**Fundamentos de Arquitectura de Software**

**Departamento de Ciencias de la Computación y la Decisión**

**Facultad de Minas**

**Universidad Nacional de Colombia**

Taller Arquitectura de la Información

*Observación: Cada vez que agregue nuevos elementos al documento, o que modifique algún componente del informe, revise la coherencia y consistencia con los otros elementos que hacen parte del mismo.*

**Responsables**

|  |
| --- |
| Nombre Completo |
| Jhojan Felipe Beltrán López |
| Julian Alejandro Rojas Betancur |
| Juan David Ruiz Echavarría |
| Jean Paul Yepes |
| Daniel Andrés Toro |

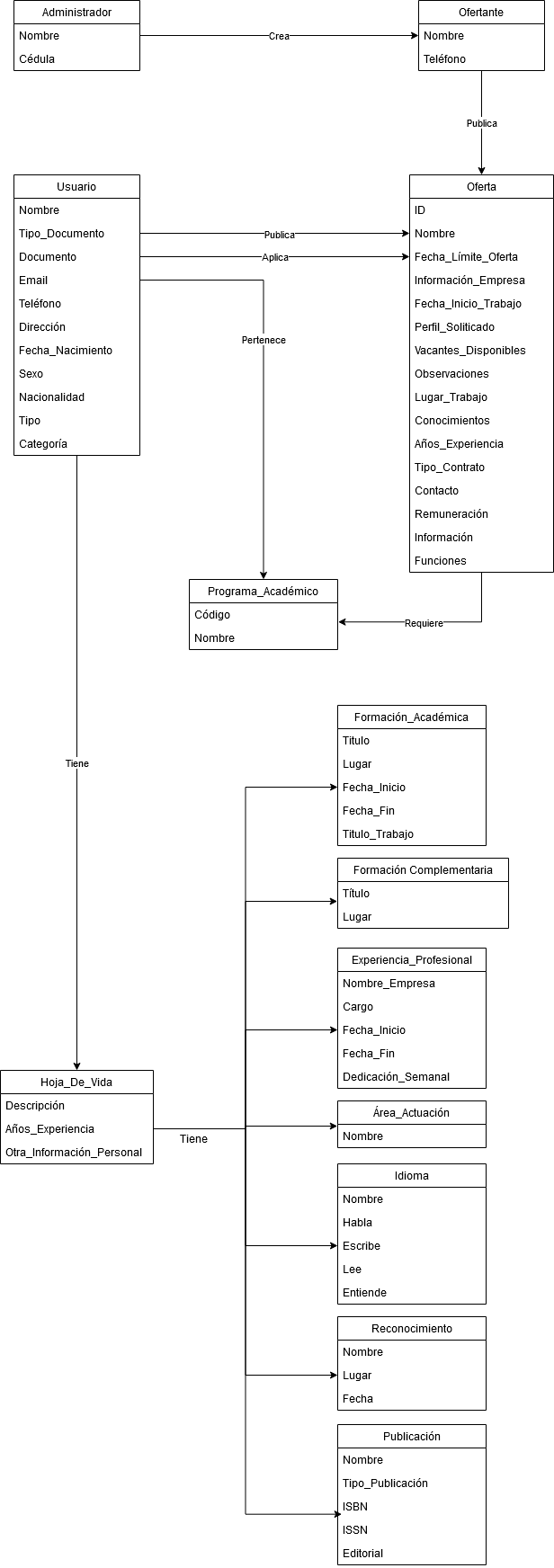
Los siguientes puntos se realizan considerando el conjunto de datos que se defina para la solución del proyecto específico, es decir, respecto al modelo del dominio.

