El futuro digital es de todos

MinTIC

Formato de Informe de Seguimiento

Universidad de Caldas

Materia: Desarrollo de Aplicaciones Web – Ciclo 4A

Grupo: 12

Docente: Andrés Felipe Escallon Portilla.

Tutora: Yurley Katherine Echeverría Leal.

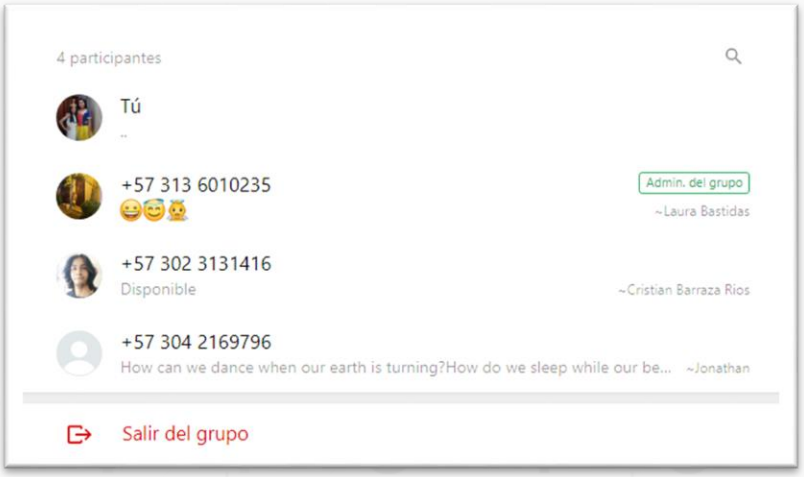
Sprint 3: Tienda Virtual de Tecnología.

Formato de Informe de Seguimiento

Equipo 2. Techno Team

Integrantes (Nombre completo)	Cédula	Rol	Nivel de participación (Alto, Medio, Bajo, Retirado)
BARRAGAN PLAZAS CARLOS EDUARDO	79536048	Líder de equipo. Administrador de Configuración.	Alto.
BARRAZA RIOS CRISTIAN	1045749373	Tester.	Alto.
BASTIDAS LAME LAURA MARCELA	1061774975	Diseñador UI.	Alto.
CAICEDO BELTRAN JONATHAN	1030579031	Diseñador de Software.	Alto.
CESPEDES RAMIREZ JOSE GIOVANNI	79854497	No se ha presentado.	No se ha presentado.

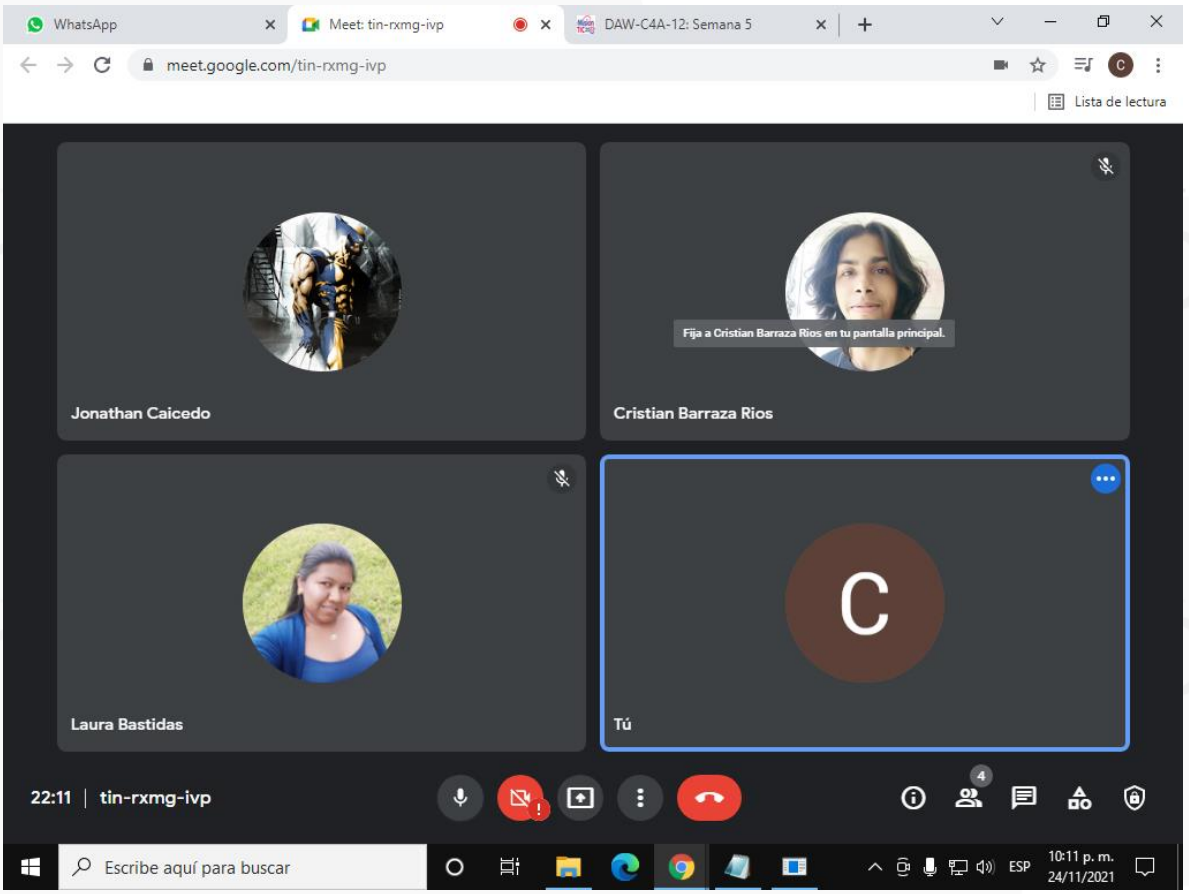
Se realiza la reunión con el equipo, nos estamos comunicando por WhatsApp; donde se creó el grupo:



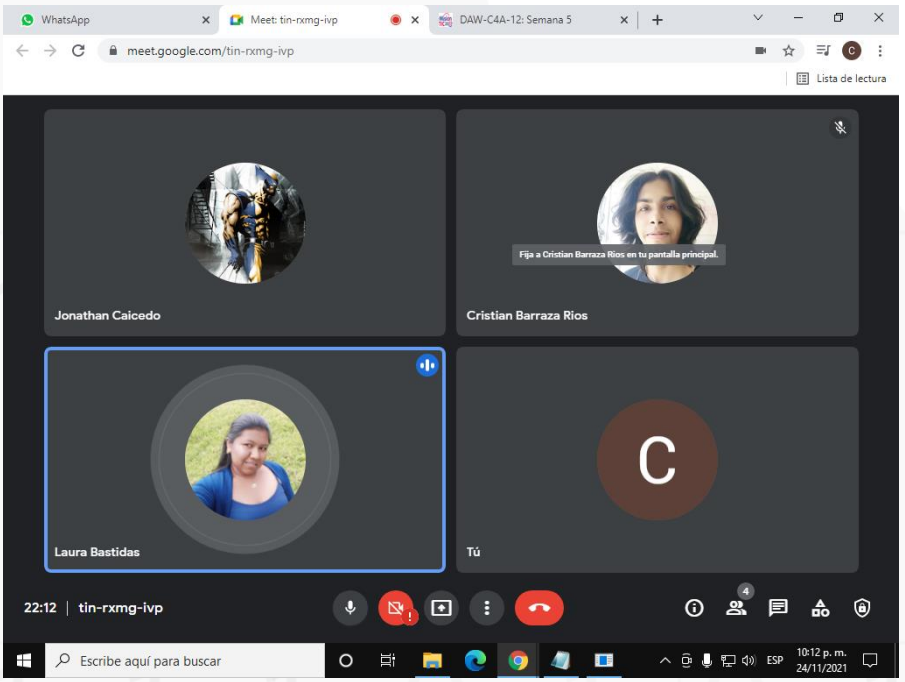
El señor José Giovanni Céspedes Ramírez, no se ha presentado.

