Sistema de Información para la Gestión de Proyectos Smart Campus

Jhoan David Ramirez Grajales, ✉ jhoandramirez@estudiante.uniajc.edu.co
Francisco Javier Martinez Vargas, ✉ fjmartinez@estudiante.uniajc.edu.co
Luis Miguel Maury Quintero, ✉ lmmaury@estudiante.uniajc.edu.co

Manual de Instalación

Asesor: Ana Milena Rojas Calero, Magíster (MSc) en Informática y Telecomunicaciones

Institución Universitaria Antonio José Camacho
Facultad de Ingenierías
Ingeniería de sistemas
Cali - Colombia
2021

Contenido

Lista de tablas

Lista de ilustraciones

1. Manual De Instalación

El siguiente documento tiene como objetivo presentar el software y configuración necesaria para realizar la instalación y despliegue del Sistema de Información para la Gestión de Proyectos Smart Campus.

Este manual fue diseñado para realizar el despliegue en un servidor con sistema operativo CentOS 7.

1.1 Definición de Tecnologías

Tabla 1: Definición de tecnologías API

Tecnologías API	
Arquitectura	MVC
Base de Datos	PostgreSQL
Lenguaje	JavaScript
Entorno de Ejecución	NodeJs
Servidor Repositorio	Gitlab
Editor de código	Visual Code
Manejador de paquetes	npm

Nota: Realizado por los autores.

2. Instalación NodeJS y Manejador de Paquetes NPM

NodeJS: es un entorno en tiempo de ejecución de fuente abierta, que se ejecuta en varias plataformas (Windows, Linux, Unix, Mac OS X, etc.). Puede realizar operaciones de gestión de archivos en el servidor, así como también, realizar la gestión de datos en una base de datos. Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente.

Npm: es el manejador de paquetes para NodeJS, facilita el descubrimiento y reutilización de código. Permite administrar las dependencias del proyecto.

Para el despliegue del Api Rest es necesario una versión de Node mayor o igual a v12.x.x a continuación, se exponen los pasos para su instalación en un Servidor CentOS 7:

- Se utiliza el comando curl para descargar los paquetes necesarios para la instalación de NodeJs

```
curl --silent --location https://rpm.nodesource.com/setup_12.x | bash -
```

- Instalar herramientas de compilación

```
yum install gcc-c++ make
```

- Instalar NodeJs y Npm

```
sudo yum install nodejs npm
```

- Comprobar versión de Node

```
node -v
```

- Comprobar versión de Npm

```
npm -v
```

3. Instalación PostgreSQL

- Los repositorios por defecto de CentOS contienen paquetes de Postgres, por lo que podemos instalarlos sin ningún problema usando el sistema de paquetes `yum`.

```
sudo yum install postgresql-server postgresql-contrib
```

- Ahora que nuestro software está instalado, tenemos que realizar algunos pasos antes de poder utilizarlo.
  Crear un nuevo clúster de base de datos PostgreSQL:

```
sudo postgresql-setup initdb
```

- Iniciamos y habilitamos el servicio de PostgreSQL

```
sudo systemctl start postgresql
```

```
sudo systemctl enable postgresql
```

