

Manipulación de grafos con Python y Orientdb

Carlos Emanuel Balcazar

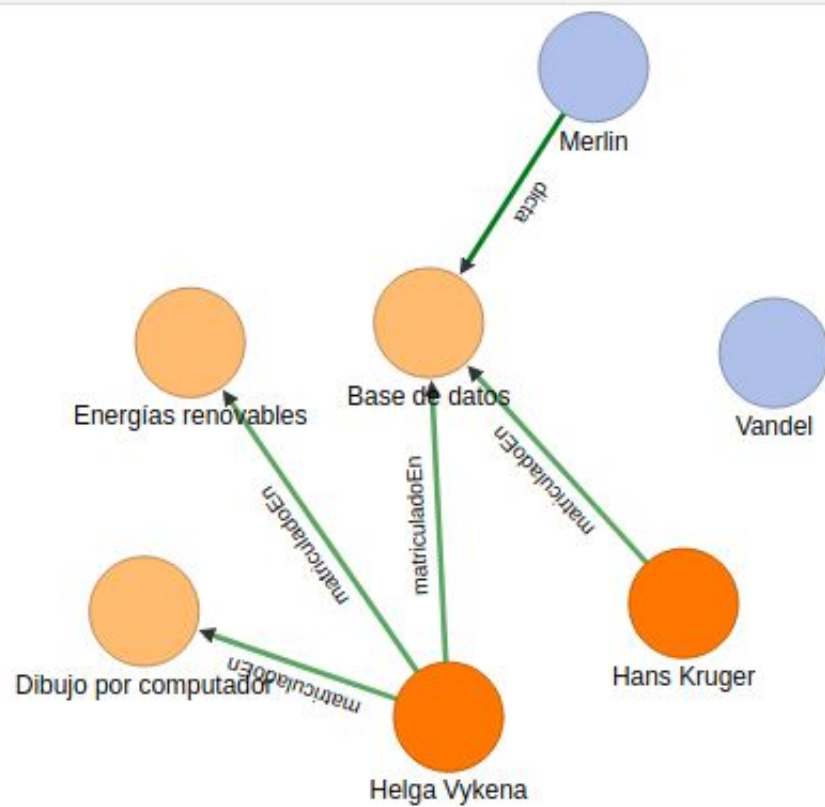


Antes de empezar... ¿grafos?

- Atributos y relaciones representadas como un grafo.
- Consultas más amplias y no limitadas por tablas
- Las búsquedas tienen un alto nivel de efectividad y eficiencia



- Alumno
- Profesor
- Curso
- matriculadoEn
- dicta



¿Qué haremos?

1. Crear una nueva base de datos.
2. Crear nuestro esquema desde un script.
3. Cargar algunos datos desde un script.
4. Conectarnos con python a la base de datos.
5. Devolver algunos datos.

¿Que es orientdb?

- OrientDB es una base de datos nosql multimodelo
- Escrito en Java y de código abierto
- Admite grafos, documentos, clave/valor y objetos.
- Admite modos sin esquema, esquema completo y esquema mixto.

OrientDB vs Neo4j

- Neo4j es una base de datos orientada a grafos, muy popular entre los desarrolladores.
- Sin embargo nos hemos encontrado con muchas complicaciones a la hora de utilizarla.
- Importación/exportación de base de datos, problemas de concurrencia, no tiene primary keys ni definición de esquemas.
- Su versión enterprise está escondida en su página!

Primero lo primero

- Antes de conectarnos con python, debemos crear nuestra propia base de datos y sus esquemas. Además cargaremos algunos datos para poder hacer una prueba.

Creando nuestras propias clases

Ver ejemplos:

- **scripts/10_schema.sql**
- **scripts/20_data.sql**

Ejecutar: `bin/command.sh script.sql`

Algunas consultas que podemos hacer

- Ver ejemplo: **scripts/casos_select_match.sql**

Conectémonos a Orient con Python!

- Para ello utilizaremos su driver oficial: **pyorient**
- Se puede descargar utilizando el sistema de gestión de paquetes de python.
- Sin embargo esta desactualizado, para hacerlo funcionar hay que hacer un pequeño truco...
- `SUPPORTED_PROTOCOL = 37`

```
import pyorient

client = pyorient.OrientDB("localhost", 2424)
session_id = client.connect("root", "root")
client.db_open("demodb", "root", "root")
records = client.query('select from OUser')

print (records[0].oRecordData)

client.db_close()
```

Ahora implementemos una API con Flask

Primero, creamos nuestro archivo **app.py** o como quieran llamarle y le incorporamos la conexión a orient

Ver ejemplo: **src/app.py**

```
app = Flask(__name__)

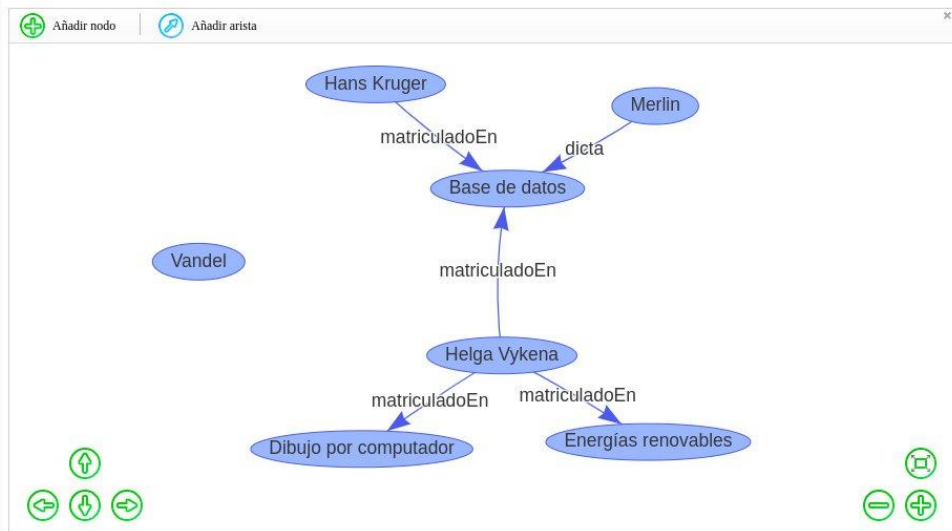
# solo para pruebas
DATABASE = "meetup"
HOST = "localhost"
PORT = 2424
USER = "root"
PASSWORD = "root"

def get_connection():
    client = pyorient.OrientDB(HOST, PORT)
    session_id = client.connect(USER, PASSWORD)
    client.db_open(DATABASE, USER, PASSWORD)

    return client
```

Una forma de visualizar los datos

Visualizador de Grafos



Se pueden utilizar librerías para visualizar los datos de la base, una de ellas es vis.js

[VIS JS](#)

¿Preguntas?