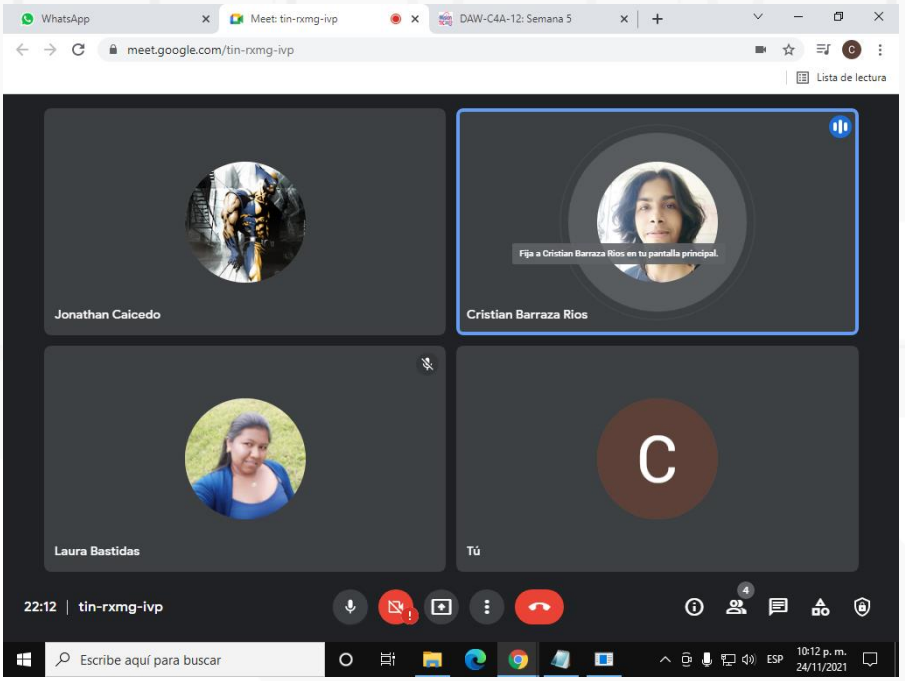
Se anexa imagen de la reunión por Meet.

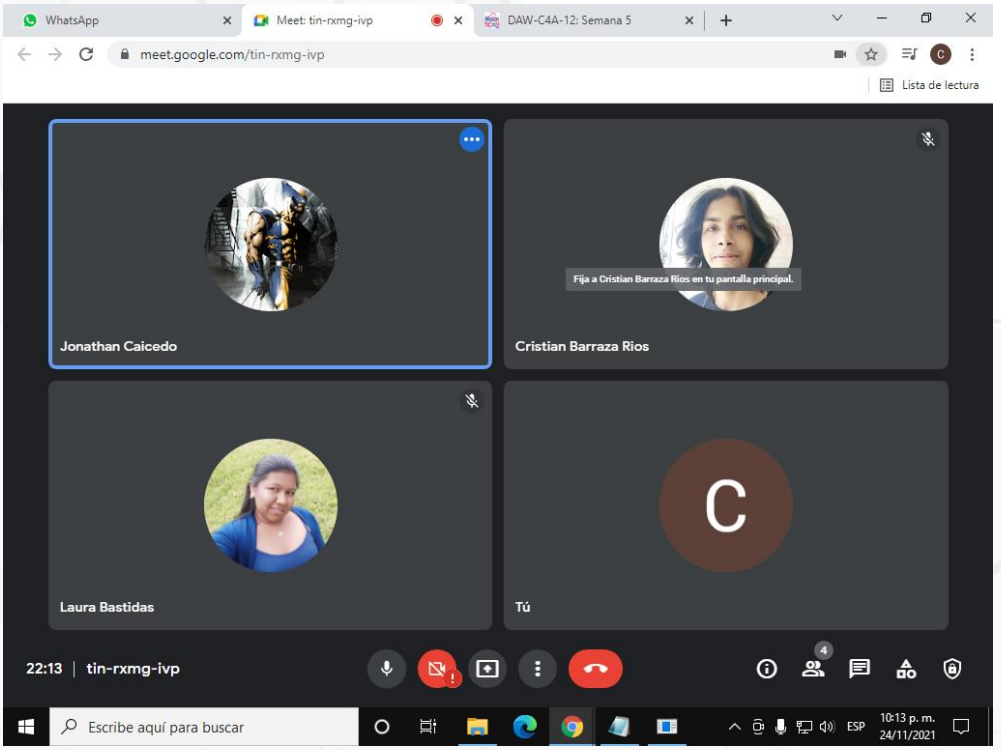
Se da inicio a la reunión previamente programada, donde hablamos sobre el trabajo entregado correspondiente al Sprint2 y empezamos a revisar el trabajo correspondiente al Sprint3.



En las imágenes que se anexan al trabajo podemos apreciar la participación de cada uno de los integrantes del grupo Techno Team, en cuanto refiere en la manera del desarrollo del trabajo que se va a presentar, como se va a recibir, verificar y organizar el trabajo.