4. Configuración api rest sharepoint (para la gestión de documentos)

En este ejemplo se crea un sitio llamado Uniajc-Test-API

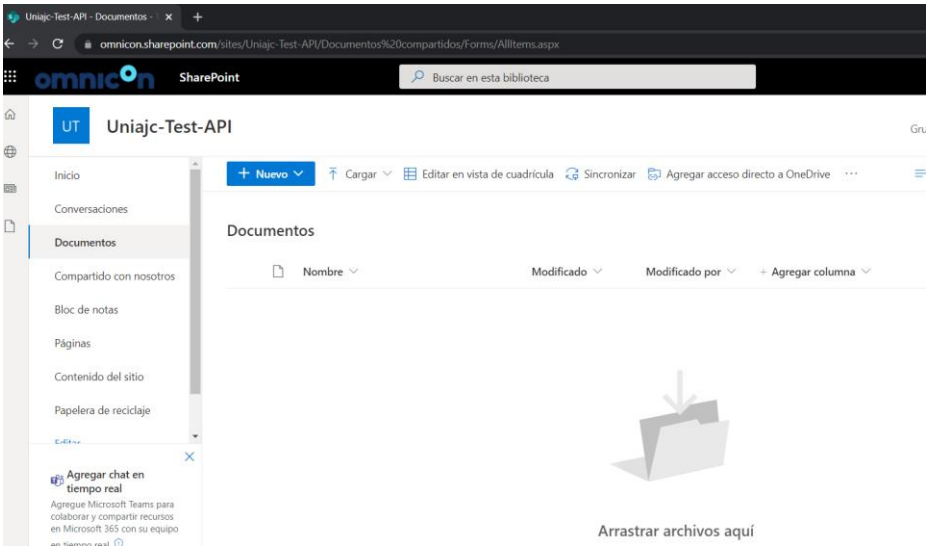


Ilustración 1: sitio de sharepoint

Con el sitio creado podemos generar el client_id y el client_secret agregando **_layouts/15/appregnew.aspx** en la url del sitio.

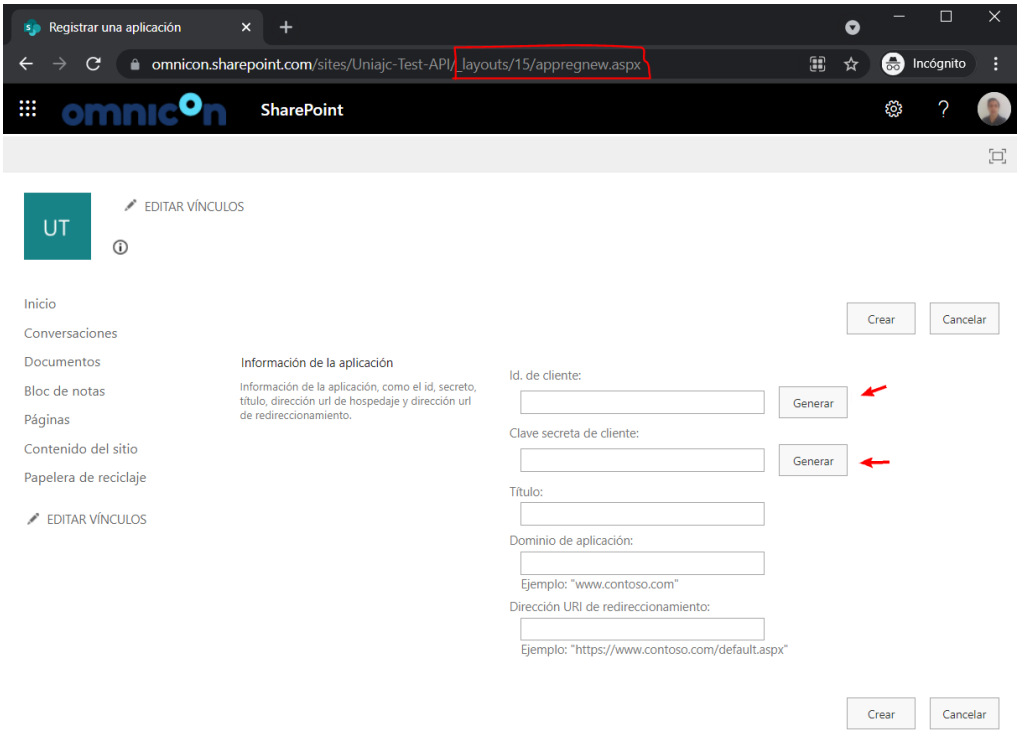


Ilustración 2: creación de client id y client secret

Una vez generados, se llena el campo de título y dominio (estos con localhost), el client_id y el client_secret debería guardarse (se utilizará mas adelante) y le damos en crear.

