

Candidato: Oscar Eduardo Ortiz Pinzón

Email: oscardo2000@gmail.com

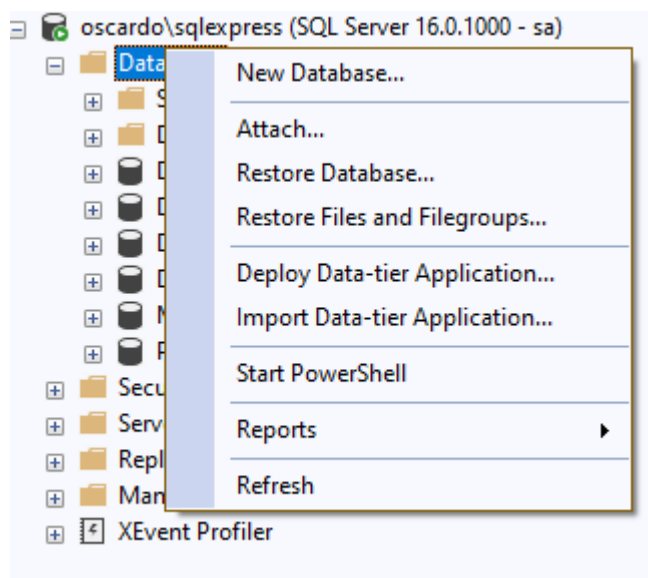
Telefono: +57 3004642976

Base de datos

En la carpeta : C:\AnyFolder\DBSIAP

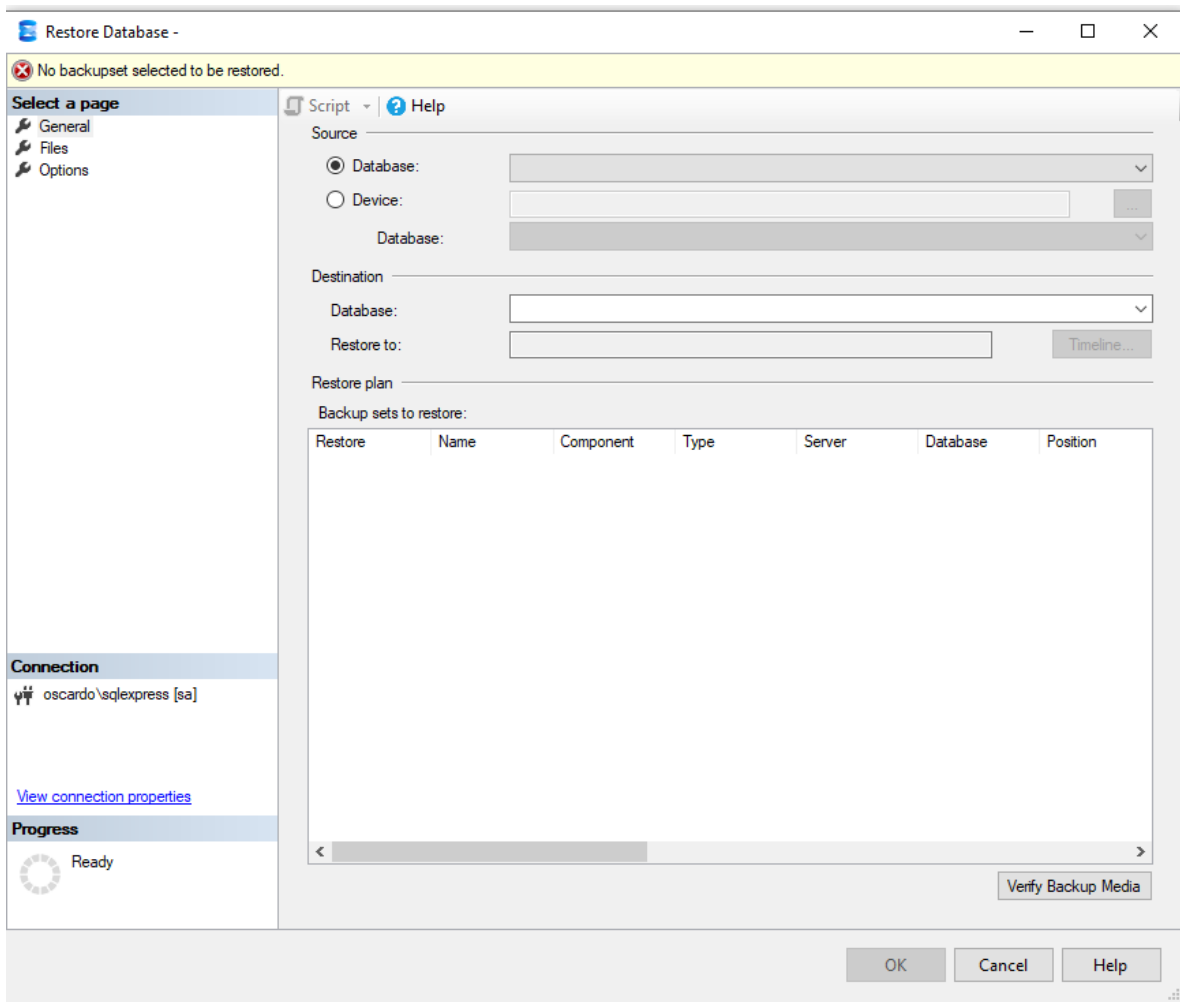
Encontrara: DBSiap.bak la cual si desea restaurar esto:

Implementar el método restauran base de datos desde el backup (DBSiap)



Restore databases

Y seguir las indicaciones paso a paso para esto.



O si desea ver toda la estructura de la base de datos:

C:\AnyFolder\DBSIAP\backup allí se encontrar los distintos scripts:

dbo.Departamento.Table.sql

dbo.Empleado.Table.sql

dbo.Sp_GetDepartamentos.StoredProcedure.sql

dbo.sp_GetEmpleadosPorDepartamento.StoredProcedure.sql

que recrear la base de datos la data (los datos de la aplicación):

C:\Users\OscardoMax\source\repos\DBSIAP\data

users.sql

Departamentos.sql

sp_GetEmpleadosPorDepartamento.sql

store_test.sql

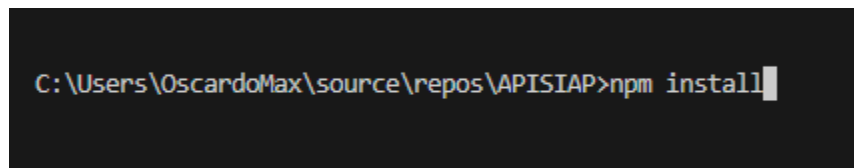
Datos de API

Se necesita tener Node js < 18 o superior, para implementar este requerimiento:

Descomprimir en cualquier carpeta: C:\AnyFolder\APISIAP\ después de esto debemos darle este comando:

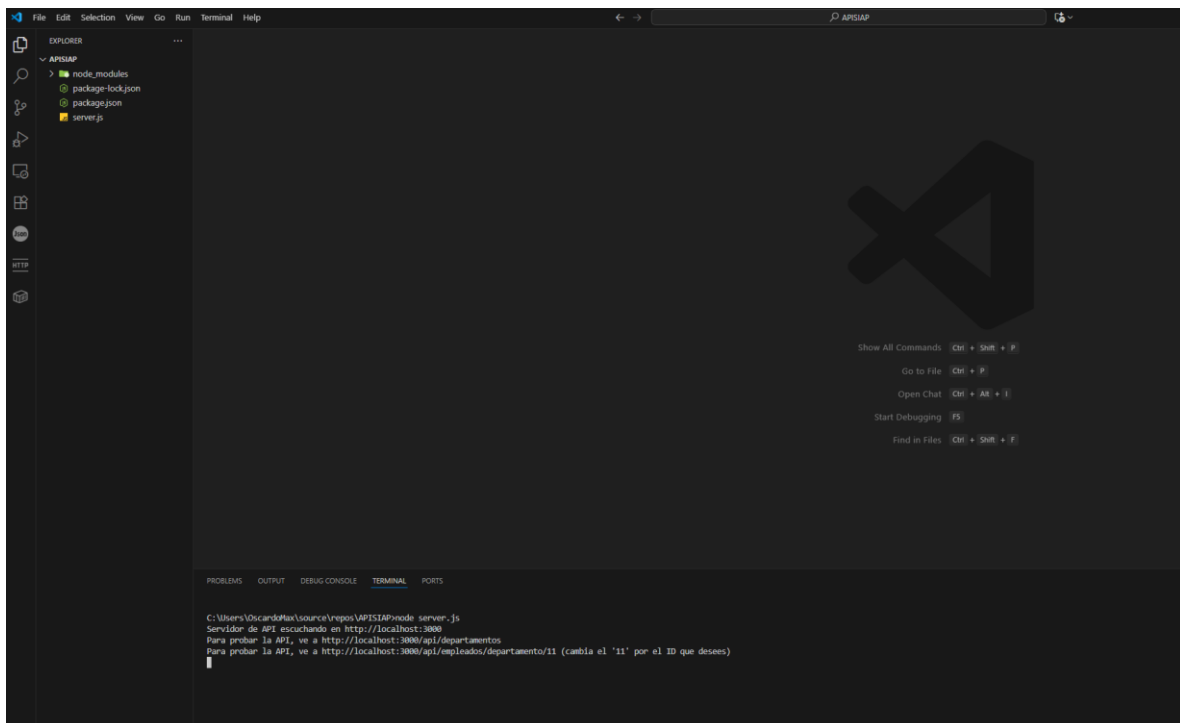
`npm install`

El cual va a instalarse dependiendo de este archivo: package.json

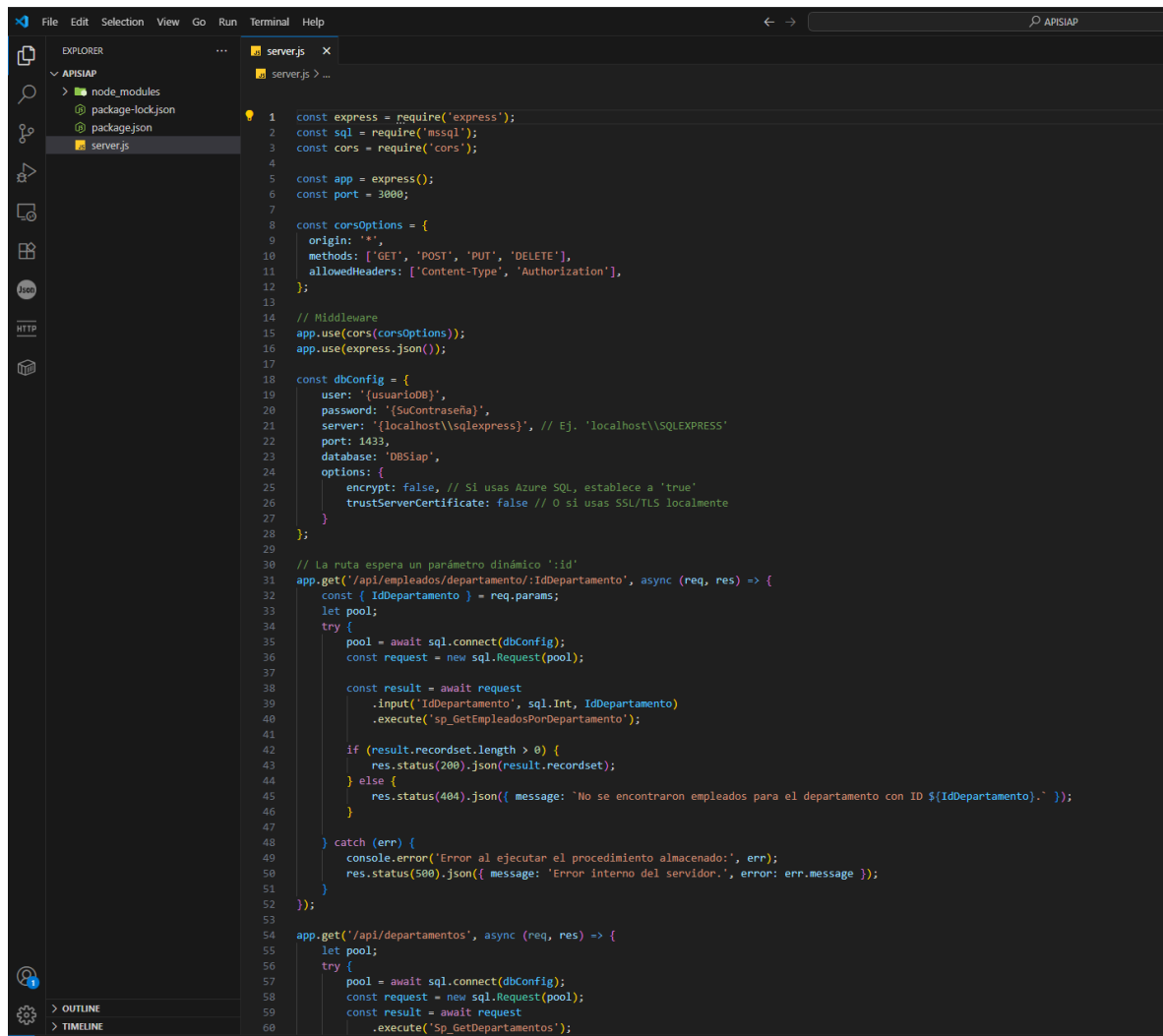


Después de esto, se necesita iniciar este componente:

`node server.js`



Pero después de esto, debe cambiarse dependiendo su base de datos:



```
1  const express = require('express');
2  const sql = require('mssql');
3  const cors = require('cors');
4
5  const app = express();
6  const port = 3000;
7
8  const corsOptions = {
9    origin: '*',
10   methods: ['GET', 'POST', 'PUT', 'DELETE'],
11   allowedHeaders: ['Content-Type', 'Authorization'],
12 };
13
14 // Middleware
15 app.use(cors(corsOptions));
16 app.use(express.json());
17
18 const dbConfig = {
19   user: '{usuarioDB}',
20   password: '{SuContraseña}',
21   server: '{localhost\\sqlexpress}', // Ej. 'localhost\\SQLEXPRESS'
22   port: 1433,
23   database: 'DBSiap',
24   options: {
25     encrypt: false, // Si usas Azure SQL, establece a 'true'
26     trustServerCertificate: false // 0 si usas SSL/TLS localmente
27   }
28 };
29
30 // La ruta espera un parámetro dinámico 'id'
31 app.get('/api/empleados/departamento/:IdDepartamento', async (req, res) => {
32   const { IdDepartamento } = req.params;
33   let pool;
34   try {