**Repositorio del proyecto**

En el siguiente enlace podrá encontrar todos los diagrama realizados, tanto en archivo .drawio como en imagen <https://github.com/JulianRojasBet/hv_estudiantes>

1. **Modelo del dominio**

***Observación****: Incluya el gráfico del modelo del dominio que representa la estructura de su problema.*

**

1. **Modelo Entidad-Relación**
   1. **Sentencia o consulta de creación del tabla(s)**

***Observación****: Escriba el código en el Sistema de Gestión de Bases de Datos Relacionales de su elección para crear la tabla que corresponda con su conjunto de datos específico.*

BEGIN;

--

-- Create model Usuario

--

CREATE TABLE "core\_usuario" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "documento" varchar(15) NOT NULL, "nombre" varchar(70) NOT NULL, "email" varchar(254) NOT NULL, "telefono" varchar(13) NOT NULL, "direccion" varchar(50) NOT NULL, "fecha\_nacimiento" date NOT NULL, "sexo" varchar(10) NOT NULL, "nacionalidad" varchar(30) NOT NULL);

--

-- Create model Reconocimiento

--

CREATE TABLE "core\_reconocimiento" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombre" text NOT NULL, "lugar" varchar(50) NOT NULL, "fecha" date NOT NULL, "usuario\_id" integer NOT NULL REFERENCES "core\_usuario" ("id") DEFERRABLE INITIALLY DEFERRED);

--

-- Create model Publicacion

--

CREATE TABLE "core\_publicacion" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombre" text NOT NULL, "titulo\_publicacion" text NOT NULL, "isbn" varchar(13) NOT NULL, "issn" varchar(8) NOT NULL, "editorial" varchar(50) NOT NULL, "usuario\_id" integer NOT NULL REFERENCES "core\_usuario" ("id") DEFERRABLE INITIALLY DEFERRED);

--

-- Create model Idioma

--

CREATE TABLE "core\_idioma" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombre" varchar(15) NOT NULL, "habla" real NOT NULL, "escribe" real NOT NULL, "lee" real NOT NULL, "escucha" real NOT NULL, "usuario\_id" integer NOT NULL REFERENCES "core\_usuario" ("id") DEFERRABLE INITIALLY DEFERRED);

--

-- Create model HojaVida

--

CREATE TABLE "core\_hojavida" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "descripcion" text NOT NULL, "experiencia" real NOT NULL, "otra\_info" text NOT NULL, "usuario\_id" integer NOT NULL UNIQUE REFERENCES "core\_usuario" ("id") DEFERRABLE INITIALLY DEFERRED);

--

-- Create model FormacionComplementaria

--

CREATE TABLE "core\_formacioncomplementaria" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "titulo" varchar(50) NOT NULL, "lugar" text NOT NULL, "usuario\_id" integer NOT NULL REFERENCES "core\_usuario" ("id") DEFERRABLE INITIALLY DEFERRED);

--

-- Create model FormacionAcademica

--

CREATE TABLE "core\_formacionacademica" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "titulo" varchar(50) NOT NULL, "lugar" varchar(50) NOT NULL, "fecha\_inicio" date NOT NULL, "fecha\_fin" date NOT NULL, "titulo\_trabajo" text NOT NULL, "usuario\_id" integer NOT NULL REFERENCES "core\_usuario" ("id") DEFERRABLE INITIALLY DEFERRED);

--

-- Create model ExperienciaProfesional

--

CREATE TABLE "core\_experienciaprofesional" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombre\_empresa" varchar(50) NOT NULL, "cargo" varchar(50) NOT NULL, "fecha\_inicio" date NOT NULL, "fecha\_fin" date NOT NULL, "dedicacion\_semanal" real NOT NULL, "usuario\_id" integer NOT NULL REFERENCES "core\_usuario" ("id") DEFERRABLE INITIALLY DEFERRED);

--

-- Create model AreaActuacion

--

CREATE TABLE "core\_areaactuacion" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombre" varchar(50) NOT NULL, "usuario\_id" integer NOT NULL REFERENCES "core\_usuario" ("id") DEFERRABLE INITIALLY DEFERRED);

CREATE INDEX "core\_reconocimiento\_usuario\_id\_e7af145f" ON "core\_reconocimiento" ("usuario\_id");

CREATE INDEX "core\_publicacion\_usuario\_id\_c5eecb05" ON "core\_publicacion" ("usuario\_id");

