

Sphere Pro

Aplicación de Gestión de Usuarios, Productos y Facturas

Aplicación web que hace gestión de usuarios, productos y facturas con base en tecnologías modernas

marzo de 2025

Michael Sebastian Barreto Tellez



Índice general

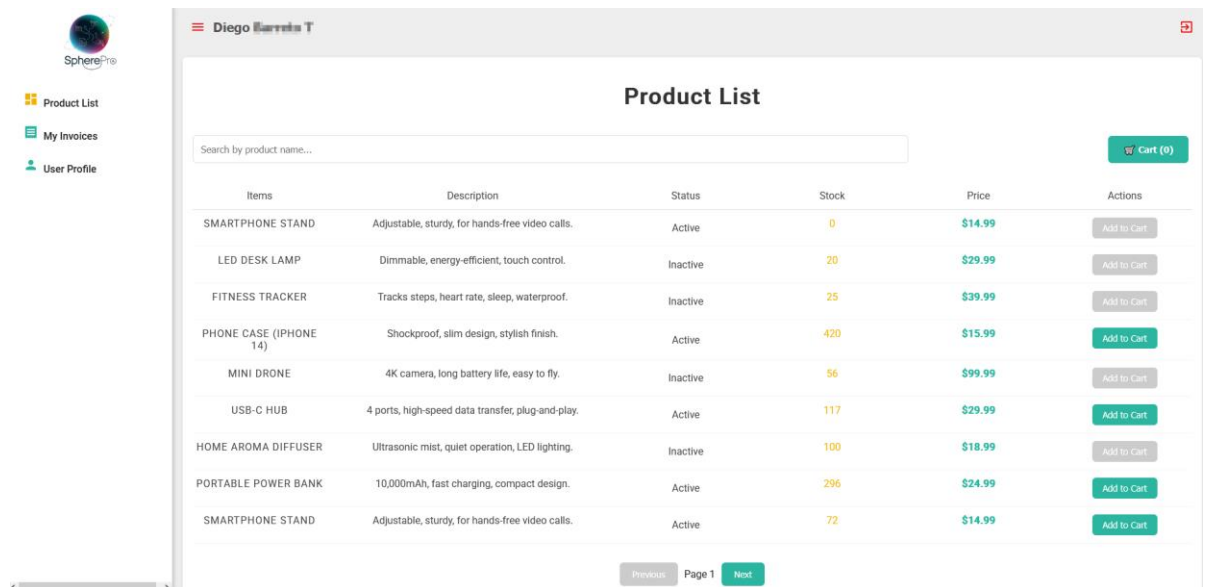
1-Presentación del proyecto	3
2-Contexto del proyecto	4
3-Funcionalidades claves	5
4-Detalles técnicos esenciales	6
5-Arquitectura y diseño	7
6-Desafío y soluciones	8
7-Resultados y demostraciones	9
8-Aprendizaje y mejoras futuras	14

Presentación del proyecto

Se creó una aplicación la cual nos permite gestionar usuarios, productos y facturas, con diferentes permisos gestionados por un rol en este caso (User, Admin), el cual por medio de un token se efectúa la validación del rol y sus permisos respectivos. Validaciones en tiempo real de (stock) en productos y un diseño intuitivo para mejorar la experiencia del usuario.

Logros clave en este proyecto:

- ✓ Implementación exitosa de autenticación JWT y permiso por roles
- ✓ Diseño UI/UX moderno y accesible
- ✓ Despliegue en AWS



Contexto del proyecto

Propósito:

El proyecto se a desarrollado con la intención de simplificar la gestión de ventas y usuarios de pequeñas empresas, garantizando seguridad y escalabilidad

Tecnologías claves:

- **Backend:** NestJS (REST API robusta y modular cumpliendo todas las funciones CRUD)
- **Frontend:** Angular (Interfaz fluida y reactiva)
- **Base de datos:** MongoDB Atlas (Almacenamiento flexible y en la nube)

```
[Nest] 19200 - 11/03/2025, 11:05:25 a. m. LOG [NestFactory] Starting Nest application...
[Nest] 19200 - 11/03/2025, 11:05:25 a. m. LOG [InstanceLoader] AppModule dependencies initialized +25ms
[Nest] 19200 - 11/03/2025, 11:05:25 a. m. LOG [InstanceLoader] MongooseModule dependencies initialized +0ms
[Nest] 19200 - 11/03/2025, 11:05:25 a. m. LOG [InstanceLoader] ConfigHostModule dependencies initialized +1ms
[Nest] 19200 - 11/03/2025, 11:05:25 a. m. LOG [InstanceLoader] ConfigModule dependencies initialized +0ms
[Nest] 19200 - 11/03/2025, 11:05:25 a. m. LOG [InstanceLoader] JwtModule dependencies initialized +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [InstanceLoader] MongooseCoreModule dependencies initialized +1118ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [InstanceLoader] MongooseModule dependencies initialized +6ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [InstanceLoader] MongooseModule dependencies initialized +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [InstanceLoader] AuthModule dependencies initialized +2ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [InstanceLoader] UsersModule dependencies initialized +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [InstanceLoader] ProductsModule dependencies initialized +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [InstanceLoader] InvoicesModule dependencies initialized +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RoutesResolver] UsersController {/users}: +4ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/users, GET} route +2ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/users/:id, GET} route +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/users/:id, PUT} route +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/users/:id, DELETE} route +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RoutesResolver] AuthController {/auth}: +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/auth/login, POST} route +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/auth/register, POST} route +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RoutesResolver] ProductsController {/products}: +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/products, GET} route +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/products, POST} route +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/products/:id, PUT} route +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/products/:id, DELETE} route +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RoutesResolver] InvoicesController {/invoices}: +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/invoices, POST} route +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/invoices, GET} route +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/invoices/:id, GET} route +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/invoices/user/:userId, GET} route +0ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [RouterExplorer] Mapped {/invoices/user/:userId/analytic, GET} route +1ms
[Nest] 19200 - 11/03/2025, 11:05:26 a. m. LOG [NestApplication] Nest application successfully started +2ms
```

Funcionalidades Clave

Usuarios:

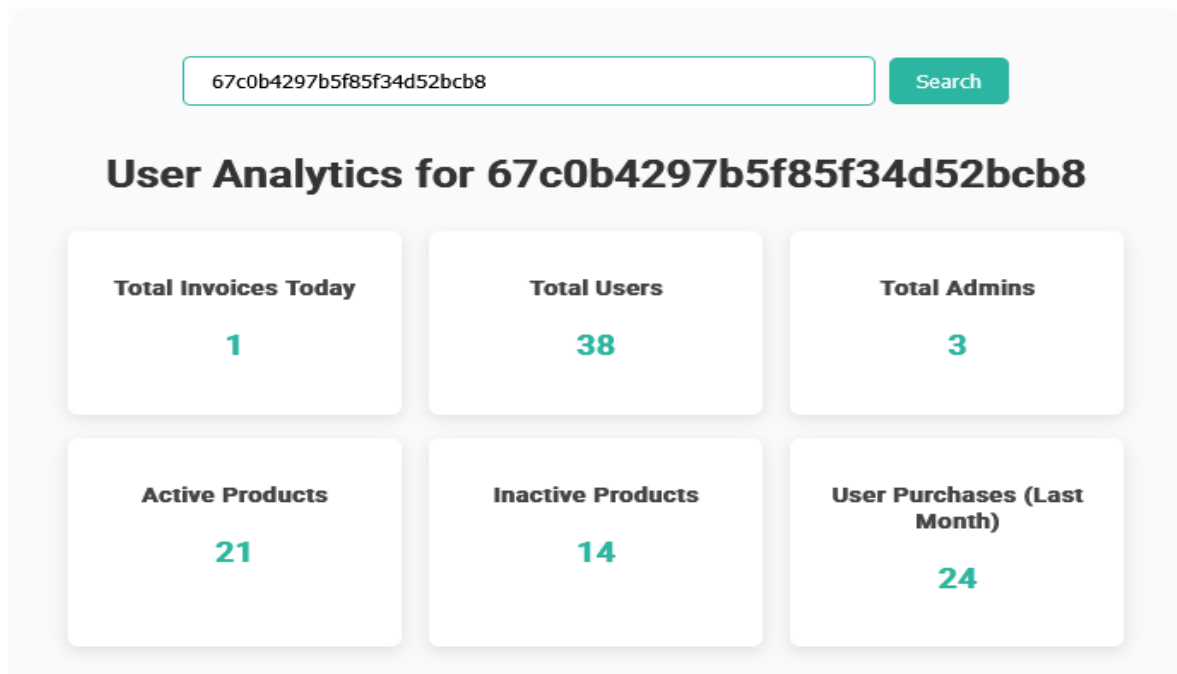
Los administradores pueden gestionar cualquier cuenta de los usuarios registrados, mientras los usuarios pueden editar su perfil de forma segura