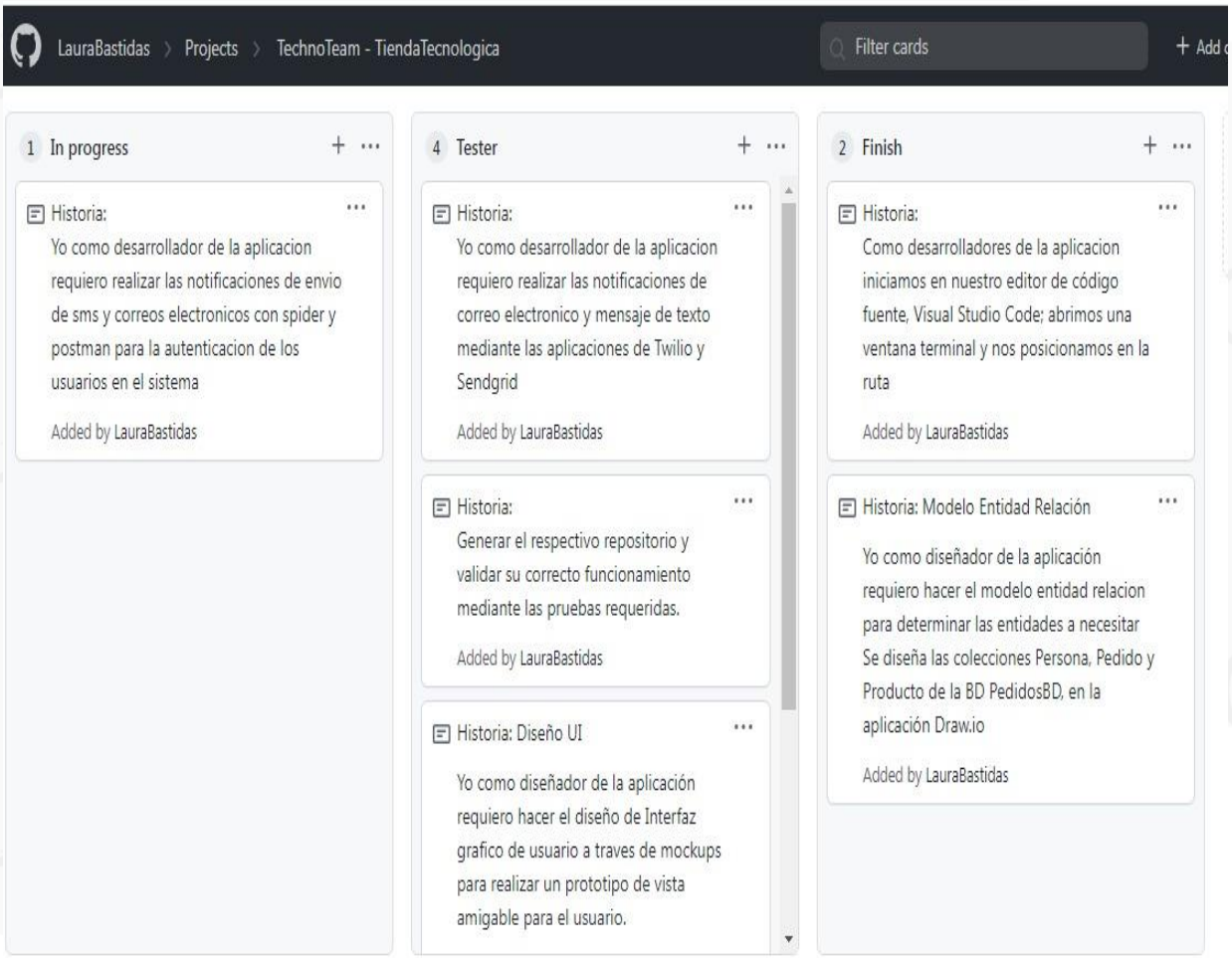



WhatsApp
Meet: tin-rxmg-ivp
DAW-C4A-12: Semana 5
meet.google.com/tin-rxmg-ivp
Lista de lectura
Jonathan Caicedo
Fija a Cristian Barraza Rios en tu pantalla principal.
Cristian Barraza Rios
Laura Bastidas
C
Tú
22:13 | tin-rxmg-ivp
Escribe aquí para buscar
ESP
10:13 p. m.
24/11/2021

Tablero Kanban.

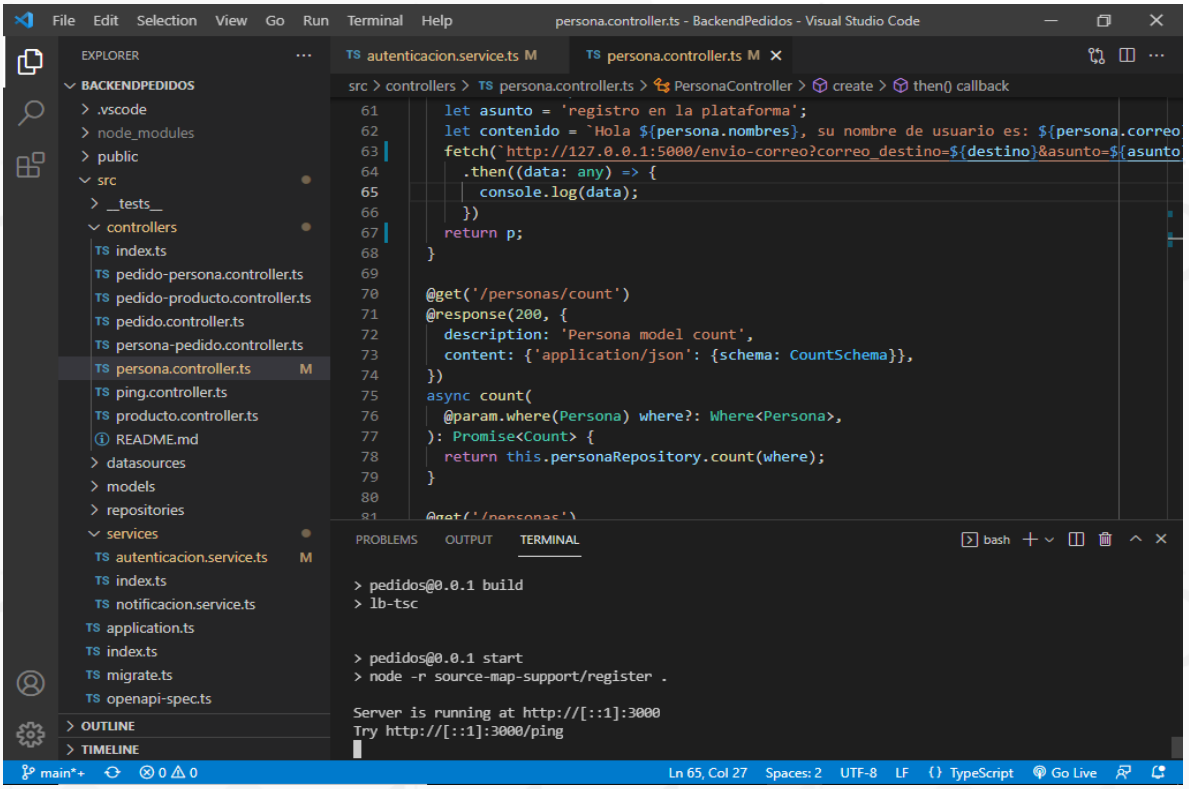
Actividades planteadas en el tablero Kanban, se actualiza el tablero Kanban con las actividades correspondientes al Sprint3, proyecto tienda virtual de tecnología.

En la siguiente imagen se puede observar cómo se visualiza el tablero:

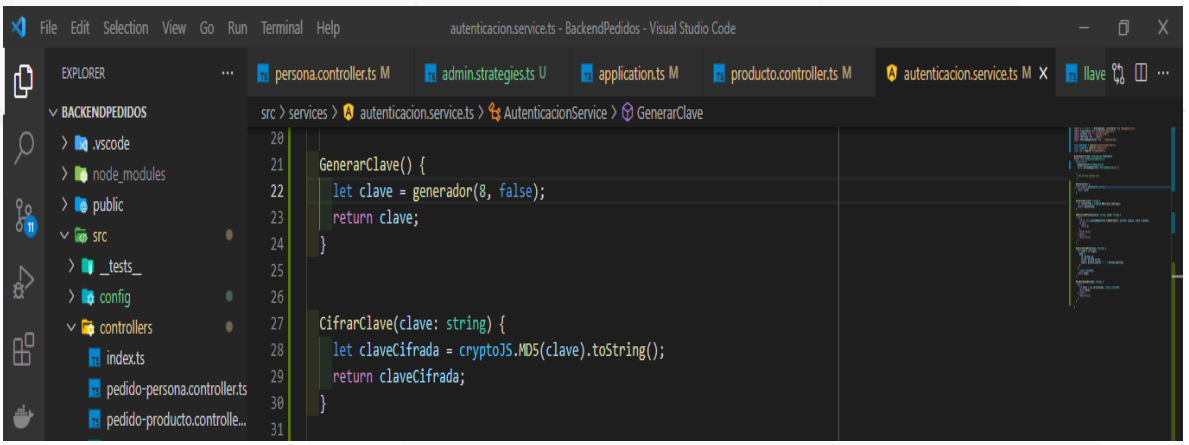


Se procederá a agregar funcionalidades de notificación en Loopback, utilizaremos el servicio de correo electrónico y mensajería de texto.

