|  |
| --- |
|  |
| Neo4j  Base de datos universitaria española  Neo4j: The World's Leading Graph Database |
| 12 enero  RI - PL6  Creado por:  Sergio Cimadevilla Torga - UO264292  Chen Xin Pan Wang - UO276967  Eduardo Blanco Bielsa – UO285176 |

Índice

[Dominio de la aplicación 3](#_Toc124425809)

[Gráfico de la base de datos 4](#_Toc124425810)

[Código de creación del grafo 4](#_Toc124425811)

[Consultas 8](#_Toc124425812)

[Aplicación en pyhton para las consultas 13](#_Toc124425813)

# Dominio de la aplicación

Se ha escogido el tema de “base de datos de ámbito universitario”.

En esta aplicación se representan una serie de grados universitarios de distintos lugares de españa: Ingeniería del Software (Oviedo), Ingeniería Informática (Barcelona) y Química (Málaga); junto con las asignaturas de dichos grados (y que pueden estar compartidas por varios grados); los profesores que imparten las diversas asignaturas; los alumnos que están apuntados en ellas y los directores de cada grado (símil a la secretaría del grado).

Diagrama

Descripción generada automáticamente

*Esquema conceptual general*

# Gráfico de la base de datos

Patrón de fondo

Descripción generada automáticamente

En esta imagen se representa la totalidad de la base de datos del trabajo.

En ella se pueden apreciar los distintos tipos de nodos existentes (Persona, Grado y Asignatura) con sus correspondientes relaciones (ESTA\_APUNTADO, IMPARTE, DIRIGE y CONTIENE). Los profesores, alumnos y directores son del mismo tipo de nodo (Persona).

# Código de creación del grafo

create (Ingenieria\_del\_software:Grado{nombre:"Ingenieria del software",facultad:"Escuela de Ingenieria Informatica", localizacion:"Oviedo"}),

(Quimica:Grado{nombre:"Quimica",facultad:"Facultad de ciencias de la UMA", localizacion:"Malaga"}),

(Ingenieria\_informatica:Grado{nombre:"Ingenieria Informatica",facultad:"Escuela politecnica de Barcelona", localizacion:"Barcelona"});

create (RI:Asignatura{codigo:"1",nombre:"Repositorios de la Informacion",rama:"Ingenieria"}),

(Algoritmia:Asignatura{codigo:"2",nombre:"Algoritmia",rama:"Ingenieria"}),

(Software\_para\_Robots:Asignatura{codigo:"3",nombre:"Software para Robots",rama:"Ingenieria"}),

(Seguridad:Asignatura{codigo:"4",nombre:"Seguridad",rama:"Ingenieria"}),

(Fisica:Asignatura{codigo:"5",nombre:"Fisica",rama:"Quimica"}),

(Quimica\_organica:Asignatura{codigo:"6",nombre:"Quimica Organica",rama:"Quimica"}),

(Quimica\_inorganica:Asignatura{codigo:"7",nombre:"Quimica inorganica",rama:"Quimica"}),

(Calculo:Asignatura{codigo:"8",nombre:"Calculo",rama:"Ingenieria"}),

(Algebra:Asignatura{codigo:"9",nombre:"Algebra",rama:"Ingenieria"});

create (Eduardo:Persona{identificador:"UO285176", nombre:"Eduardo", apellidos:"Blanco Bielsa", edad:20, correo:"UO285176@uniovi.es"}),

(Sergio:Persona{identificador:"UO264292", nombre:"Sergio", apellidos:"Cimadevilla Torga", edad:22, correo:"UO264292@uniovi.es"}),

(Ruben:Persona{identificador:"0611133597", nombre:"Ruben", apellidos:"Sanchez Macia", edad:19, correo:"rubensama2004@uma.es"}),

(Marcos:Persona{identificador:"0611133598", nombre:"Marcos", apellidos:"Perez Gonzalez", edad:19, correo:"marcosperez@uma.es"}),

(Chen\_Xin:Persona{identificador:"UO276967", nombre:"Chen Xin", apellidos:"Pan Wang", edad:21, correo:"UO276967@uniovi.es"}),

