Guía de instalación para el proyecto

SISTEMAS DE INFORMACION DE GESTION

Y BUSINESS INTELLIGENCE

YouPlay: Aplicación de un sistema de recomendación

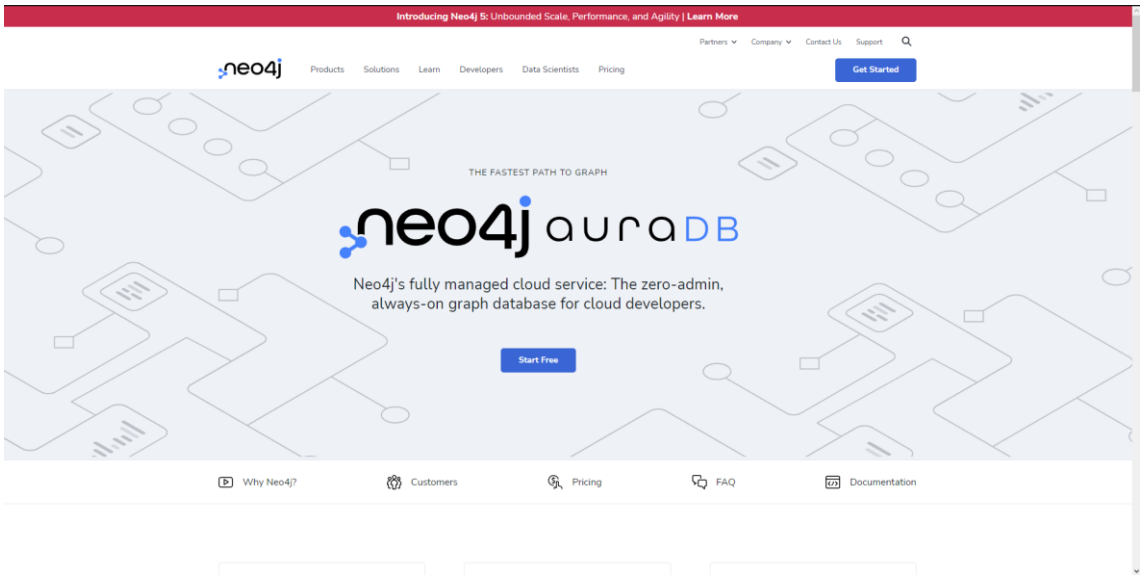
Autor: Víctor de la Fuente Martínez

ÍNDICE

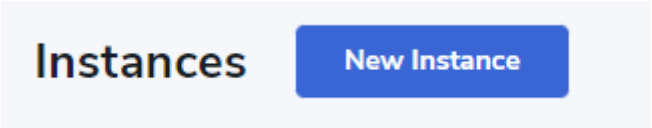
Base de datos y Neo4j

Lo primero que tenemos que hacer es dirigirnos a Neo4j AuraDB: https://neo4j.com/cloud/platform/aura-graph-database/

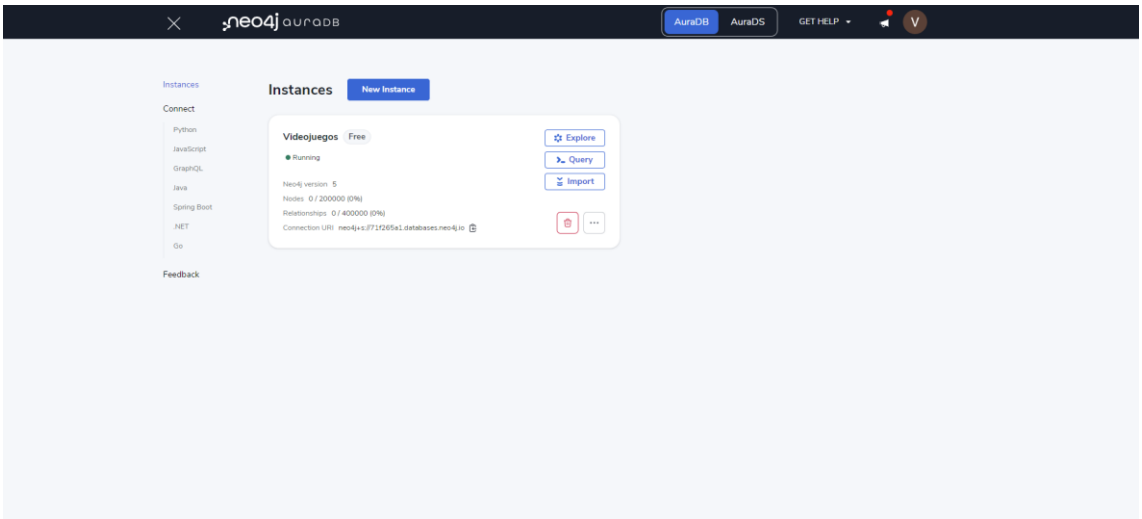
En esta, pinchamos al botón de 'Get Started' para loguearnos y acceder a una base de datos.