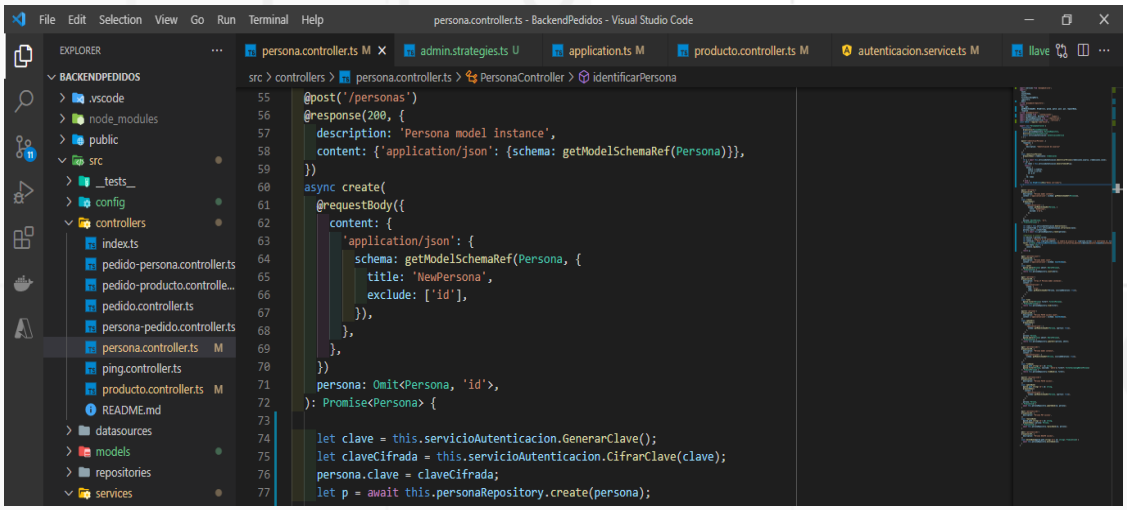
Se modifica el archivo autenticacion.services.ts



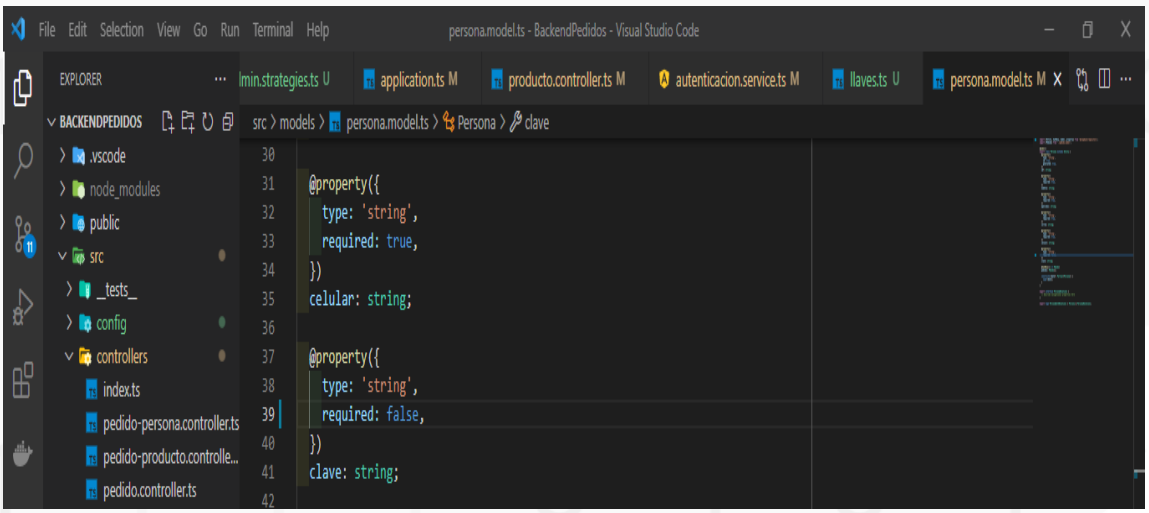
Se crean los métodos de generación y cifrado de contraseña



Se crean las variables para las contraseñas, en el controlador de personas.