CREATE INDEX "core\_idioma\_usuario\_id\_b0767e29" ON "core\_idioma" ("usuario\_id");

CREATE INDEX "core\_formacioncomplementaria\_usuario\_id\_fe6a2c9b" ON "core\_formacioncomplementaria" ("usuario\_id");

CREATE INDEX "core\_formacionacademica\_usuario\_id\_937df8d2" ON "core\_formacionacademica" ("usuario\_id");

CREATE INDEX "core\_experienciaprofesional\_usuario\_id\_378e4bff" ON "core\_experienciaprofesional" ("usuario\_id");

CREATE INDEX "core\_areaactuacion\_usuario\_id\_79b00cbe" ON "core\_areaactuacion" ("usuario\_id");

COMMIT;

* 1. **Sentencias para Insertar datos**

***Observación****: Escriba el código para insertar al menos 5 elementos en la tabla(s) creadas en el punto 2.1.*

INSERT INTO core\_usuario(documento, nombre, email, telefono, direccion, fecha\_nacimiento, sexo, nacionalidad)

VALUES ("12345", "Benito", "benito@correo.com", "321000000", "cra 55 #12", "26/12/58", "Masculino", "Colombia");

INSERT INTO core\_usuario(documento, nombre, email, telefono, direccion, fecha\_nacimiento, sexo, nacionalidad)

VALUES ("54321", "Esteban", "esteban@correo.com", "314045123", "avenida 33 - calle 12", "23/11/92", "Masculino", "Mexico");

INSERT INTO core\_idioma(nombre, habla, escribe, lee, escucha, usuario\_id)

VALUES ("Ingles", 3, 5, 5, 4, 1);

INSERT INTO core\_idioma(nombre, habla, escribe, lee, escucha, usuario\_id)

VALUES ("Italiano", 2, 3, 3, 2, 2);

INSERT INTO core\_hojavida(descripcion, experiencia, otra\_info, usuario\_id)

VALUES ("Soy interesante", 1, "otra info", 1);

INSERT INTO core\_hojavida(descripcion, experiencia, otra\_info, usuario\_id)

VALUES ("I am great", 3, "otra info", 2);

INSERT INTO core\_formacioncomplementaria(titulo, lugar, usuario\_id)

VALUES ("Ing. Agricola", "UNAL", 1);

INSERT INTO core\_formacioncomplementaria(titulo, lugar, usuario\_id)

VALUES ("Ing. Sistemas", "UDEA", 2);

INSERT INTO core\_experienciaprofesional(nombre\_empresa, cargo, fecha\_inicio, fecha\_fin, dedicacion\_semanal, usuario\_id)

VALUES ("Colanta", "Jefe taller campo", '05/05/2019', '05/05/2020', '2', 1);

INSERT INTO core\_experienciaprofesional(nombre\_empresa, cargo, fecha\_inicio, fecha\_fin, dedicacion\_semanal, usuario\_id)

VALUES ("Colanta", "Jefe taller campo", '05/05/2019', '05/05/2020', '2', 1);