(Daniel:Persona{identificador:"UO276978", nombre:"Daniel", apellidos:"Fernandez de Juan", edad:21, correo:"UO276978@uniovi.es"});

create (Dario:Persona{identificador:"dario", nombre:"Dario", apellidos:"Alvarez Gutierrez", edad:40, correo:"darioa@uniovi.es"}),

(Enol:Persona{identificador:"enol", nombre:"Enol", apellidos:"Garcia Gonzalez", edad:25, correo:"garciaenol@uniovi.es"}),

(Enrique:Persona{identificador:"galiana", nombre:"Enrique J.", apellidos:"de Andres Galiana", edad:40, correo:"andresenrique@uniovi.es"}),

(Alberto:Persona{identificador:"alb", nombre:"Alberto Manuel", apellidos:"Fernandez Alvarez", edad:42, correo:"alb@uniovi.es"}),

(Lucas:Persona{identificador:"lvc4s", nombre:"Lucas", apellidos:"Blanco Martinez", edad:56, correo:"lvc4s@uma.es"}),

(Jose:Persona{identificador:"jose", nombre:"Jose", apellidos:"Fernandez Fernandez", edad:58, correo:"jose@uma.es"});

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Eduardo" and p.apellidos = "Blanco Bielsa" AND (a.codigo = "1" OR a.codigo = "2" OR a.codigo = "3" OR a.codigo ="4")

CREATE (p)-[:ESTA\_APUNTADO]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Sergio" and p.apellidos = "Cimadevilla Torga" AND (a.codigo = "1" OR a.codigo = "9" OR a.codigo = "4")

CREATE (p)-[:ESTA\_APUNTADO]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Ruben" and p.apellidos = "Sanchez Macia" AND (a.codigo = "9" OR a.codigo = "5" OR a.codigo = "6" OR a.codigo ="7" OR a.codigo ="8")

CREATE (p)-[:ESTA\_APUNTADO]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Marcos" and p.apellidos = "Perez Gonzalez" AND (a.codigo = "9" OR a.codigo = "6" OR a.codigo ="8")

CREATE (p)-[:ESTA\_APUNTADO]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Chen Xin" and p.apellidos = "Pan Wang" AND (a.codigo = "1" OR a.codigo = "3" OR a.codigo = "4" OR a.codigo ="9" OR a.codigo ="2")

CREATE (p)-[:ESTA\_APUNTADO]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Daniel" and p.apellidos = "Fernandez de Juan" AND (a.codigo = "2" OR a.codigo = "9" OR a.codigo = "5" OR a.codigo ="8")

CREATE (p)-[:ESTA\_APUNTADO]->(a);

MATCH (p: Persona), (g:Grado)

WHERE p.nombre = "Dario" and p.apellidos = "Alvarez Gutierrez" AND g.nombre = "Ingenieria del software"

CREATE (p)-[:DIRIGE]->(g);

MATCH (p: Persona), (g:Grado)

WHERE p.nombre = "Enrique J." and p.apellidos = "de Andres Galiana" AND g.nombre = "Ingenieria Informatica"

CREATE (p)-[:DIRIGE]->(g);

MATCH (p: Persona), (g:Grado)

WHERE p.nombre = "Lucas" and p.apellidos = "Blanco Martinez" AND g.nombre = "Quimica"

CREATE (p)-[:DIRIGE]->(g);

MATCH (a: Asignatura), (g:Grado)

WHERE g.nombre = "Ingenieria del software" AND (a.codigo = "1" OR a.codigo = "2" OR a.codigo = "3" OR a.codigo = "4" OR a.codigo = "8" OR a.codigo = "9")

CREATE (g)-[:CONTIENE]->(a);

MATCH (a: Asignatura), (g:Grado)

WHERE g.nombre = "Ingenieria Informatica" AND (a.codigo = "2" OR a.codigo = "4" OR a.codigo = "5" OR a.codigo = "8" OR a.codigo = "9")

CREATE (g)-[:CONTIENE]->(a);

MATCH (a: Asignatura), (g:Grado)