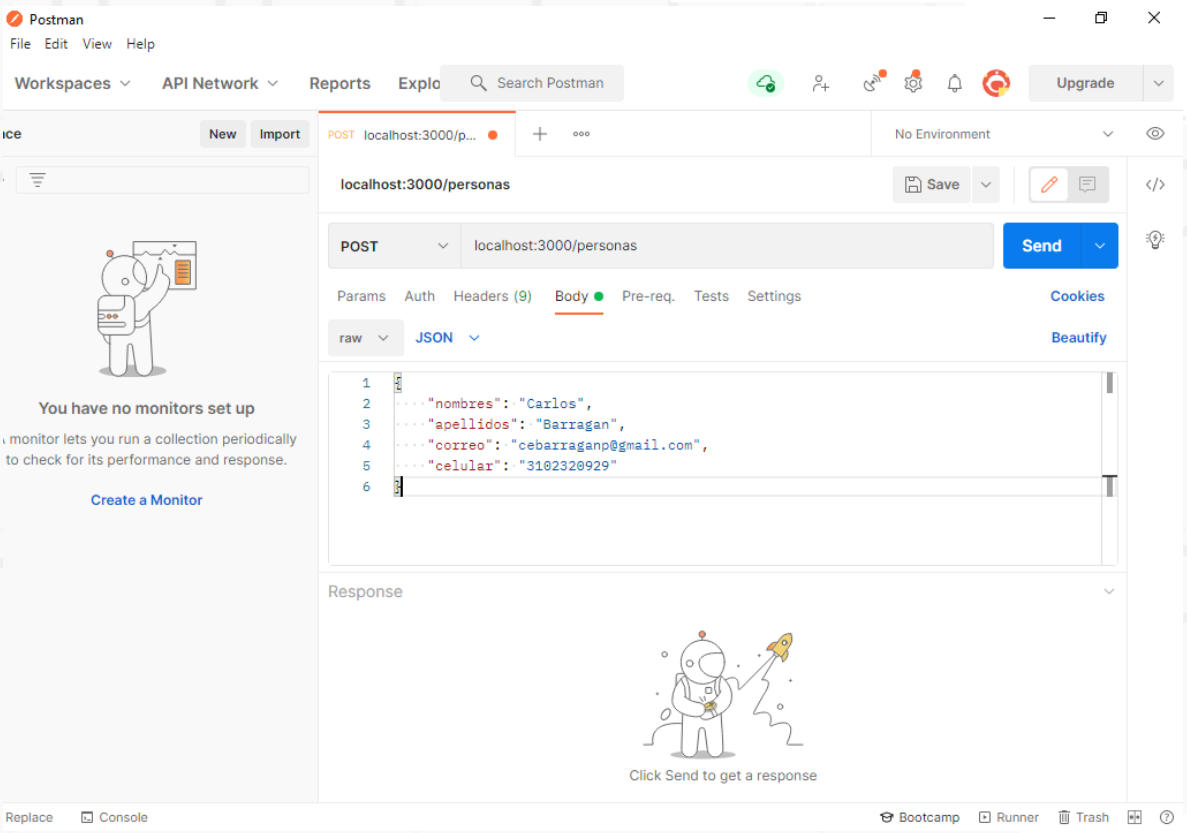


Se modifica el atributo clave para que el programa no la requiera

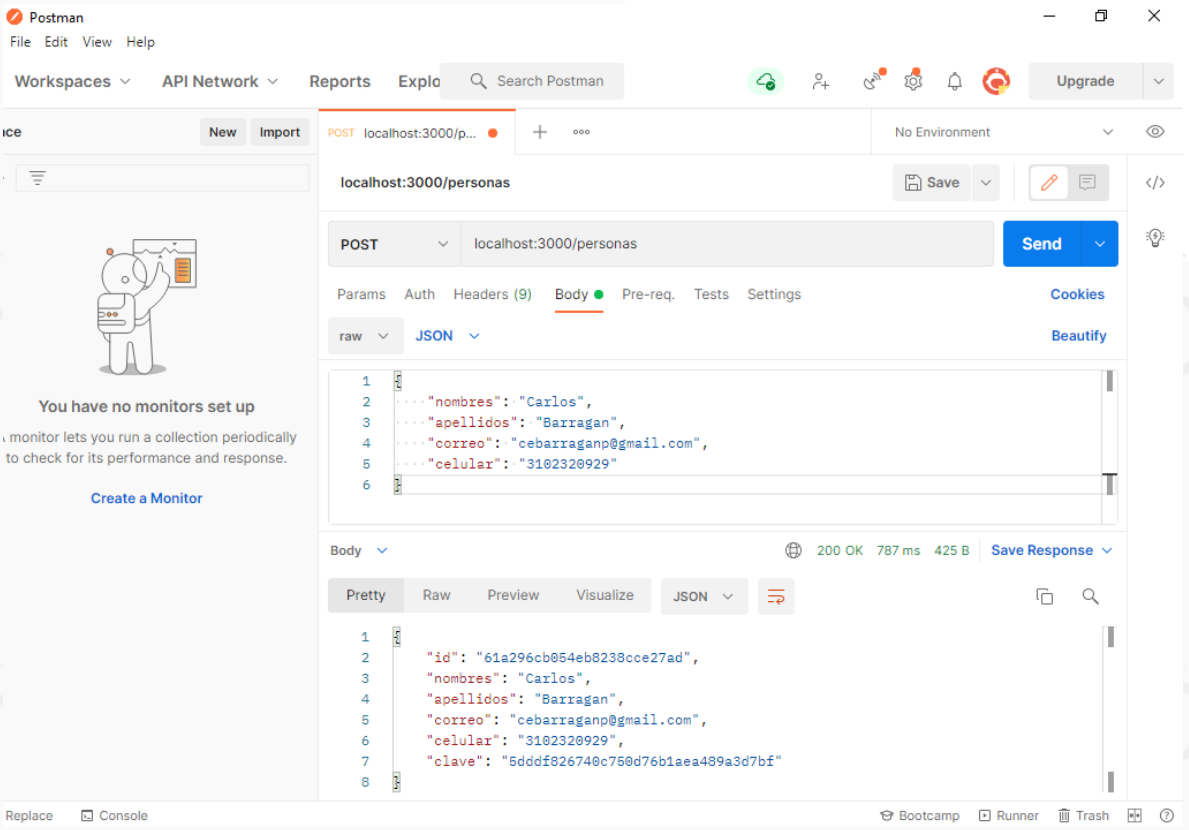


Se procede a realizar las pruebas de funcionamiento en el aplicativo POSTMAN para verificar que las claves se están generando de manera correcta.

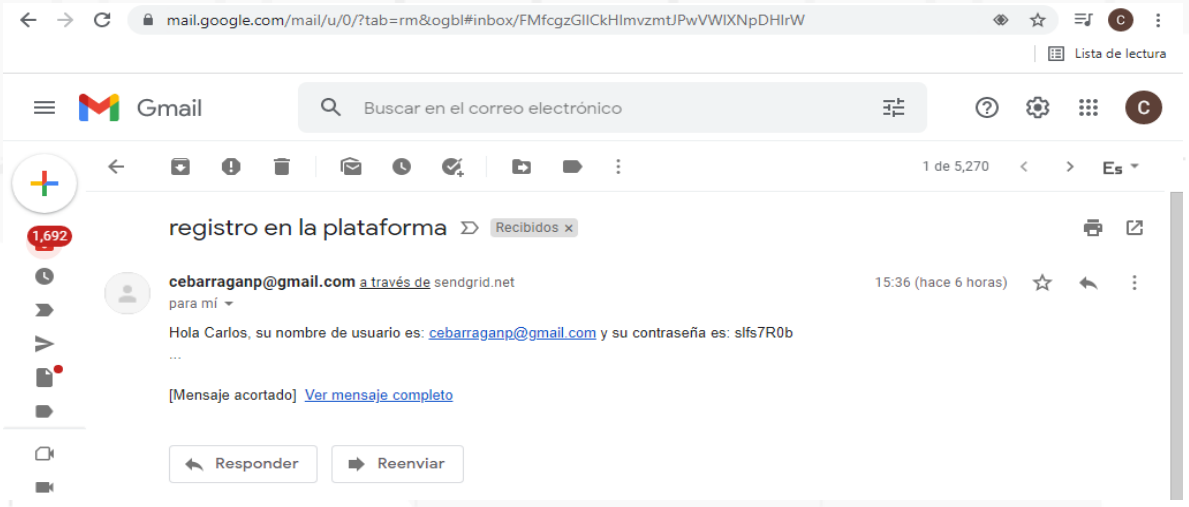
Una vez que se inicia el aplicativo procedemos a diligenciar la información correspondiente a la prueba de notificación, como sabemos una vez que iniciamos el programa en loopback este se ejecuta en la URL localhost:3000/personas



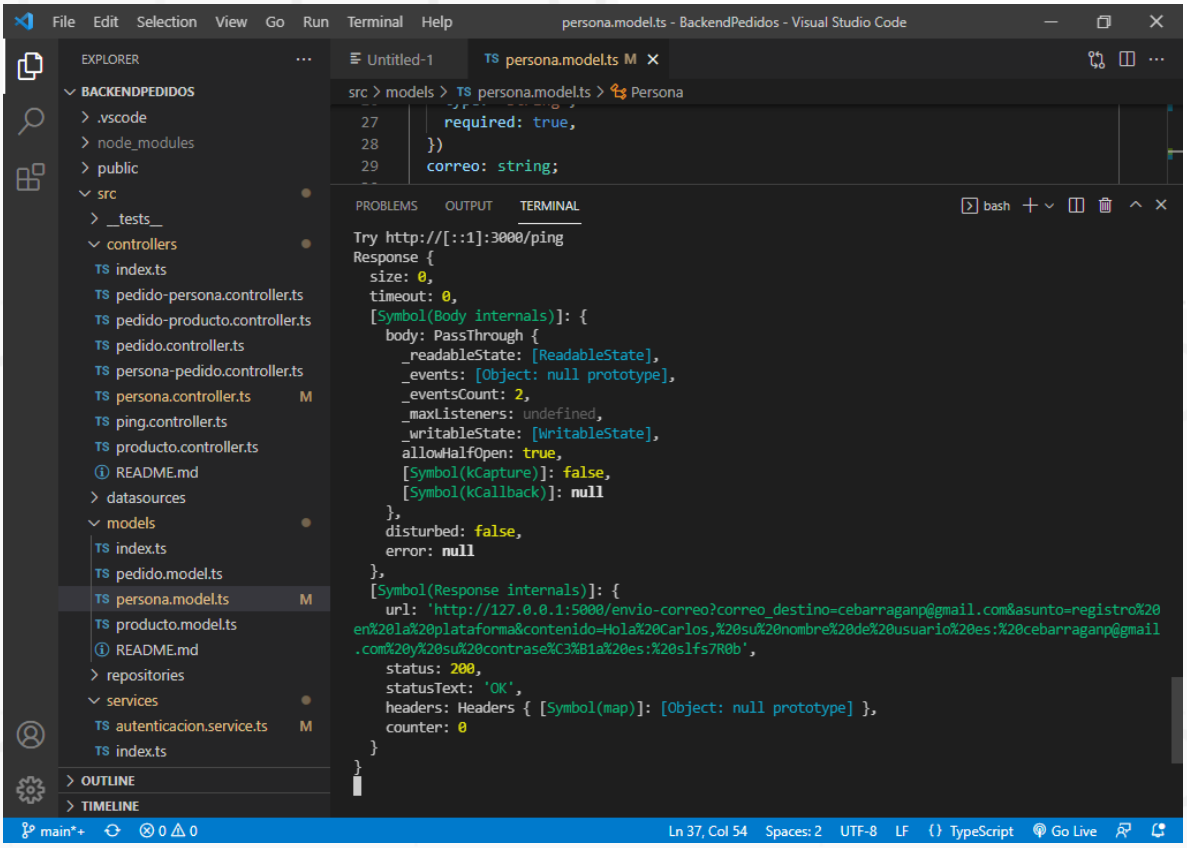
Como se puede apreciar en la imagen generada por el sistema las pruebas de funcionamiento fueron exitosas:



