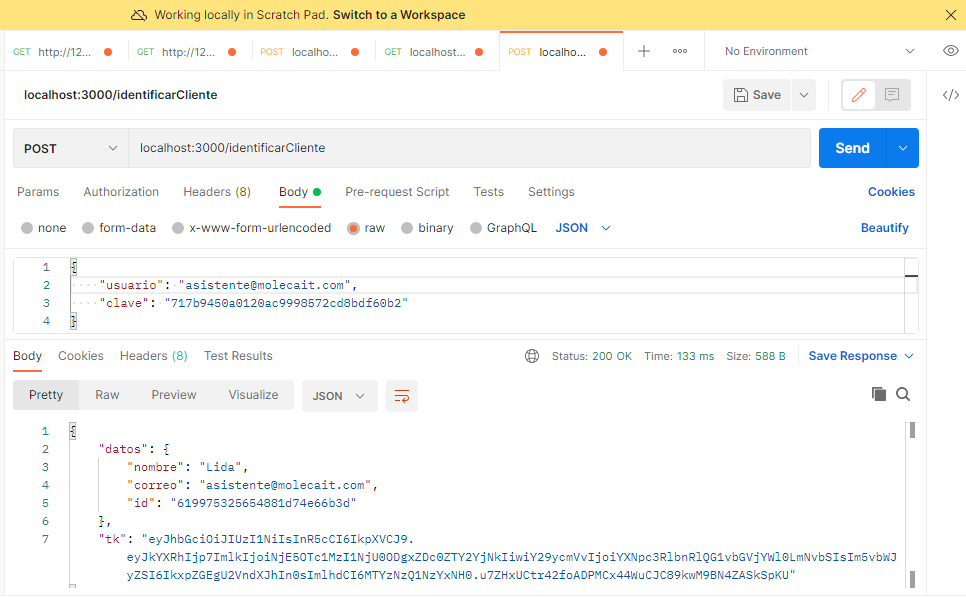


Se crea un nuevo modelo que se llama Credenciales

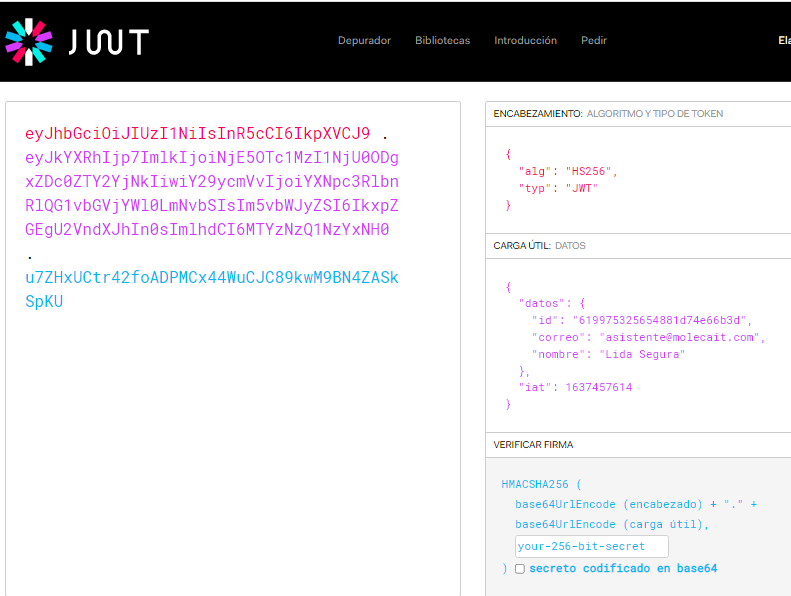


Código para identificar cliente con token





Se hace la prueba en Postman, dando como respuesta el token.



Se hace la confirmación del token en [www.jwt.io](http://www.jwt.io), donde validan que es valido.