WHERE g.nombre = "Quimica" AND (a.codigo = "5" OR a.codigo = "6" OR a.codigo = "7" OR a.codigo = "8" OR a.codigo = "9")

CREATE (g)-[:CONTIENE]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Dario" and p.apellidos = "Alvarez Gutierrez" AND (a.codigo = "1" OR a.codigo = "2")

CREATE (p)-[:IMPARTE]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Enol" and p.apellidos = "Garcia Gonzalez" AND (a.codigo = "1" OR a.codigo = "2" OR a.codigo = "4")

CREATE (p)-[:IMPARTE]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Enrique J." and p.apellidos = "de Andres Galiana" AND (a.codigo = "4" OR a.codigo = "5" OR a.codigo = "8")

CREATE (p)-[:IMPARTE]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Lucas" and p.apellidos = "Blanco Martinez" AND (a.codigo = "6" OR a.codigo = "7" OR a.codigo = "8" OR a.codigo = "9")

CREATE (p)-[:IMPARTE]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Alberto Manuel" and p.apellidos = "Fernandez Alvarez" AND (a.codigo = "1" OR a.codigo = "8" OR a.codigo = "9")

CREATE (p)-[:IMPARTE]->(a);

MATCH (p: Persona), (a:Asignatura)

WHERE p.nombre = "Jose" and p.apellidos = "Fernandez Fernandez" AND (a.codigo = "3" OR a.codigo = "2")

CREATE (p)-[:IMPARTE]->(a);

# Consultas

1. Elementales
2. Devolver el nombre de las asignaturas que contiene Grado de “Ingeniería informática”

match (a: Asignatura) <-[:CONTIENE]- (:Grado{nombre:"Ingenieria Informatica"})

return a

╒════════════════════════════════════════════════════════╕

│"a" │

╞════════════════════════════════════════════════════════╡

│{"codigo":"9","rama":"Ingenieria","nombre":"Algebra"} │

├────────────────────────────────────────────────────────┤

│{"codigo":"5","rama":"Quimica","nombre":"Fisica"} │

├────────────────────────────────────────────────────────┤

│{"codigo":"8","rama":"Ingenieria","nombre":"Calculo"} │

├────────────────────────────────────────────────────────┤

│{"codigo":"4","rama":"Ingenieria","nombre":"Seguridad"} │

├────────────────────────────────────────────────────────┤

│{"codigo":"2","rama":"Ingenieria","nombre":"Algoritmia"}│

└────────────────────────────────────────────────────────┘

Patrón de fondo

Descripción generada automáticamente con confianza media

1. Devolver el nombre de los alumnos que estén apuntados en Álgebra

match (p:Persona)-[:ESTA\_APUNTADO]->(a:Asignatura{nombre:"Algebra"})

return p

╒═══════════════════════════════════════════════════════════╕

│"p" │

╞══════════════════════════════════════════════════════════════════════╡

│{"apellidos":"Fernandez de Juan","correo":"UO276978@uniovi.es","nombre│

│":"Daniel","edad":"21","identificador":"UO276978"} │

├──────────────────────────────────────────────────────────────────────┤

│{"apellidos":"Pan Wang","correo":"UO276967@uniovi.es","nombre":"Chen X│

│in","edad":"21","identificador":"UO276967"} │

├──────────────────────────────────────────────────────────────────────┤

│{"apellidos":"Perez Gonzalez","correo":"marcosperez@uma.es","nombre":"│

│Marcos","edad":"19","identificador":"0611133598"} │

├──────────────────────────────────────────────────────────────────────┤

│{"apellidos":"Sanchez Macia","correo":"rubensama2004@uma.es","nombre":│

│"Ruben","edad":"19","identificador":"0611133597"} │

├──────────────────────────────────────────────────────────────────────┤

│{"apellidos":"Cimadevilla Torga","correo":"UO264292@uniovi.es","nombre│

│":"Sergio","edad":"22","identificador":"UO264292"} │

└──────────────────────────────────────────────────────────────────────┘

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Intermedias

1. Devolver el nombre de los alumnos que están apuntados en Cálculo en un grado que no esté en Oviedo

match (a: Asignatura) <- [:CONTIENE] - (g:Grado)

where g.localizacion <> "Oviedo" and a.nombre = "Calculo"

match (a) <- [:ESTA\_APUNTADO] - (p:Persona)

return distinct(p)