Nos dirigimos al correo electrónico para verificar que se recibe la notificación:

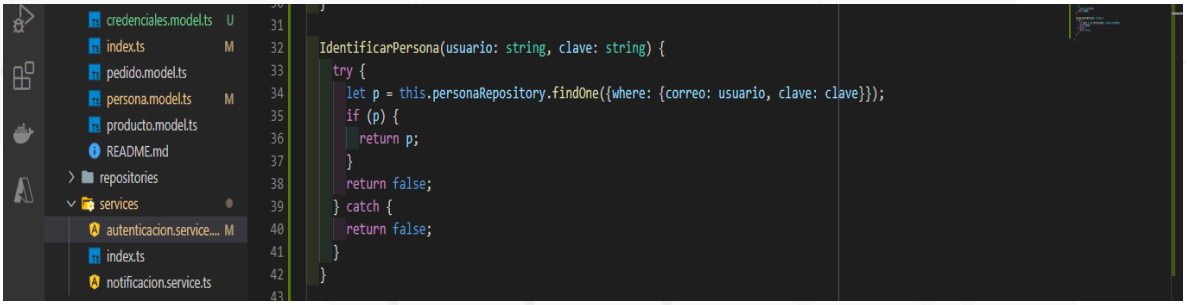


Podemos verificar el mensaje en la aplicación Loopback.

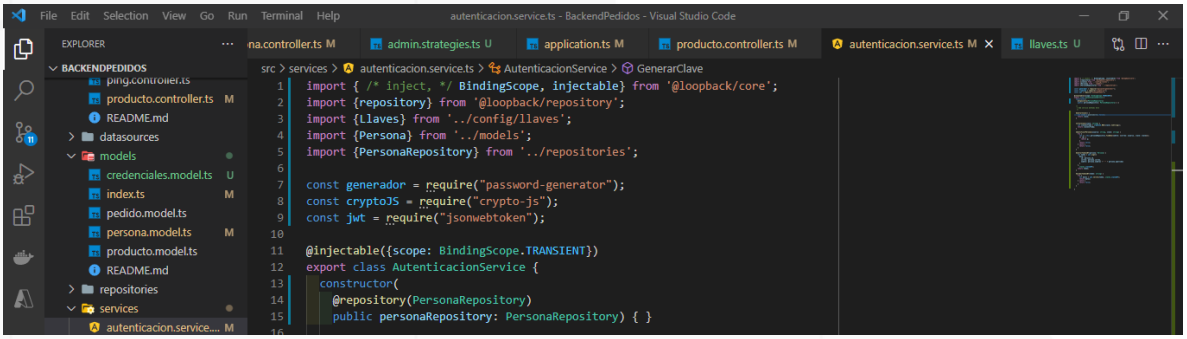


Estrategia Autenticación con JWT

En el archivo autenticacion.services.ts se crea el método para identificar una persona



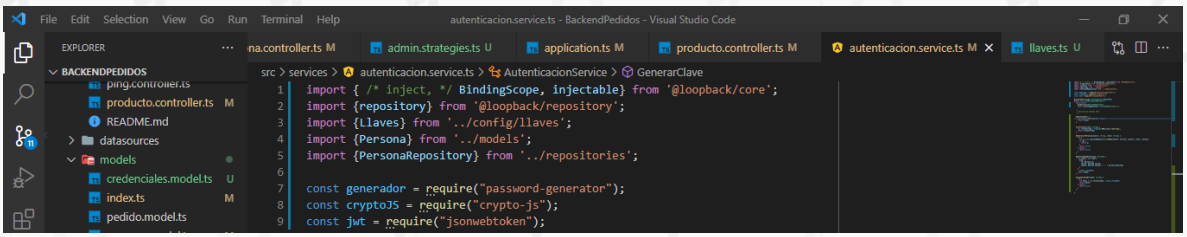
Y se inyecta en el constructor, repositorio de persona.



Se crea el método para generar el token

Se instala npm i jsonwebtoken

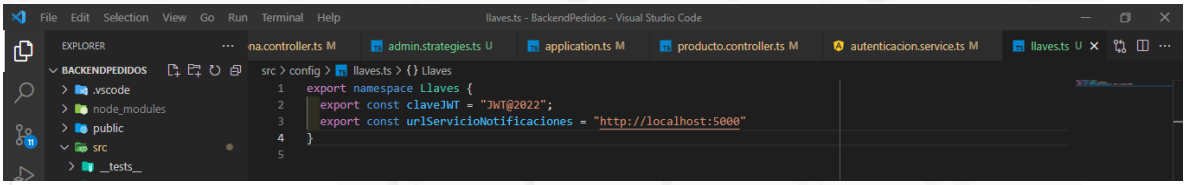
se agrega una constante; la constante const jwt = require("jsonwebtoken");



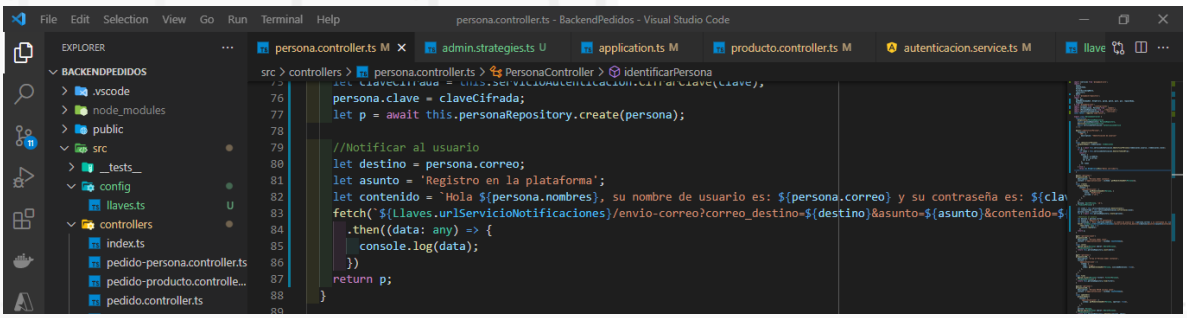
Debemos generar o crear una carpeta y un archivo.