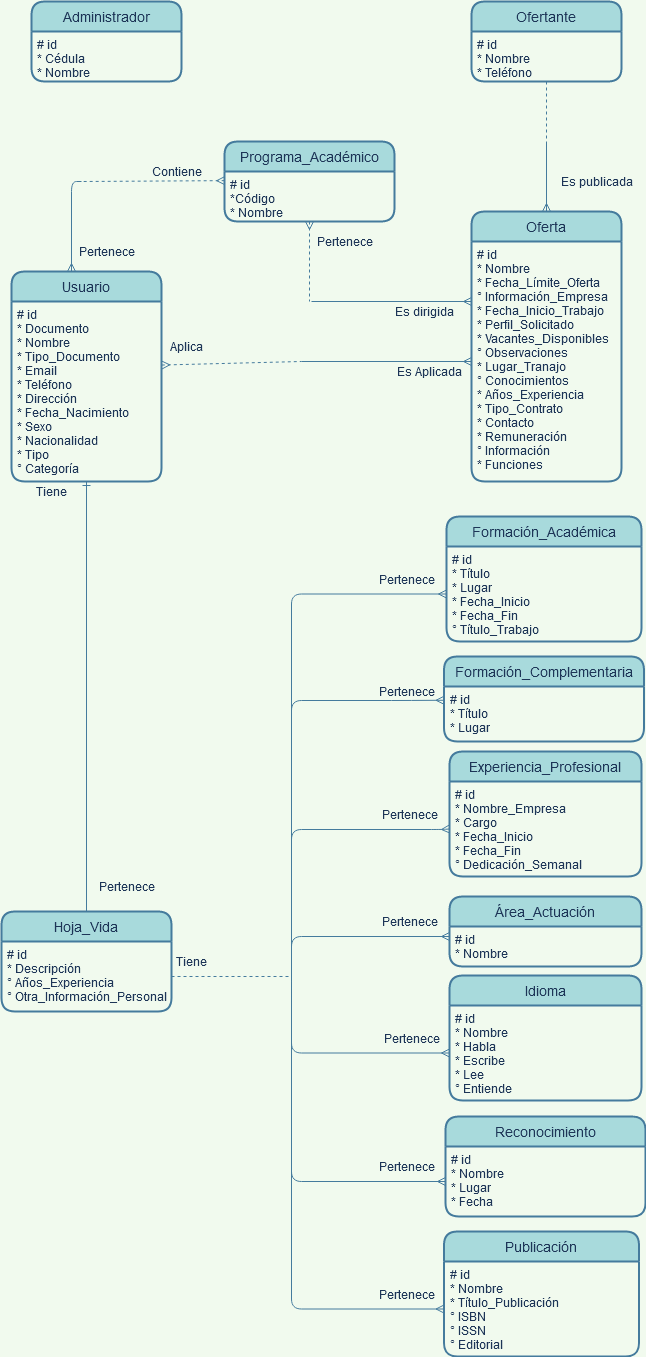
* 1. **Sentencia de consulta**

***Observación****: Escriba el código para consultar solo tres atributos específicos de los insertados en la sección 2.2.*

SELECT nombre, documento, email FROM core\_usuario WHERE nacionalidad = 'Colombia';

SELECT habla, lee, escribe FROM core\_idioma WHERE usuario\_id=1;

* 1. **Toma de pantalla del modelo E-R**



1. **MongoDB**
   1. **Sentencia o consulta de creación del documento(s)**

***Observación****: Escriba el código en MongoDB para crear al menos cuatro documentos que correspondan a su conjunto de datos específico.*

use bdhv

db.usuarios.insert({

"nombre": "Diego",

"tipo\_documento": "CC",

"documento": 1234567890,

"numero\_telefonico": 3123456789,

"email": "diegomail@gmail.com",

"contraseña": "Diego123"

})

db.usuarios.insert({

"nombre": "Ana",

"tipo\_documento": "CC",

"documento": 1134567890,

"numero\_telefonico": 3323456789,

"email": "anamail@gmail.com",

"contraseña": "Ana123"

})

db.usuarios.insert({

"nombre": "Claudia",

"tipo\_documento": "CC",

"documento": 1114567890,

"numero\_telefonico": 333456789,

"email": "claudiamail@gmail.com",

"contraseña": "Claudia123"

})

db.usuarios.insert({

"nombre": "Juan",

"tipo\_documento": "CC",

"documento": 1111567890,

"numero\_telefonico": 3333456789,

"email": "juanmail@gmail.com",

"contraseña": "Juan123"

})

db.hojaDeVida.insert({

"documento": 1234567890,

"descripcion": "texto",

"añosExperiencia": 0,

"otraInformacionPersonal": "texto"

})

db.hojaDeVida.insert({

"documento": 1134567890,

"descripcion": "texto",

"añosExperiencia": 2,

"otraInformacionPersonal": "texto"

})

db.hojaDeVida.insert({

"documento": 1114567890,

"descripcion": "texto",

"añosExperiencia": 5,

"otraInformacionPersonal": "texto"