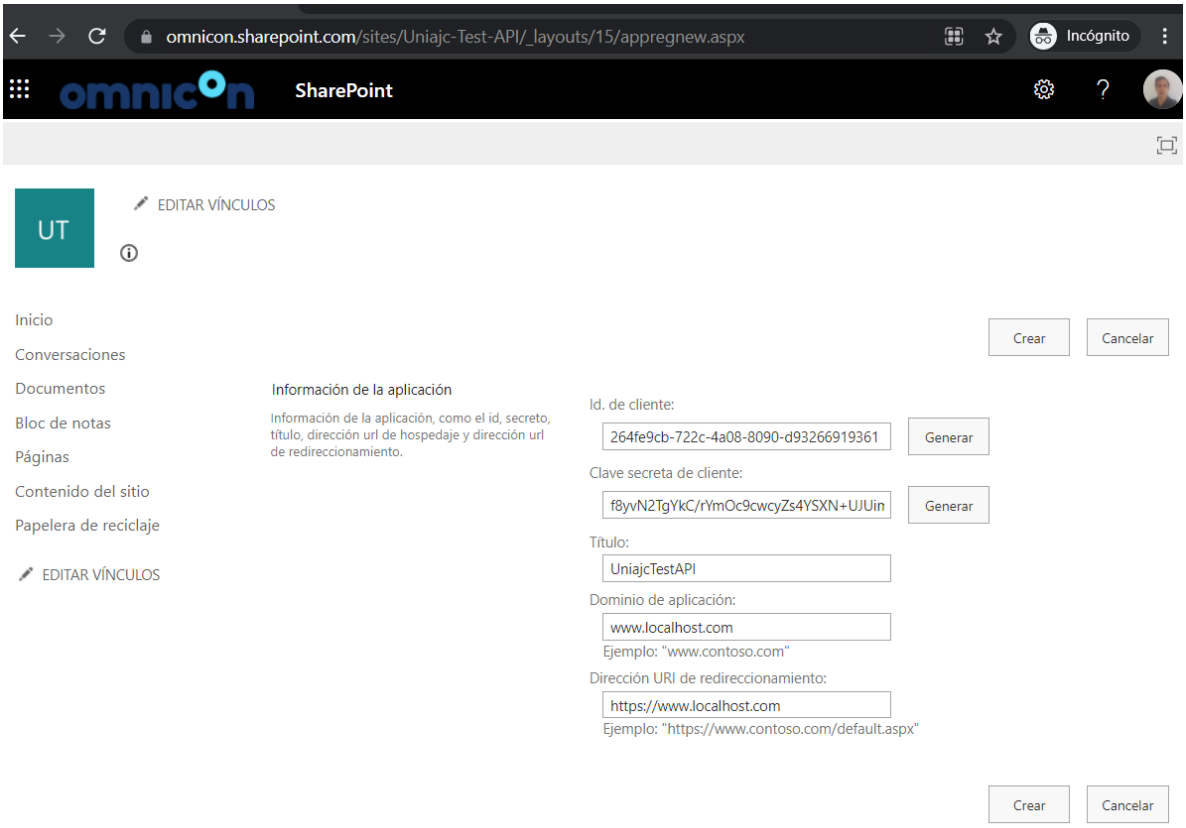


Ilustración 3: formulario

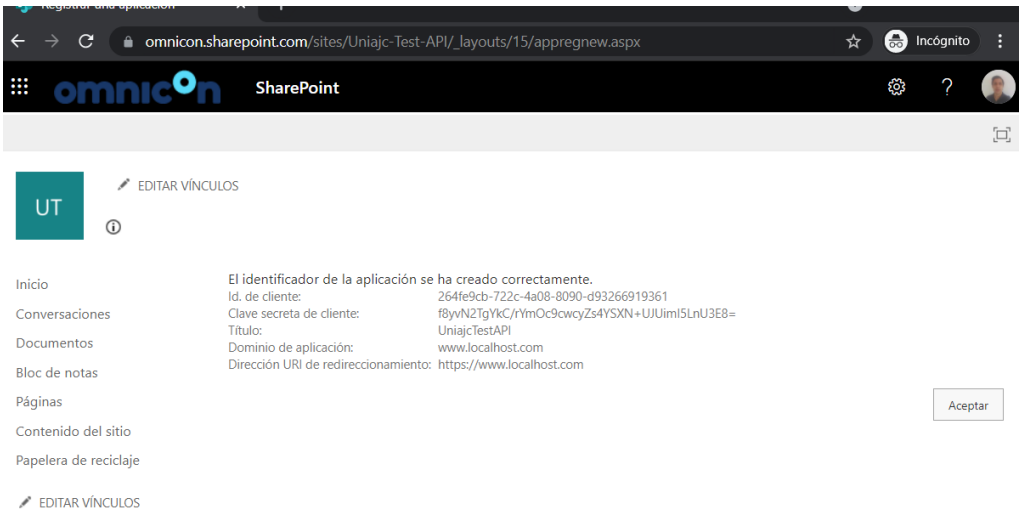


Ilustración 4: datos creados

Ahora le damos permisos de administrador agregando **_layouts/15/appinv.aspx** en la url del sitio.