Una vez dentro, con el botón de New Instance, crearemos una nueva base de datos vacía.

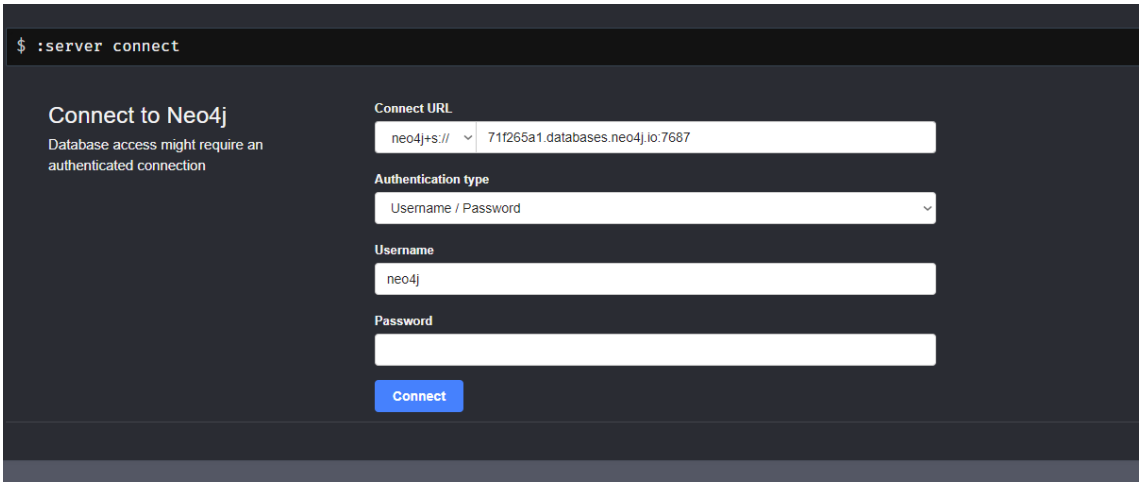


Una vez creada, nos aparecerá algo por el estilo.



En este caso, la he llamado Videojuegos. Aquí nos aparece la uri, justo al lado del botón rojo de papelera. Este es un campo muy importante para, en un futuro, realizar la conexión con la base de datos, donde nos harán falta el username y la password utilizadas para entrar a la base de datos.

Entramos en el botón de query, en la cual accedemos con las credenciales que nos dan al crear la base de datos.



Una vez hemos accedido, procedemos a introducir las queries en el sistema.

Esta sería para crear los nodos Videojuego, Genero, Plataforma y Empresa con sus relaciones:

```
LOAD CSV WITH HEADERS //Cargamos el archivo de videojuegos del Github
FROM 'https://raw.githubusercontent.com/vdelam00/VideojuegosSIBI/main/DatosVideojuegos.csv'
AS row
//Con un pequeño limite para que no colapse por problemas propios de hardware
WITH row LIMIT 3000

//Creamos el nodo Videojuego con los campos que estos tienen y los que necesito en un futuro
CREATE (juego:Videojuego{nombre:row.Name, plataforma:row.Platform, lanzamiento:row.Year_of_Release, genero:row.Genre, empresa:row.Publisher,
puntuacionCritica:row.Critic_Score, puntuacionUsuario:row.User_Score, pegi:row.Rating, like: 0, dislike: 0, favorito: 0, ranking: 0})

//Lo mismo que para el nodo Videojuego
//Creamos el nodo Empresa y su relacion
FOREACH(x IN(CASE WHEN row.Publisher IS NULL THEN[]ELSE[1]END) | MERGE(e:Empresa{nombre:row.Publisher}) CREATE(e)<-[:CREADO_POR]-(juego))

//Creamos el nodo Plataforma y su relacion
FOREACH(x IN(CASE WHEN row.Platform IS NULL THEN[]ELSE[1]END) | MERGE(pl:Plataforma{nombre:row.Platform}) CREATE(pl)<-[:CREADO_PARA]-(juego))

//Creamos el nodo Genero y su relacion
FOREACH(x IN(CASE WHEN row.Genre IS NULL THEN[]ELSE[1]END) | MERGE(g:Genero{nombre:row.Genre}) CREATE(juego)-[:DE_TIPO]->(g))