╒═══════════════════════════════════════════════════════════╕

│"p" │

╞══════════════════════════════════════════════════════════════════════╡

│{"apellidos":"Fernandez de Juan","correo":"UO276978@uniovi.es","nombre│

│":"Daniel","edad":"21","identificador":"UO276978"} │

├──────────────────────────────────────────────────────────────────────┤

│{"apellidos":"Perez Gonzalez","correo":"marcosperez@uma.es","nombre":"│

│Marcos","edad":"19","identificador":"0611133598"} │

├──────────────────────────────────────────────────────────────────────┤

│{"apellidos":"Sanchez Macia","correo":"rubensama2004@uma.es","nombre":│

│"Ruben","edad":"19","identificador":"0611133597"} │

└──────────────────────────────────────────────────────────────────────┘

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Devolver el nombre de todos los profesores que imparten alguna asignatura y no son directores de ningún grado

MATCH (p:Persona)-[:IMPARTE]->(s:Asignatura)

OPTIONAL MATCH (p)-[:DIRIGE]->(g:Grado)

WITH p, COUNT(g) AS contador

WHERE contador = 0

RETURN p

╒═══════════════════════════════════════════════════════════╕

│"p" │

╞══════════════════════════════════════════════════════════════════════╡

│{"apellidos":"Fernandez Alvarez","correo":"alb@uniovi.es","nombre":"Al│

│berto Manuel","edad":"42","identificador":"alb"} │

├──────────────────────────────────────────────────────────────────────┤

│{"apellidos":"Garcia Gonzalez","correo":"garciaenol@uniovi.es","nombre│

│":"Enol","edad":"25","identificador":"enol"} │

├──────────────────────────────────────────────────────────────────────┤

│{"apellidos":"Fernandez Fernandez","correo":"jose@uma.es","nombre":"Jo│

│se","edad":"58","identificador":"jose"} │

└──────────────────────────────────────────────────────────────────────┘

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Avanzadas
2. Devolver el nombre de los grados cuyo director tiene exactamente 40 años y además dicho grado contiene asignaturas de la rama de ingenieria

MATCH (g:Grado)

WHERE EXISTS{

MATCH (g)<-[:DIRIGE]-(p:Persona)

WHERE p.edad=40 AND EXISTS {

MATCH (g)-[:CONTIENE]->(a:Asignatura)

where a.rama="Ingenieria"

}

}

RETURN g

╒═══════════════════════════════════════════════════════════╕

│"g" │

╞══════════════════════════════════════════════════════════════════════╡

│{"localizacion":"Oviedo","nombre":"Ingenieria del software","facultad"│

│:"Escuela de Ingenieria Informatica"} │

├──────────────────────────────────────────────────────────────────────┤

│{"localizacion":"Barcelona","nombre":"Ingenieria Informatica","faculta│

│d":"Escuela politecnica de Barcelona"} │

└──────────────────────────────────────────────────────────────────────┘

Captura de pantalla de un celular

Descripción generada automáticamente

1. Obtener el camino más corto entre el director Dario y el alumno Sergio junto con el número de nodos intermedios (sin contar a los nodos origen y destino)

MATCH (a:Persona{nombre:"Dario"}),(b:Persona{nombre:"Sergio"}), p = shortestPath((a)-[:DIRIGE|CONTIENE|ESTA\_APUNTADO\*..4]-(b))

RETURN p,SIZE(NODES(p))-2;

Texto

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

# Aplicación en pyhton para las consultas

Requerimientos:

$> pip install neo4j

Código Python (3.11.0):

from neo4j import GraphDatabase

import logging

from neo4j.exceptions import ServiceUnavailable

class App:

def \_\_init\_\_(self, uri, user, password):

self.driver = GraphDatabase.driver(uri, auth=(user, password))

def close(self):