35     pool = await sql.connect(dbConfig);
36     const request = new sql.Request(pool);
37
38     const result = await request
39       .input('IdDepartamento', sql.Int, IdDepartamento)
40       .execute('sp_GetEmpleadosPorDepartamento');
41
42     if (result.recordset.length > 0) {
43       res.status(200).json(result.recordset);
44     } else {
45       res.status(404).json({ message: 'No se encontraron empleados para el departamento con ID ${IdDepartamento}.' });
46     }
47   } catch (err) {
48     console.error('Error al ejecutar el procedimiento almacenado:', err);
49     res.status(500).json({ message: 'Error interno del servidor.', error: err.message });
50   }
51 });
52
53 app.get('/api/departamentos', async (req, res) => {
54   let pool;
55   try {
56     pool = await sql.connect(dbConfig);
57     const request = new sql.Request(pool);
58     const result = await request
59       .execute('Sp_GetDepartamentos');
60   }
```

0



```
const dbConfig = {
  user: '{usuarioDB}',
  password: '{SuContraseña}',
  server: '{localhost\\sqlexpress}', // Ej. 'localhost\\SQLEXPRESS'
  port: 1433,
  database: 'DBSiap',
  options: {
    encrypt: false, // Si usas Azure SQL, establece a 'true'
    trustServerCertificate: false // 0 si usas SSL/TLS localmente
  }
};
```