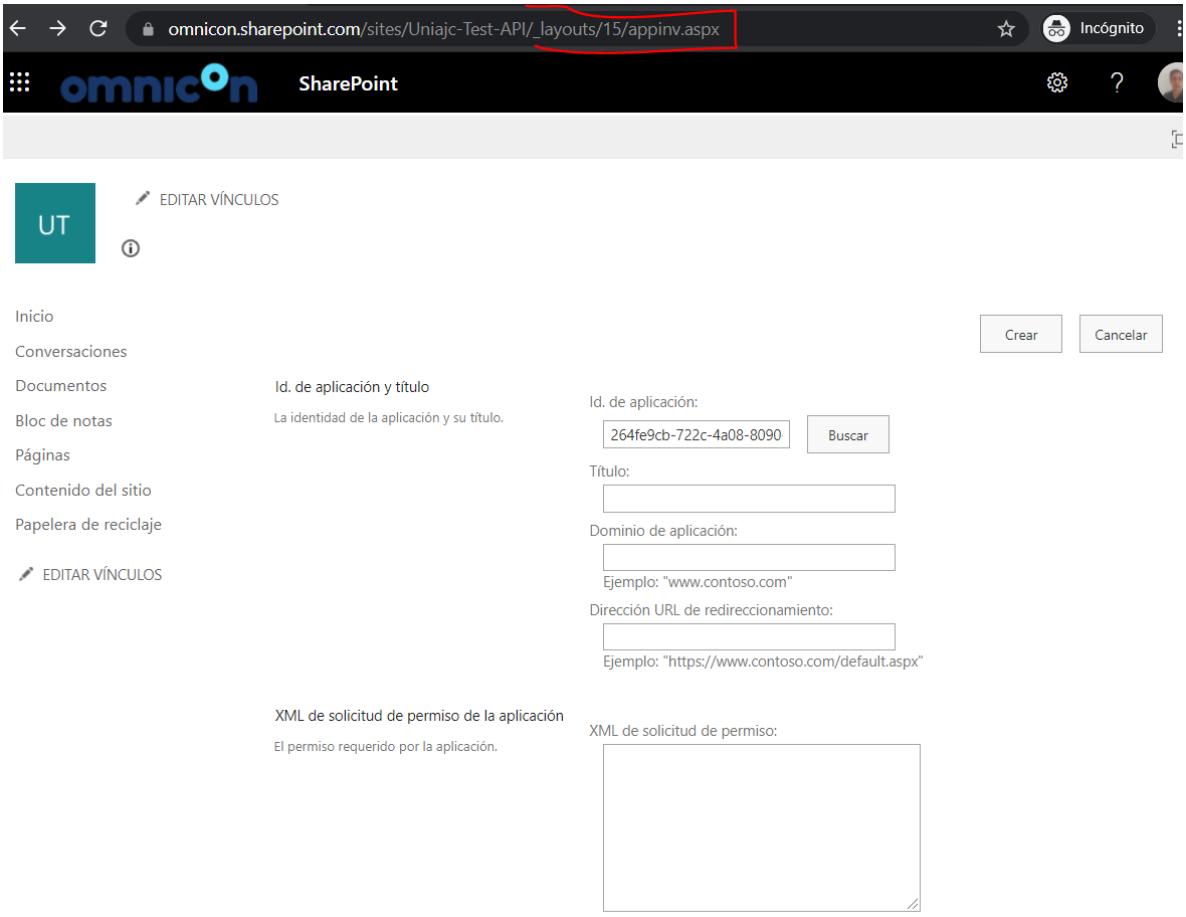


Ilustración 5: Permisos de la app

En id de aplicación pegamos el client_id y le damos click en buscar y nos debería traer el titulo y dominio que le colocamos, en el apartado de permisos de la aplicación le damos permisos de administrador con el siguiente xml