```

En texto:

//Cargamos del GitHub el archivo csv en que tengo los datos

LOAD CSV WITH HEADERS //Cargamos el archivo de videojuegos de Github

FROM 'https://raw.githubusercontent.com/vdelam00/VideojuegosSIBI/main/DatosVideojuegos.csv'

AS row

//Con un pequeño limite para que no colapse por problemas de hardware

WITH row LIMIT 3000

//Creamos el nodo Videojuego con los campos que estos tienen y los que necesito en un futuro

CREATE (juego:Videojuego{nombre:row.Name, plataforma:row.Platform, lanzamiento:row.Year_of_Release, genero:row.Genre, empresa:row.Publisher, puntuacionCritica:row.Critic_Score, puntuacionUsuario:row.User_Score, pegi:row.Rating, like: 0, dislike: 0, favorito: 0, ranking: 0})

//Lo mismo que para videojuego

//Creamos el nodo Empresa y su relacion

```
FOREACH(x IN(CASE WHEN row.Publisher IS NULL
THEN[]ELSE[1]END) | MERGE(e:Empresa{nombre:row.Publisher})
CREATE(e)<-[:CREADO_POR]-(juego))
```

//Creamos el nodo Plataforma y su relacion

```
FOREACH(x IN(CASE WHEN row.Platform IS NULL
THEN[]ELSE[1]END) |
MERGE(pl:Plataforma{nombre:row.Platform}) CREATE(pl)<-
[:CREADO_PARA]-(juego))
```

//Creamos el nodo Genero y su relacion

```
FOREACH(x IN(CASE WHEN row.Genre IS NULL
THEN[]ELSE[1]END) | MERGE(g:Genero{nombre:row.Genre})
CREATE(juego)-[:DE_TIPO]->(g))
```

Y despues los nodos Usuario:

```
LOAD CSV WITH HEADERS //Cargamos el archivo de usuarios del Github
FROM 'https://raw.githubusercontent.com/vdelam00/VideojuegosSIBI/main/Users.csv'
AS row
WITH row LIMIT 3000

//LLamamos a los nodos Videojuego
MATCH (juego:Videojuego)

//Creamos los nodos usr con sus propiedades cogiendo los atributos de las rows del csv
CREATE (u:Usuario{nombreUsr:row.Usuario,passUsr:row.Pass})

//Relacion con visualizar un videojuego
CREATE (juego)<-[:VISITADO_POR]-(u)
```

En texto:

//VIDEOJUEGOS CREADOS.

//Cargamos los datos de los usuarios del csv guardado en mi GitHub

LOAD CSV WITH HEADERS //Cargamos el archivo de usuarios de Github

FROM 'https://raw.githubusercontent.com/vdelam00/VideojuegosSIBI/main/Users.csv'

AS row

WITH row LIMIT 3000

//LLamamos a los nodos Videojuegos

MATCH (juego:Videojuego)