dentro de src crear carpeta con el nombre config

dentro de la carpeta config crear el archivo con el nombre llaves.ts

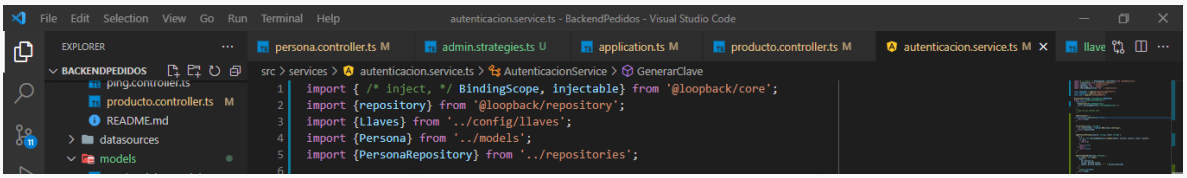


Modificar la línea fetch del controlador de persona luego del método POST

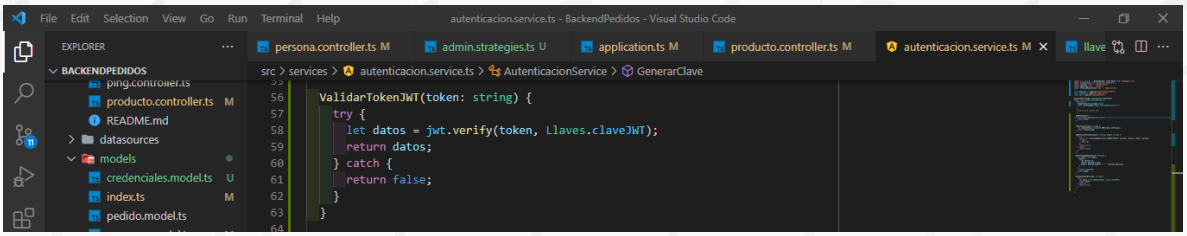


importar archivo llaves

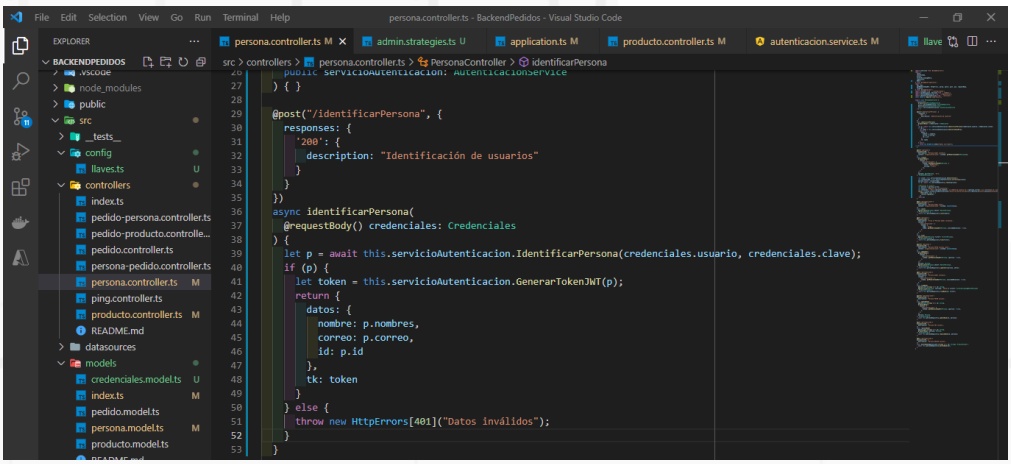
import {Llaves} from '../config/llaves';