Importante tener todos los datos de base de datos

User

Password

Server

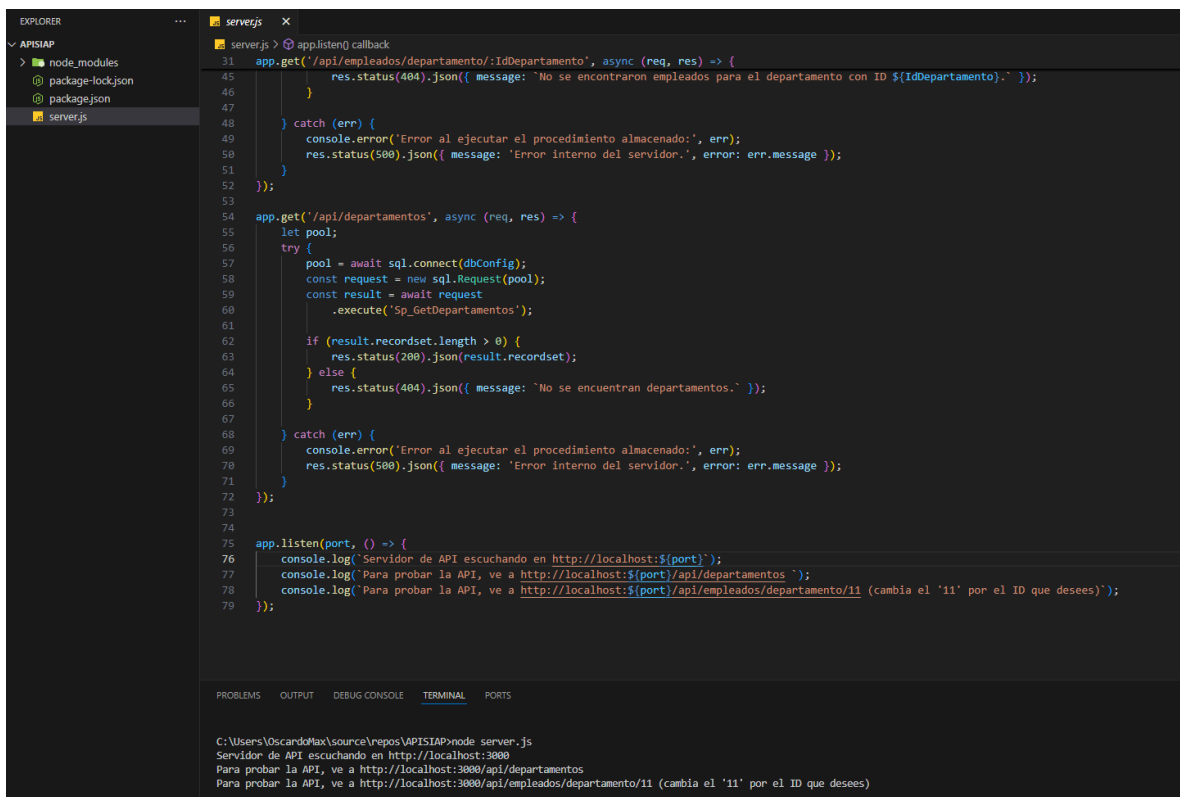
Database

Option {dependiendo si se esta trayendo de una instancia de conexión (Standard Security) SA o desde la trusted connections}.