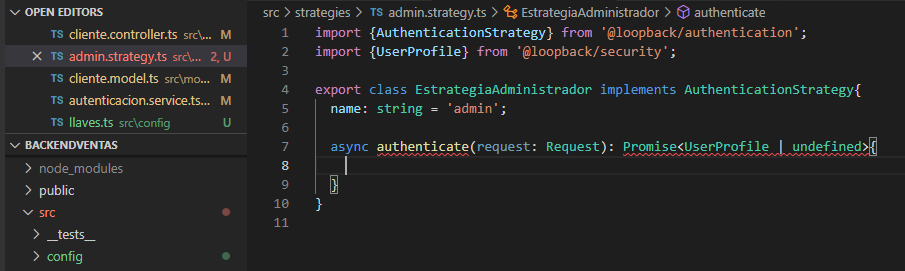
Parte video 4.2 – parte 2

Se instalan los paquetes

npm i --force @loopback/authentication

npm i --force @loopback/security

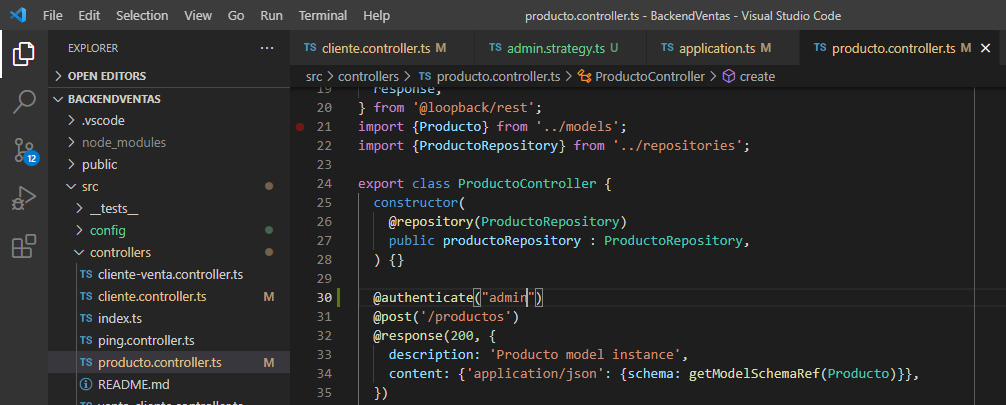
se crea una nueva carpeta



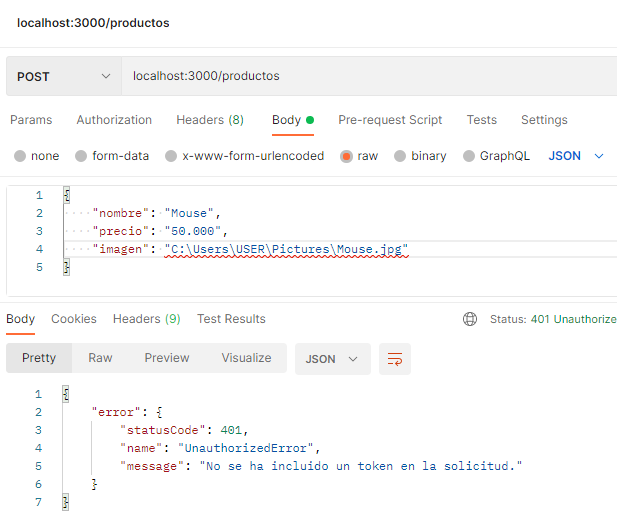
Se instalan nuevos paquetes

npm i parse-bearer-token

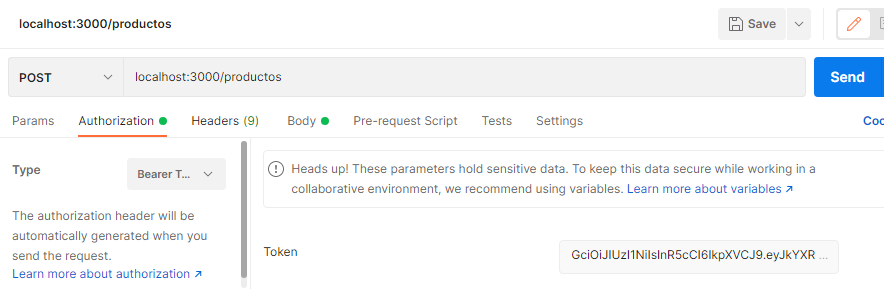
Se ingresa en productos la solicitud de autenticación.

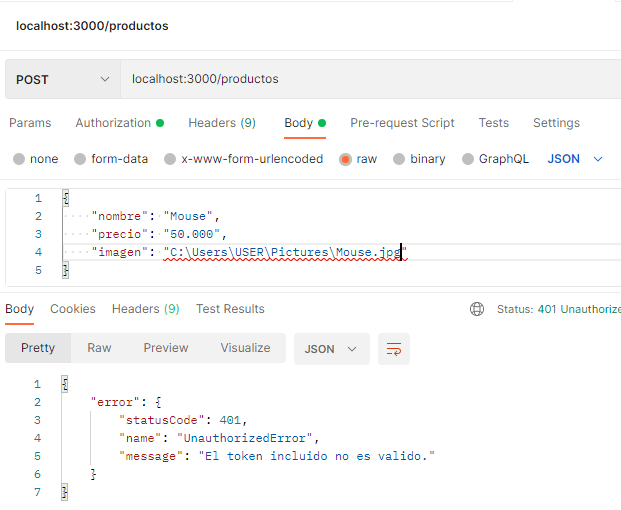


Se hace las pruebas en postman y como no se ingresó la solicitud de autenticación, el resultado es el error que se ingresó como respuesta.



Se ingresa el token y se le modifican el código y sale el error esperado.





Con el ingreso del token correcto debe mostrar que:



El producto fue creado – ejemplo del profesor (imagen en el servidor), pero la ruta de nuestra imagen no está en el servidor sino en el disco duro del equipo de cómputo por eso genera un error, y no crea el producto.