})

db.hojaDeVida.insert({

"documento": 1111567890,

"descripcion": "texto",

"añosExperiencia": 10,

"otraInformacionPersonal": "texto"

})

db.formacionAcademica.insert({

"documento": 1234567890,

"lugar": "universidad nacional",

fechaInicio: new Date("2012, 12, 21"),

fechaFin: new Date("2017, 12, 21"),

"titulo": "Ingeniero de sistemas"

})

db.oferta.insert({

"id": "IngSis123",

"nombre": "Programador junior",

"fecha\_limite": new Date("2020, 12, 21"),

"infomacion\_empresa": "explotadora, se trabaja domingos hasta las 10",

fecha\_inicio\_trabajo: new Date("2020, 12, 24"),

"perfil\_solicitado": "estudiante noveno semestre",

"vacantes\_disponibles": 1,

"observaciones": "salario minimo",

"lugar\_trabajo": "industriales medellin",

"conocimientos": "full stack",

"años\_experiencia": 10,

"tipo\_contrato": "a 6 meses",

"contacto": 5123456,

"remuneracion": 900000,

"informacion": "texto",

"funciones": "programacion"

})

db.administrador.insert({

"nombre": "dbadmin",

"cedula": 8123456

})

* 1. **Sentencia de consulta**

***Observación****: Escriba el código en mongo para consultar 2 documentos (use limit) y muestre dos atributos específicos que usted considere, de los creados en el punto 1.1.*

db.hojaDeVida.find({"añosExperiencia":{$gt:4}}, {descripcion:0, otraInformacionPersonal:0, \_id:0}).limit(2).pretty();

1. **Hypertable**
   1. **Sentencia o consulta de creación del tabla(s)**

***Observación****: Escriba el código en hypertable para crear la tabla que corresponda con su conjunto de datos específico.*

CREATE TABLE Administrador (

nombre,

email,

contrasenia

);

CREATE TABLE Ofertante (

nombre,

email,

contrasenia,

telefono

);

CREATE TABLE Oferta(

id\_ofertante,

id\_programa\_academico,

nombre,

fecha\_limite\_oferta,

informacion\_empresa,

fecha\_inicio\_trabajo,

perfil\_solicitado,

vacantes\_disponibles,

observaciones,

lugar\_trabajo,

conocimientos,

anios\_experiencia,

tipo\_contrato,

contacto,

remuneracion,

informacion,

funciones

);

CREATE TABLE Aplicacion (

id\_usuario,

id\_oferta

);

CREATE TABLE Usuario(

id\_usuario,

id\_programa\_academico,

correo,

contrasenia,

nombre,

tipo\_documento,

documento,

email,

telefono,

direccion,

fecha\_nacimiento,

sexo,

nacionalidad,

tipo,

categoria

);

CREATE TABLE Programa\_academico (

nombre

);

CREATE TABLE Hoja\_de\_vida (

id\_usuario,

descripcion,

anios\_experiencia,

otra\_info\_personal

);

CREATE TABLE Formacion\_academica(

id\_hoja\_de\_vida,

titulo,

lugar,

fecha\_inicio,

fecha\_fin,

titulo\_trabajo

);

CREATE TABLE Formacion\_complementaria(

id\_hoja\_de\_vida,

titulo,

lugar

);

CREATE TABLE Experiencia\_profesional(

id\_hoja\_de\_vida,

nombre\_empresa,

cargo,

fecha\_inicio,

fecha\_fin,

dedicacion\_semanal

);

CREATE TABLE Area\_actuacion(

id\_hoja\_de\_vida,

nombre

);

CREATE TABLE Idioma (

id\_hoja\_de\_vida,

nombre,

habla,

escribe,

lee,

entiende

);

CREATE TABLE Reconocimiento (

id\_hoja\_de\_vida,

nombre,

lugar,

fecha

);

CREATE TABLE Publicacion(

id\_hoja\_de\_vida,

nombre,

tipo\_publicacion,

isbn,

issn,

editorial

);

* 1. **Sentencias para Insertar datos**

***Observación****: Escriba el código en hypertable para insertar al menos 5 elementos en la tabla(s) creadas en el punto 2.1.*