//Creamos los nodos usr con sus propiedades cogiendo las rows del csv

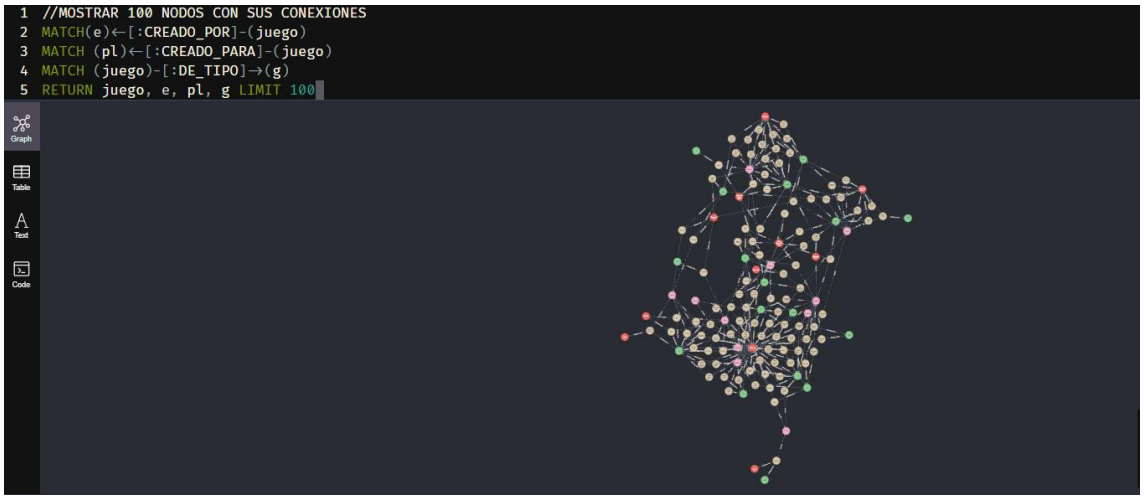
CREATE (u:Usuario{nombreUsr:row.Usuario,passUsr:row.Pass})

//Relacion con visualizar un videojuego

CREATE (juego)<-[:VISITADO_POR]-(u)

Con esto crearíamos todos los nodos y relaciones que nos hacen falta.

Si queremos ver los nodos usaríamos la siguiente instrucción:



Una vez creada la base de datos, en nuestro código iríamos a la parte en la que hace falta conectar la base de datos.

Esta clase se llama db-connector.service.ts. Dentro de esta tenemos el siguiente método:

```
export class DbConnectorService {
  public driver: any;
  constructor() {
    const uri = 'neo4j+s://71f265a1.databases.neo4j.io';
    const user = 'neo4j';
    const password = 'WnRif4GLOsQwEpRoGy4jWevlV5F6E_660b7jlqu_FQg';
    this.driver = neo4j.driver(uri, neo4j.auth.basic(user, password));
  }
```

Aquí tendríamos que modificar la uri, el usuario y la contraseña que pertenezcan a la base de datos que hemos creamos anteriormente. Con esto ya tendríamos conectada la base de datos y conectada a nuestra aplicación para poder empezar a trabajar con ella.

Aplicación

Para hacer uso de la aplicación, primero descargaremos el proyecto completo del GitHub desde: https://github.com/vdelam00/VideojuegosSIBI

A partir de aquí entraremos en la carpeta con la instrucción 'cd .\proyecto-sibi\'

Lo más fácil es abrir una terminal desde el programa Visual Studio Code en la carpeta que acabamos de descargar.

Una vez estemos dentro de la carpeta simplemente metemos la línea de comando 'npm install'. Con esto instalamos todos los componentes necesarios para correr el programa.

Finalmente, escribimos en la línea de comandos de la terminal, desde la carpeta de proyecto-sibi, la siguiente instrucción:

'npm start'

Y con esto se te abrirá en el navegador la página web con el proyecto.

En caso de que no se te abra puedes escribir en el navegador:

'localhost:4200'

Esto te llevara directamente a la página en caso de que no se te abra automáticamente.

```
PS E:\VICTOR\Escritorio\UNI\ULTIMO AÑO COSITAS\SIBI\SIBIVideojuegos> cd .\proyecto-sibi\
```

```
PS E:\VICTOR\Escritorio\UNI\ULTIMO AÑO COSITAS\SIBI\SIBIVideojuegos\proyecto-sibi> npm install
```

```
PS E:\VICTOR\Escritorio\UNI\ULTIMO AÑO COSITAS\SIBI\SIBIVideojuegos\proyecto-sibi> npm start
```

```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

√ Compiled successfully.
```

Para todo este proceso se necesitará la versión 17.9.1 de Node.js ya que es la que utiliza el npm como instalador e iniciador de la aplicación.