Productos:

Los productos tienen stock controlado, si un usuario intenta comprar sin stock en el inventario o un Producto en modo “Inactive” el sistema no lo dejara agregar al carrito y el botón tendrá un color gris haciendo referencia a que el producto esta inhabilitado.

Facturas:

Los usuarios son los únicos que pueden generar facturas, estas facturas salen a nombre propio y con los productos agregados al carrito, los administradores pueden ver todos los detalles incluyendo el ID del usuario que genero la factura y ver cuantas facturas a generado el usuario desde un principio y un contador por las factuas generadas en el dia y el mes



Detalles técnicos esenciales

Seguridad:

Usamos tokens JWT para proteger el acceso a diferentes funcionalidades y vistas según su rol, cada rol cuenta con permisos específicos, como por ejemplo el administrador puede ver todos los usuarios registrados.

Validaciones:

Ningún producto puede quedar en stock negativo, el sistema lo evita incluso si hay errores humanos

Consulta analítica:

Contamos con un Endpoint para saber cuantas compras hizo un usuario en el día y mes útil para análisis comerciales.

Search by

ID

Search term

Search

Role

Select role

Apply

ID	Full Name	E-mail	Administrator	Actions
67bcf83ea9889bef5db31161	sebas barreto T	shebasbt@gmail.com	✓	Edit
67be353901dcc17e064332ef	micahel Barreto	sebas@gmail.com	✓	Edit
67be4e4c16723f48befc50b2	miley barreto t	miley@gmail.com		Edit
67c0b4297b5f85f34d52bcb8	diego barreto T	diegob@gmail.com		Edit
67cb433b2ce783924c4af75f	juan camilo buitrago	juanb@gmail.com	✓	Edit
67cc79308a1e0efe6af24889	Andres Lopes	andresl@example.com		Edit
67cc79768a1e0efe6af248ec	Juan perez	juanperez@example.com		Edit

Arquitectura y diseño

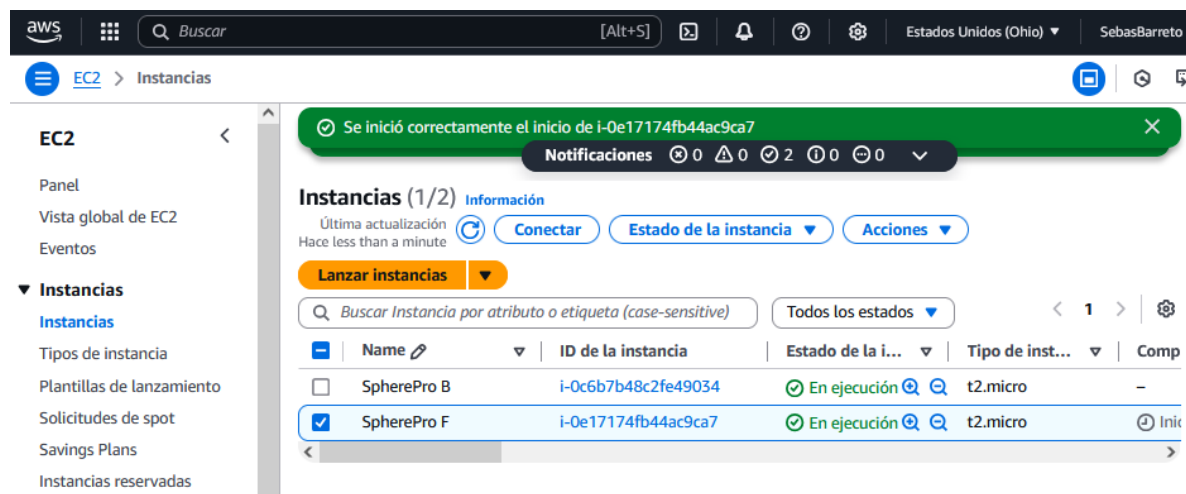
Diagrama simplificado:

Frontend en Angular -> API en NestJS -> MongoDB Atlas -> Despliegue en AWS (EC2)

Principios aplicados:

Código modular (SOLID) para facilitar mantenimiento y pruebas

Diseño UI/UX enfocado en reducir clics, formularios limpios, navegación intuitiva



Desafíos y soluciones

Implementar JWT con NestJS fue un reto, tuve un error por mas de 1 dia, ya que en mi código estaba el “Token Access” y en una parte tenia el “Token Acces” este pequeño detalle de la falta de una “s” no me dejaba avanzar, hasta que tuve que revisar código por código para dar con la solución, con la ayuda de Guards personalizados logramos un flujo seguro en la aplicación

La sincronización del stock, el carrito, y las facturas. Fue algo que tuve que estar al 100% para no tener errores y que todo estuviera sincronizado. Una vez terminado probe con diferentes usuarios y verificación en mongo para no tener inconsistencia alguna.

All Invoices

Attention! Only the latest 50 invoices are being displayed.

Invoice Code				View
INV-024573				View
INV-032130				View
INV-033761				View
INV-062639				View
INV-071026				View
INV-103947				View
INV-132376	\$79.95	3/6/25, 11:16 PM		View
INV-153131	\$172.92	3/8/25, 12:12 PM		View
INV-157734	\$175.89	3/9/25, 12:55 PM		View

\$164.94 Paid

Invoices - Sphere Pro

Invoice Code: INV-024573

Date: 3/6/25, 2:42 PM

Buyer User ID: 67c0b4297b5f85f34d52bcb8

Products:

Smartphone Stand	3	\$14.99	\$44.97
Fitness Tracker	3	\$39.99	\$119.97
Total			\$164.94

Delete Invoice

- **Copyright (c) 2025 Sebas Barreto**
- All rights reserved.
- This code is the property of <https://github.com/SebasBarreto/InvoiceSphere-Pro>.
- No part of this code may be reproduced or used without the express permission of the copyright holder.
- This code is licensed under the MIT License, which can be found at <https://opensource.org/licenses/MIT>.

Resultados y demostraciones

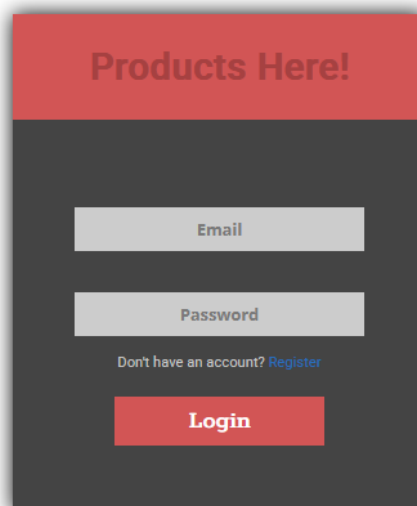
¡E subido mi proyecto a AWS (EC2) el cual estará disponible por un tiempo no mayor a 15 días para su demostración! Para que no genere un cobro adicional. Para poder verificar su funcionalidad puedes ingresar a la siguiente dirección el cual va hacer nuestro Frontend, este ya estará configurado con la instancia del Backend y será 100% funcional