Ver esta web: <https://www.connectionstrings.com/sql-server-2019/>

Después de esto debe iniciar la aplicación:

node server.js



The screenshot shows a Visual Studio Code editor with a project named 'APISIAP'. The Explorer sidebar on the left shows the file structure: 'node_modules', 'package-lock.json', 'package.json', and 'server.js'. The 'server.js' file is open in the main editor. The code is a Node.js Express application that uses a SQL Server database connection pool. It has two GET endpoints: one for '/api/empleados/departamento/:IdDepartamento' and another for '/api/departamentos'. Both endpoints use a database connection pool to execute queries. The first endpoint returns a 404 status if no employees are found for the given department ID, and a 500 status if there is an internal server error. The second endpoint returns a 200 status if departments are found, and a 404 status if no departments are found, and a 500 status if there is an internal server error. The application is listening on port 3000. The terminal at the bottom shows the command 'C:\Users\OscardoMax\source\repos\APISIAP>node server.js' and the output: 'Servidor de API escuchando en http://localhost:3000', 'Para probar la API, ve a http://localhost:3000/api/departamentos', and 'Para probar la API, ve a http://localhost:3000/api/empleados/departamento/11 (cambia el '11' por el ID que desees)'.

```
server.js > app.listen() callback
31 app.get('/api/empleados/departamento/:IdDepartamento', async (req, res) => {