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
  <AppPermissionRequest Scope="http://sharepoint/content/tenant"
Right="FullControl" />
</AppPermissionRequests>
```

y le damos crear

Ilustración 6: formulario de permisos

Una vez creado el permiso de la aplicación debemos conseguir el **tenant_id** agregando **_layouts/15/AppPrincipals.aspx** en la url del sitio.

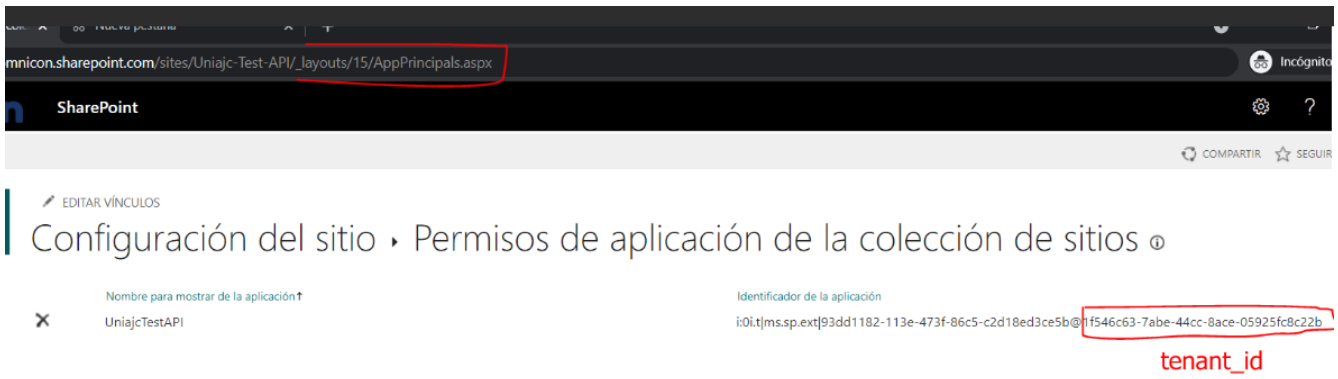


Ilustración 7: verificacion de información

Ya solo nos faltaría la ruta relativa de la carpeta donde se guardaran todos los archivos subidos por medio de la aplicación, para el ejemplo de la imagen 8 la ruta relativa es **"/sites/KairosTest-Sharepoint/Documentos compartidos/Kairos/proyecto de prueba 1"**

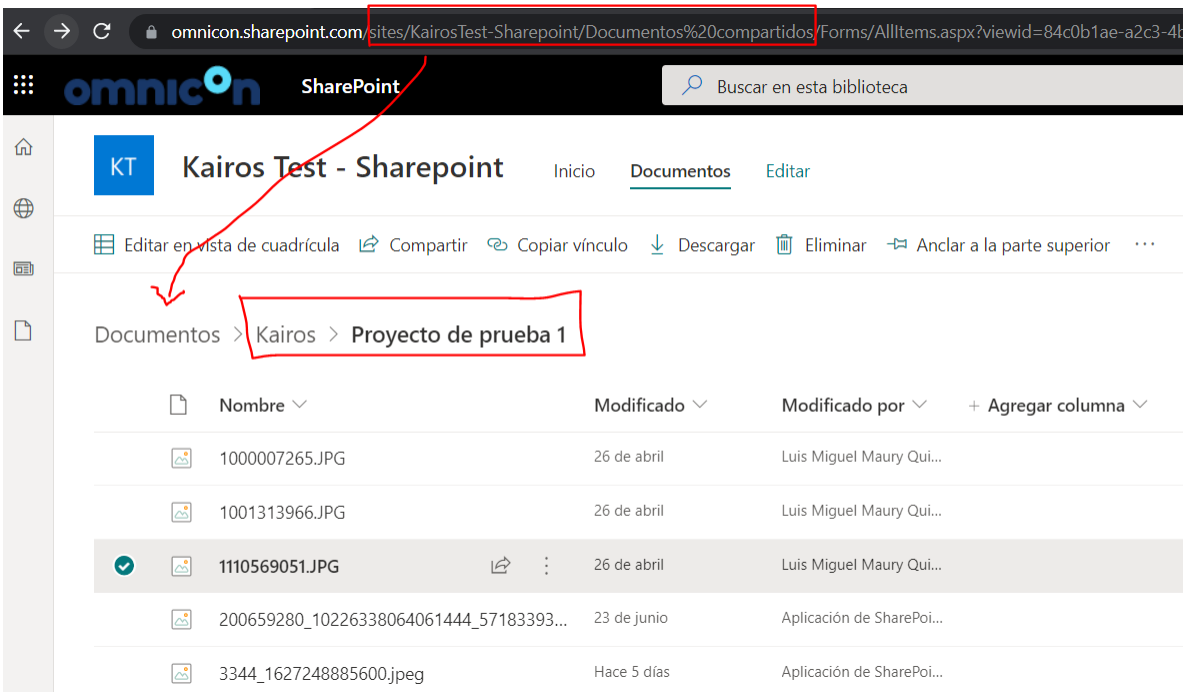


Ilustración 8: carpeta de sharepoint