# Don't forget to close the driver connection when you are finished with it

self.driver.close()

def consultas(self):

with self.driver.session(database="neo4j") as session:

#------------------------Apartado a)-----------------------------

result = session.execute\_read(self.consultaSencilla1) # consulta 1

print("Consulta Sencilla 1: ",result,"\n")

result = session.execute\_read(self.consultaSencilla2) # consulta 2

print("Consulta Sencilla 2: ",result,"\n")

#------------------------Apartado b)-----------------------------

result = session.execute\_read(self.consultaIntermedia1) # consulta 1

print("Consulta Intermedia 1: ",result,"\n")

result = session.execute\_read(self.consultaIntermedia2) # consulta 2

print("Consulta Intermedia 2: ",result,"\n")

#------------------------Apartado c)-----------------------------

result = session.execute\_read(self.consultaAvanzada1) # consulta 1

print("Consulta Avanzada 1: ",result,"\n")

result, count = session.execute\_read(self.consultaAvanzada2) # consulta 2

print("Consulta Avanzada 2: \n $> Camino: ",result, "\n $> Nodos intermedios: ", count,"\n")

def procesarNodos(nodes, nodo):

res = []

for x in nodes:

res.append(x.data()[nodo])

return res

@staticmethod

def consultaSencilla1(tx):

query = (

"match (a: Asignatura) <-[:CONTIENE]- (:Grado{nombre:\"Ingenieria Informatica\"}) "

"return a"

)

nodes = tx.run(query)

return App.procesarNodos(nodes, "a")

@staticmethod

def consultaSencilla2(tx):

query = (

"match (p:Persona)-[:ESTA\_APUNTADO]->(a:Asignatura{nombre:\"Algebra\"}) "

"return p"

)

nodes = tx.run(query)

return App.procesarNodos(nodes, "p")

@staticmethod

def consultaIntermedia1(tx):

query = (

"match (a: Asignatura) <- [:CONTIENE] - (g:Grado) "

"where g.localizacion <> \"Oviedo\" and a.nombre = \"Calculo\" "

"match (a) <- [:ESTA\_APUNTADO] - (p:Persona) "

"return distinct(p) "

)

nodes = tx.run(query)

return App.procesarNodos(nodes, "p")

@staticmethod

def consultaIntermedia2(tx):

query = (

"MATCH (p:Persona)-[:IMPARTE]->(s:Asignatura) "

"OPTIONAL MATCH (p)-[:DIRIGE]->(g:Grado) "

"WITH p, COUNT(g) AS contador "

"WHERE contador = 0 "

"RETURN p"

)

nodes = tx.run(query)

return App.procesarNodos(nodes, "p")

@staticmethod

def consultaAvanzada1(tx):

query = (

"MATCH (g:Grado) "

"WHERE EXISTS{ "

"MATCH (g)<-[:DIRIGE]-(p:Persona) "

"WHERE p.edad=40 AND EXISTS { "

"MATCH (g)-[:CONTIENE]->(a:Asignatura) "

"where a.rama=\"Ingenieria\" "

"} "

"} "

"RETURN g"

)

nodes = tx.run(query)

return App.procesarNodos(nodes, "g")

@staticmethod

def consultaAvanzada2(tx):

query = (

"MATCH (a:Persona{nombre:\"Dario\"}),(b:Persona{nombre:\"Sergio\"}), p = shortestPath((a)-[:DIRIGE|CONTIENE|ESTA\_APUNTADO\*..4]-(b))"

"RETURN p, SIZE(NODES(p))-2;"

)

return tx.run(query).values()[0][0], tx.run(query).values()[0][1]

if \_\_name\_\_ == "\_\_main\_\_":

# Aura queries use an encrypted connection using the "neo4j+s" URI scheme

uri = "neo4j+s://c4d04a65.databases.neo4j.io"

user = "neo4j"

password = "iS6unODm5uW\_20BFiogMwGR93P\_M4kcTxKNyKNz0\_cw"

app = App(uri, user, password)

app.consultas() # aquí se ejecutan las consultas

app.close()

Pantalla de un video juego

Descripción generada automáticamente

 