45     res.status(404).json({ message: 'No se encontraron empleados para el departamento con ID ${IdDepartamento}.' });
46 }
47
48 } catch (err) {
49     console.error('Error al ejecutar el procedimiento almacenado:', err);
50     res.status(500).json({ message: 'Error interno del servidor.', error: err.message });
51 }
52 });
53
54 app.get('/api/departamentos', async (req, res) => {
55     let pool;
56     try {
57         pool = await sql.connect(dbConfig);
58         const request = new sql.Request(pool);
59         const result = await request
60             .execute('Sp_GetDepartamentos');
61
62         if (result.recordset.length > 0) {
63             res.status(200).json(result.recordset);
64         } else {
65             res.status(404).json({ message: 'No se encuentran departamentos.' });
66         }
67     } catch (err) {
68         console.error('Error al ejecutar el procedimiento almacenado:', err);
69         res.status(500).json({ message: 'Error interno del servidor.', error: err.message });
70     }
71 }
72 });
73
74
75 app.listen(port, () => {
76     console.log(`Servidor de API escuchando en http://localhost:${port}`);
77     console.log(`Para probar la API, ve a http://localhost:${port}/api/departamentos `);
78     console.log(`Para probar la API, ve a http://localhost:${port}/api/empleados/departamento/11 (cambia el '11' por el ID que desees)`);
79 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

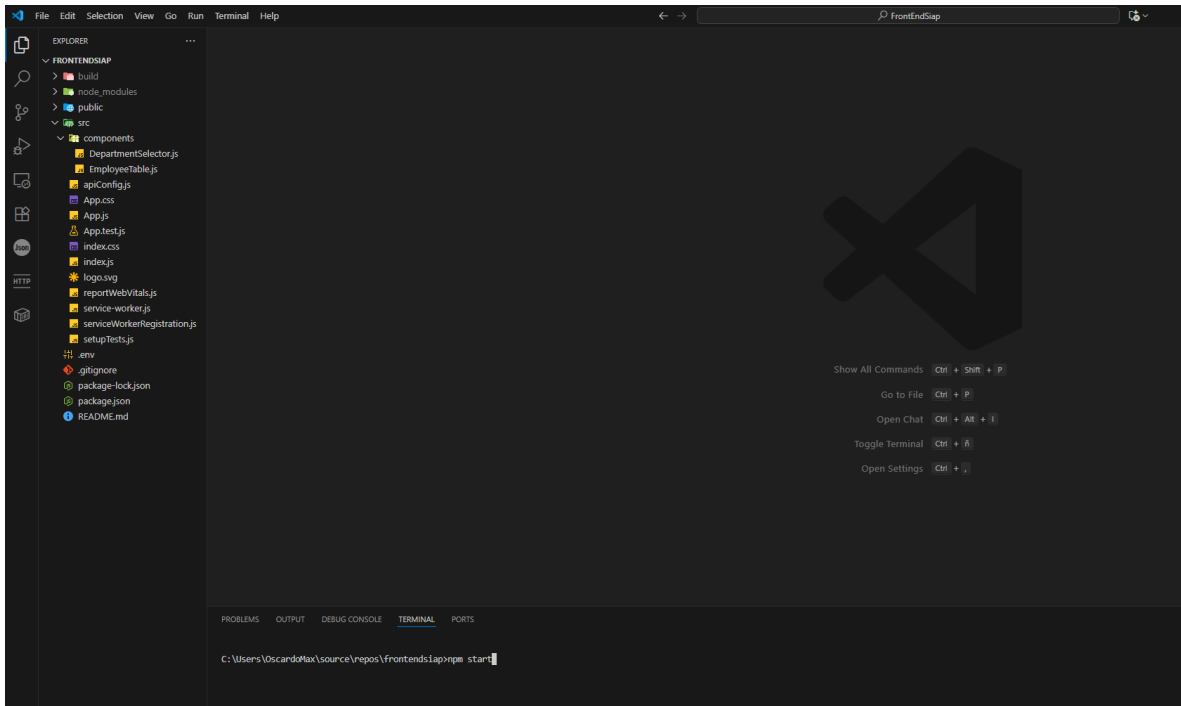
C:\Users\OscardoMax\source\repos\APISIAP>node server.js