Con esto ya tenemos todos los valores requeridos para el archivo .env de nuestro backend:

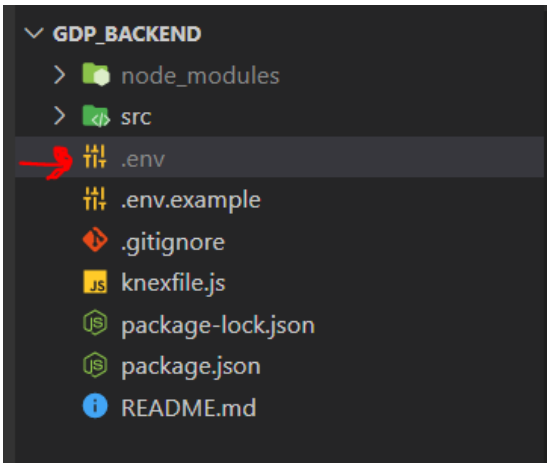
CLIENT_ID = client id del api de sharepoint (ver imagen 3)
CLIENT_SECRET = client secret del api de sharepoint (ver imagen 3)
TENANT = tenant de sharepoint (en este caso es omnicon que es la organización)
TENANTID = tenant id del sharepoint (ver imagen 7)
SITE= sitio del sharepoint (en este ejemplo es Uniajc-Test-API)
RELATIVEFOLDERURL= url relativa de la carpeta de sharepoint donde se guardarán los archivos de la aplicación (ver imagen 8)

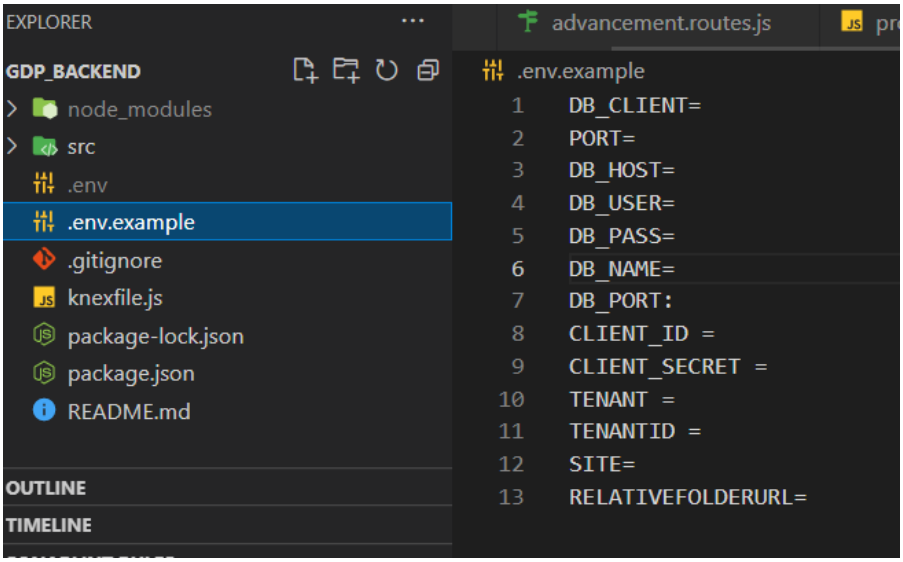
5. configuración del proyecto nodejs

Una vez clonado el repositorio gdp_backend ingresamos en el e instalamos las dependencias configuradas en el proyecto con el comando