Usuario Admin

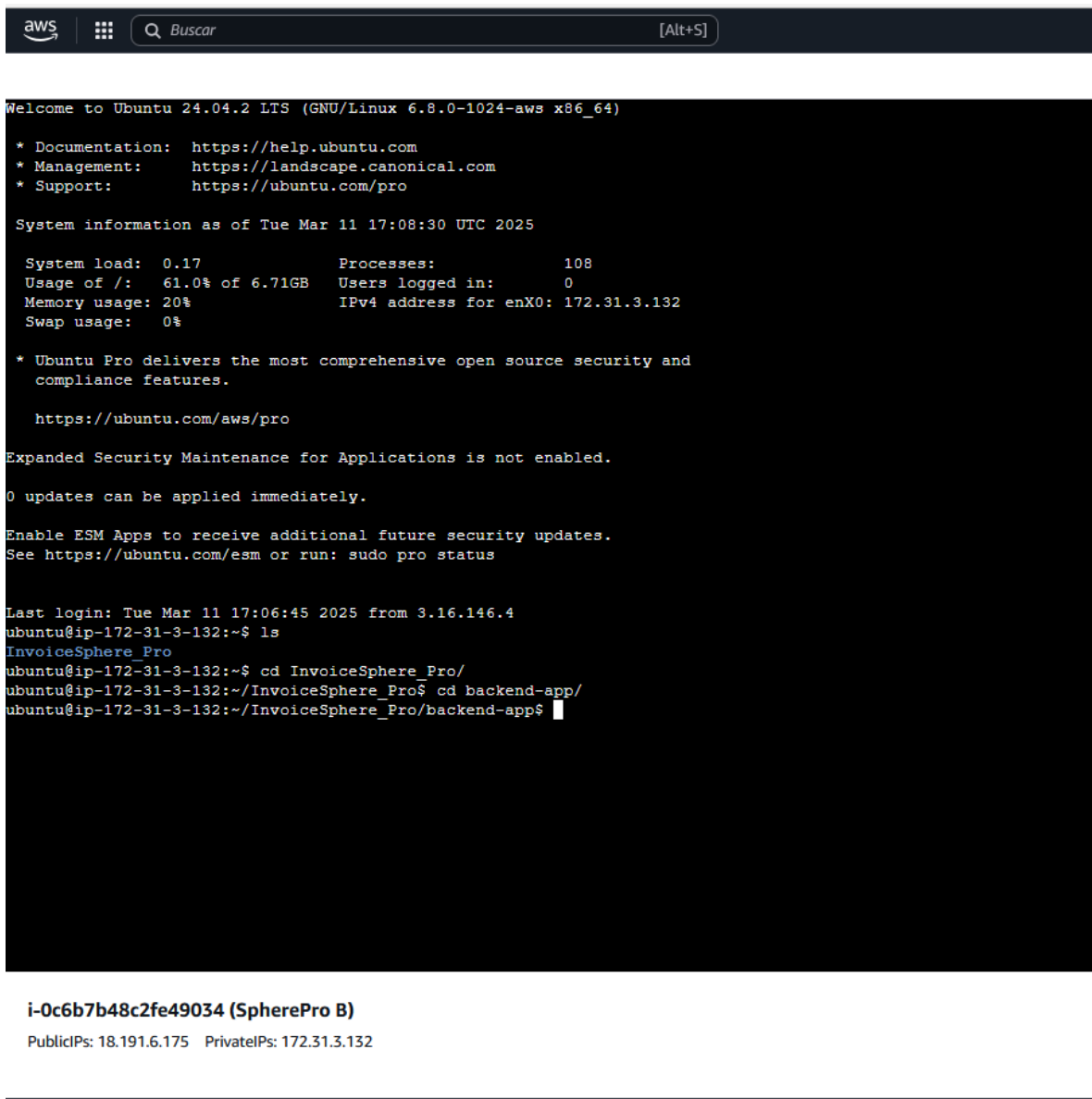
E-mail: ["shebasbt@gmail.com"](mailto:shebasbt@gmail.com)