Se crea el método para validar token



En el controlador de persona se crea el método POST para identificar personas



Se crea un modelo para las Credenciales

lb4 model

nombre-> Credenciales

tipo-> model

prop. Adic? -> no

prop -> usuario

tipo -> string

es id? -> no

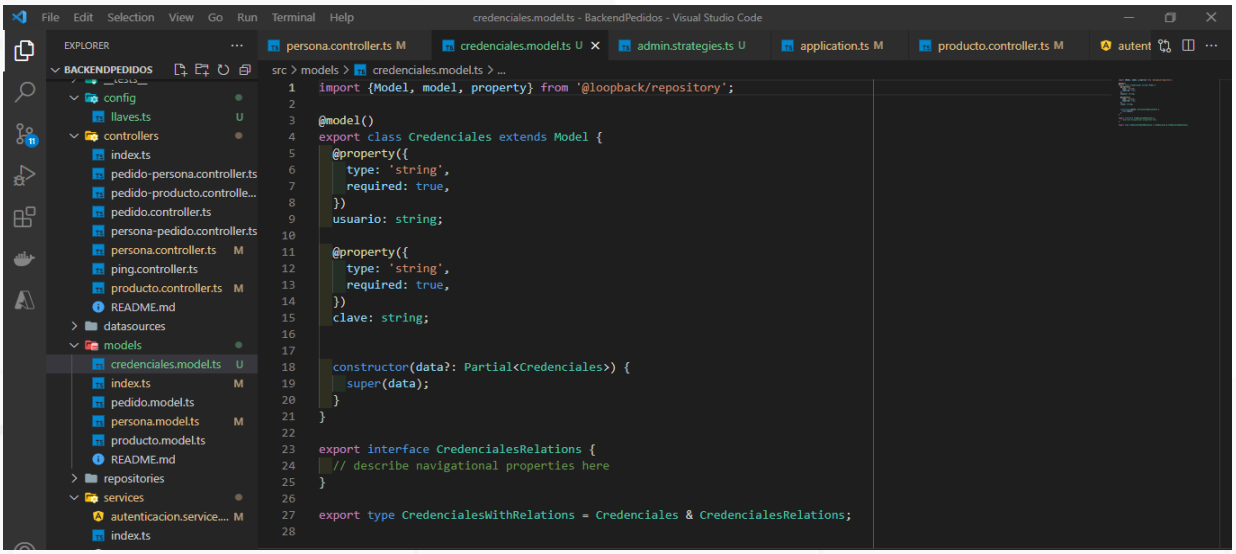
es necesario? -> si

prop -> clave

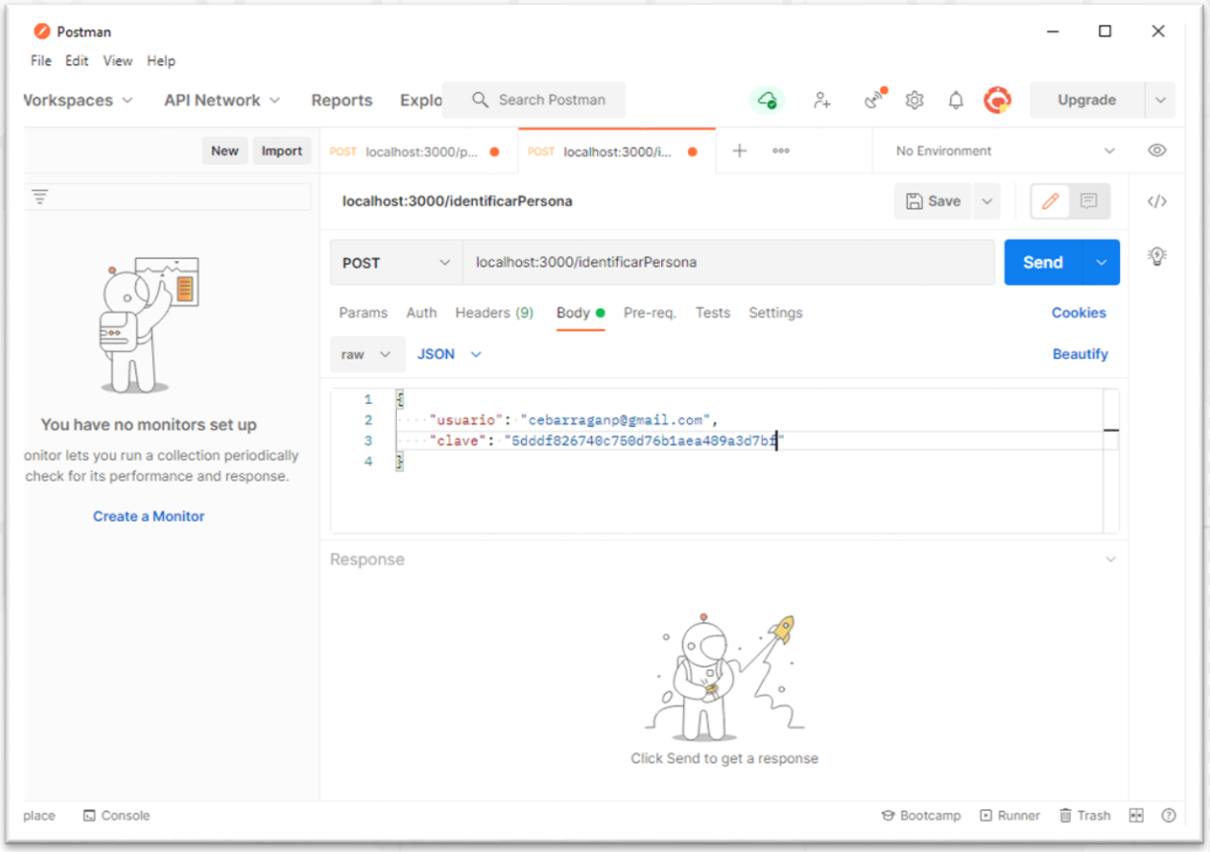
tipo -> string

es id? -> no

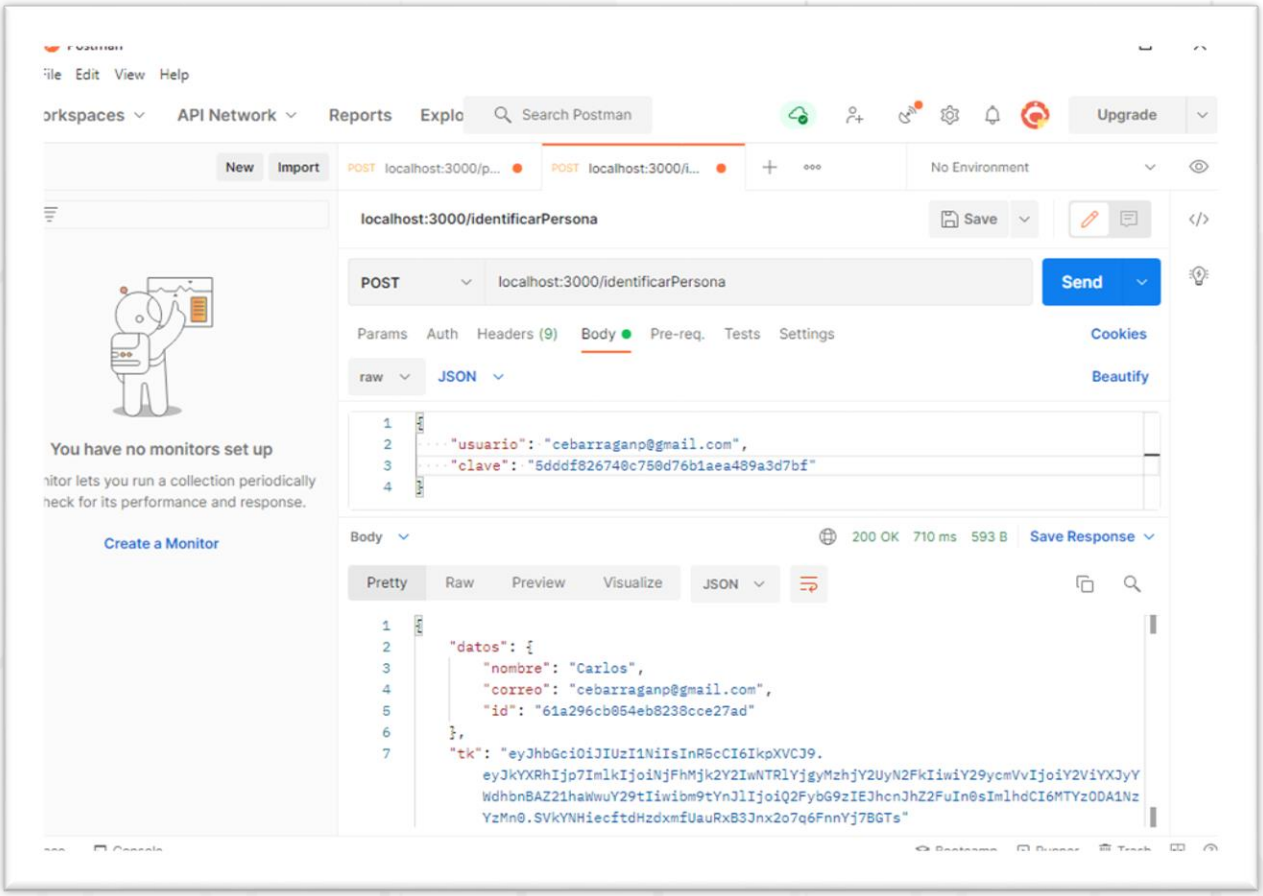
es necesario? -> si



Se ejecuta y se hacen las pruebas en el aplicativo POSTMAN. Donde se utilizan los datos que se recibieron por correo.



El token se genera correctamente:



SEGUNDA PARTE

Se procede a instalar los paquetes en la consola de git

npm i --force @loopback/authentication

npm i --force @loopback/security

El paquete correspondiente a loopback/authentication, provee todo lo que concierne a la captura

de las solicitudes y la validación de las mismas.

El paquete correspondiente loopback/security, provee todo lo que concierne y permite el acceso al user profile, que corresponde al perfil de la cuenta del usuario.

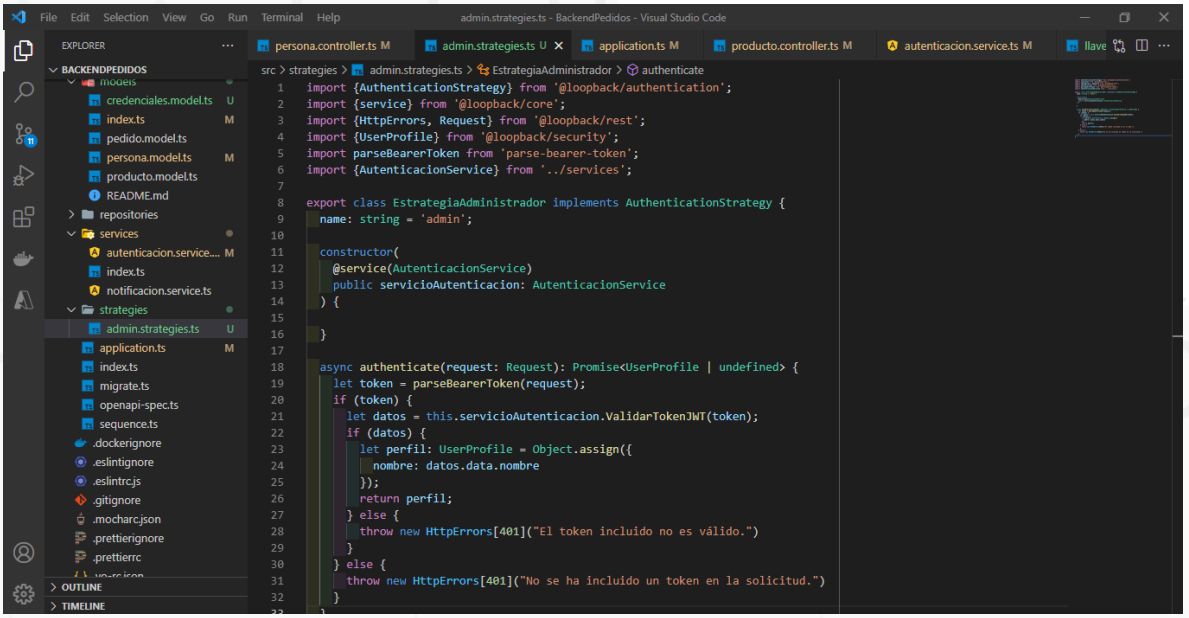
crear carpeta y archivo

dentro de src crear una nueva carpeta con el nombre strategies

dentro de strategies crear un nuevo archivo con el nombre admin.strategies.ts

El proceso completo del JWT consta de dos pasos:

- El usuario de una aplicacion web/móvil/desktop hace login con sus credenciales en el servidor donde esta publicada el API.
- El usuario es validado en el servidor y se crea un nuevo Token JWT (usando nuestro "secret-key") para entregárselo al usuario.



instalar
npm i parse-bearer-token

En el archivo aplication.ts

En el archivo producto.controller.ts

Se coloca antes del metodo POST

@authenticate("admin") para proteger todos los métodos con autenticación, al que no se quiera proteger se le coloca @authenticate.skip()