```
npm ci
```

una vez instalada las dependencias creamos un archivo .env en la carpeta base del proyecto tomando como plantilla .env.example





DB_CLIENT= el tipo de base de datos: pg
PORT= puerto por el cual el api estará escuchando: 8880
DB_HOST= ip o url de la base de datos: localhost
DB_USER= usuario de base de datos
DB_PASS= contraseña del usuario de base de datos
DB_NAME= nombre de la base de datos
DB_PORT= puerto de la base de datos
CLIENT_ID = client id del api de sharepoint
CLIENT_SECRET = client secret del api de sharepoint
TENANT = tenant de sharepoint
TENANTID = tenant id del sharepoint
SITE= sitio del sharepoint
RELATIVEFOLDERURL= url relativa de la carpeta de sharepoint donde se guardarán los archivos de la aplicación

6. Configurar tablas en base de datos

Una vez teniendo la base de datos podemos crear tablas con el siguiente comando:

```
npx knex migrate:latest
```

- Se llena de datos paramétricos

```
npx knex seed:run
```

Nota: si se desea eliminar las tablas se hace con

```
npx knex migrate:rollback
```

Iniciar el proyecto nodejs

- Para iniciar la aplicación recomendamos utilizar pm2 que es un administrador de procesos basados en node.

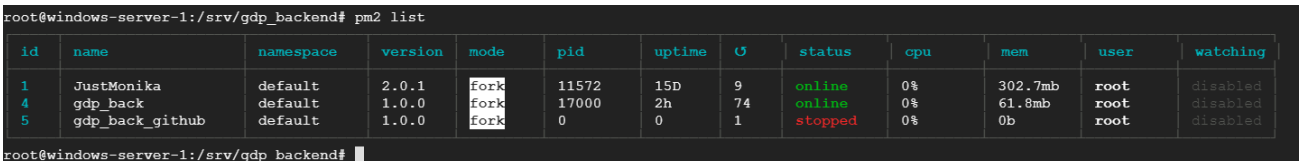
```
npm install pm2 -g
```

- Iniciamos la aplicación de esta forma

```
pm2 start src/index.js --name “gdp_back”
```

- listar los procesos

```
pm2 list
```



Para mas información visitar la página oficial de pm2 https://pm2.keymetrics.io/