Password: "Prueba2025"

Fecha de instancia: "23 marzo de 2025" / <http://3.141.202.8:4200>



A continuación dejo capturas de como poder configurar la aplicación en las instancia (Ubuntu) en AWS, una vez tengamos el Backend en una instancia y el Frontend en la otra, tenemos que conectarnos a la instancia del Backend, en la parte de abajo vemos la IP publica, navegamos al directorio de nuestro proyecto para una vez ejecutar el comando (`npm run start`) y ejecutar nuestro Backend.



```
aws [icon] [icon] Q Buscar [Alt+S]

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

System information as of Tue Mar 11 17:08:30 UTC 2025

System load:  0.17           Processes:            108
Usage of /:   61.0% of 6.71GB Users logged in:      0
Memory usage: 20%           IPv4 address for enX0: 172.31.3.132
Swap usage:   0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Mar 11 17:06:45 2025 from 3.16.146.4
ubuntu@ip-172-31-3-132:~$ ls
InvoiceSphere_Pro
ubuntu@ip-172-31-3-132:~$ cd InvoiceSphere_Pro/
ubuntu@ip-172-31-3-132:~/InvoiceSphere_Pro$ cd backend-app/
ubuntu@ip-172-31-3-132:~/InvoiceSphere_Pro/backend-app$
```

i-0c6b7b48c2fe49034 (SpherePro B)
PublicIPs: 18.191.6.175 PrivateIPs: 172.31.3.132

Una vez ejecutado nuestro Backend vamos a nuestra segunda instancia para poder modificar el archivo “enviroment” el cual va a tener nuestra IP de nuestra Api (anteriormente copiado a nuestro portapapeles) y así poder hacer peticiones al Backend. Navegamos por la carpeta `./src/enviroment/` y allí encontraremos el archivo y con nano podemos ver su contenido y modificarlo.

```
aws [Alt+S] Q Buscar

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Mar 11 02:09:02 UTC 2025

System load: 0.0          Processes: 104
Usage of /:  53.7% of 6.71GB   Users logged in: 0
Memory usage: 21%          IPv4 address for enX0: 172.31.4.22
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

132 updates can be applied immediately.
41 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Mar 11 02:09:03 2025 from 3.16.146.5
ubuntu@ip-172-31-4-22:~$ ls
74cK-14854.pem  README.md  angular.json  node_modules  package-lock.json  package.json  src  tsconfig.app.json  tsconfig.json  tsconfig.spec.json
ubuntu@ip-172-31-4-22:~$ cd src/environments
ubuntu@ip-172-31-4-22:~/src/environments$ nano enviroment.ts
```

Una vez dentro del archivo pegamos nuestra ip publica en donde esta subrayado y teniendo en cuenta en dejar nuestro puerto “:4200” en escucha y guardamos para proceder a ejecutar nuestro servidor Frontend con el comando (ng serve --host 0.0.0.0)

```
export const environment = {  
  production: false,  
  apiUrl: 'http://18.191.6.175:3000'  
};
```

VS Code interface showing the file `environment.ts` with the following content:

```
export const environment = {  
  production: false,  
  apiUrl: 'http://18.191.6.175:3000'  
};
```

The status bar at the bottom indicates the instance is `i-0e17174fb44ac9ca7 (SpherePro F)` with `PublicIPs: 3.141.202.8` and `PrivateIPs: 172.31.4.22`.

Aprendizaje y mejoras futuras

- ❖ Aprendi a optimizar consultas en MongoDB para manejar miles de facturas eficientemente. En el futuro podemos agregar
- ❖ Docker para optimizar el rendimiento de las instancias
- ❖ Notificación por correo electrónico al completar una compra
- ❖ Verificación de correo electrónico al crear usuario
- ❖ AWS con mayor capacidad para mantener el programa funcional todo el tiempo sin algún limite en especial