Servidor de API escuchando en http://localhost:3000
Para probar la API, ve a http://localhost:3000/api/departamentos
Para probar la API, ve a http://localhost:3000/api/empleados/departamento/11 (cambia el '11' por el ID que desees)

Front End SIAP

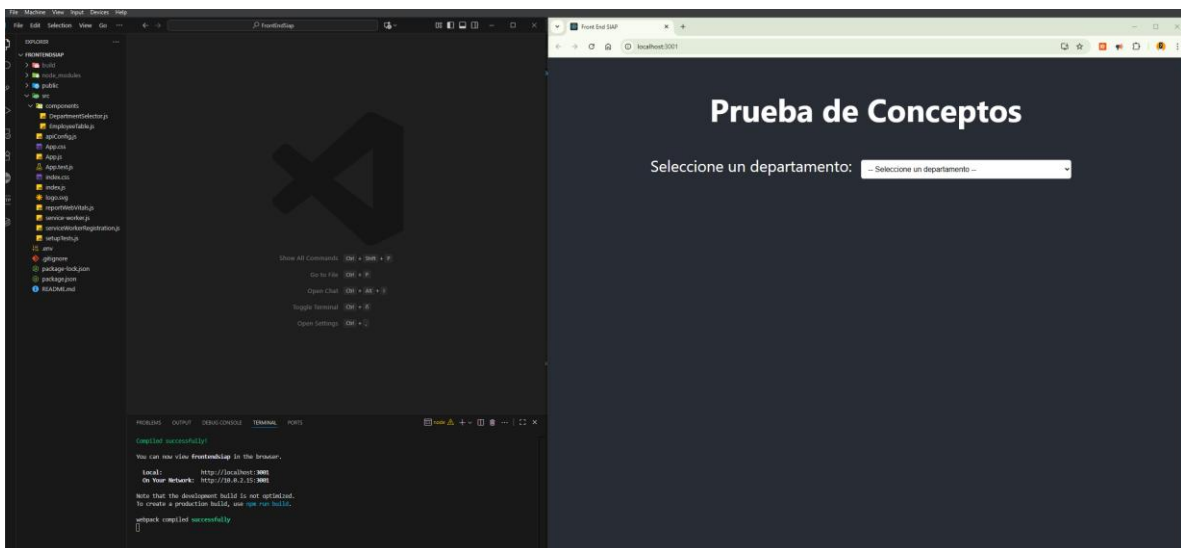
Donde deseamos descomprimir se hará en: C:\AnyFolder\frontendsiap

Tenemos que iniciar la aplicación

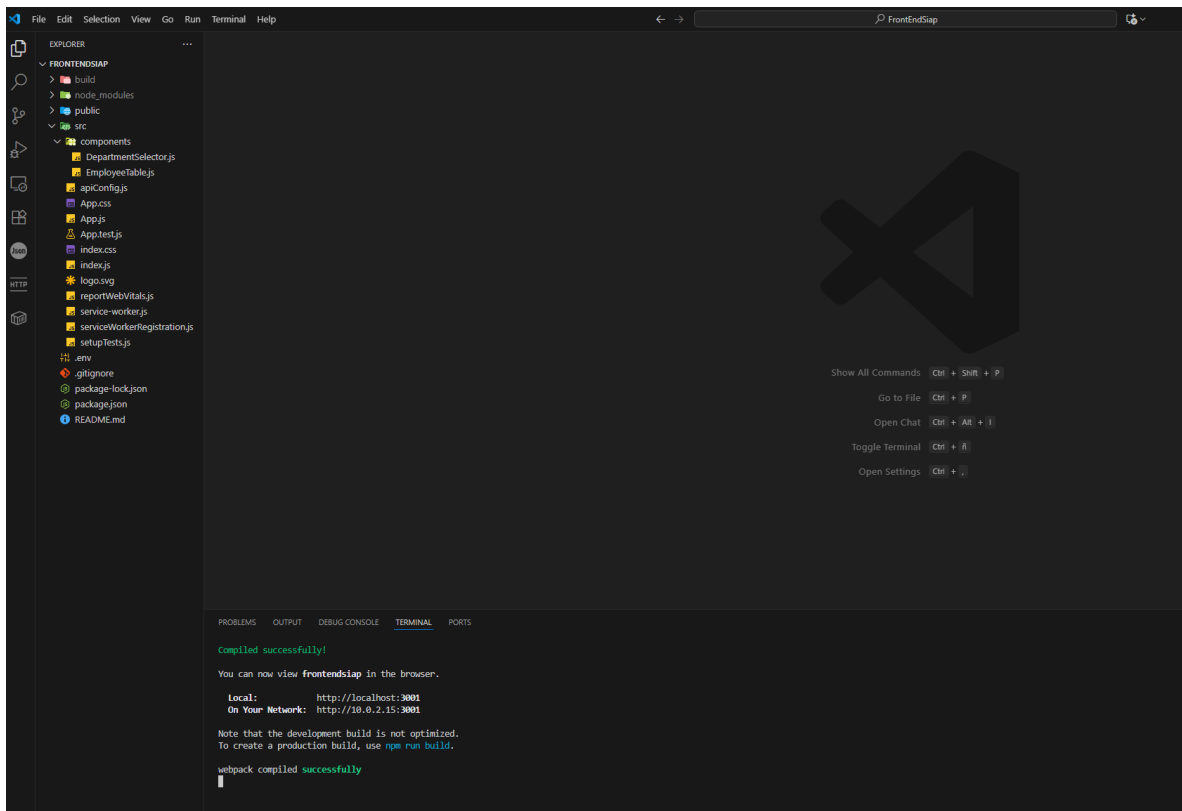
npm install



Este podríamos ejecutar el archivo:



Aquí se ejecuta este archivo:



Y este se ejecutara de esta manera:



Prueba de Conceptos

Seleccione un departamento:

-- Seleccione un departamento --

-- Seleccione un departamento --

Antioquia

Atlántico

Bogotá, D.C.

Bolívar

Boyacá

Caldas

Caquetá

Cauca

Cesar

Córdoba

Cundinamarca

Chocó

Huila

La Guajira

Magdalena

Meta

Nariño

Norte de Santander

Quindío

Prueba de Conceptos

Seleccione un departamento:

Empleados del Departamento

ID Empleado	Nombres	Apellidos	Cargo	Departamento
1	Juan	Pérez	Analista de Datos	Bogotá, D.C.
2	Ana	García	Contador	Bogotá, D.C.
3	Carlos	Rodríguez	Desarrollador Backend	Bogotá, D.C.
4	Laura	Martínez	Diseñadora UX	Bogotá, D.C.
5	Andrés	López	Administrador de Base de Datos	Bogotá, D.C.
6	María	Jiménez	Gerente de Proyecto	Bogotá, D.C.
7	Pedro	Gómez	Técnico de Soporte	Bogotá, D.C.
8	Juliana	Moreno	Reclutadora	Bogotá, D.C.
9	Santiago	Castro	Consultor TI	Bogotá, D.C.

Prueba de Conceptos

Seleccione un departamento:

No se encontraron empleados para este departamento.