INSERT INTO Administrador VALUES ("1", "nombre", "admin"), ("1", "email", "admin@admin.com"), ("1", "contrasenia", "qwerty");

INSERT INTO Ofertante VALUES ("1", "nombre", "Banco Chibchombia"), ("1", "email", "reclutador@banco.com"), ("1", "contrasenia", "123456789");

INSERT INTO Programa\_academico VALUES ("1", "nombre", "Ingenieria de sistemas");

INSERT INTO Oferta VALUES ("1", "id\_ofertante", "1"),

("1", "id\_programa\_academico", "1"),

("1", "nombre", "Practicante Ingenieria de Sistemas"),

("1", "fecha\_limite\_oferta", "2020-12-1"),

("1", "informacion\_empresa", "Somos Banco Chibchombia, le estamos poniendo el alma"),

("1", "fecha\_inicio\_trabajo", "2021-2-1"),

("1", "perfil\_solicitado", "Buscamos ingenieria de sistemas con 20 años de experiencia que sepa todos los lenguajes de programacion y tecnologias existentes"),

("1", "vacantes\_disponibles", "2"),

("1", "observaciones", "Con vision y liderazgo"),

("1", "lugar\_trabajo", "Medellin, Antioquia"),

("1", "conocimientos", "Javascript, XML, JSON, CSS, HTML, mongodb, python, SQL, fortran, C, C++, C#, .net, docker, hadoop, git, vim, android, ios, kotlin, java"),

("1", "anios\_experiencia", "20"),

("1", "tipo\_contrato", "Practicante"),

("1", "contacto", "reclutador@banco.com"),

("1", "remuneracion", "1 SLMV"),

("1", "informacion", "Somos una compania con gran cultura"),

("1", "funciones", "Lo que surga");

INSERT INTO Aplicacion VALUES ("1", "id\_usuario", "1"), ("1", "id\_oferta", "1");

* 1. **Sentencia de consulta**

***Observación****: Escriba el código en hypertable consultar solo tres atributos específicos de los insertados en la sección 2.2.*

SELECT nombre, email, contrasenia FROM Administrador WHERE row = "1";

SELECT nombre, email, telefono FROM Ofertante WHERE row = "1";

SELECT nombre, remuneracion, funciones FROM Oferta WHERE row = "1";

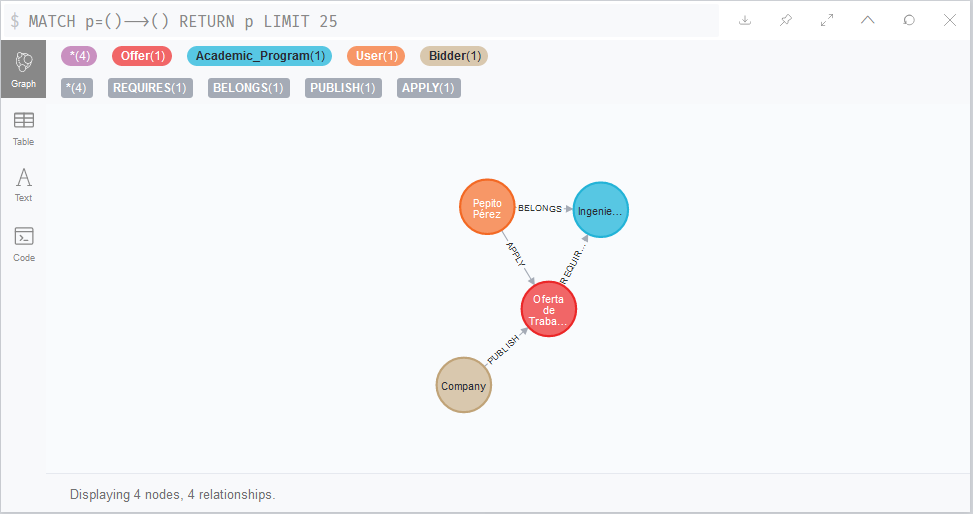
1. **Neo4j**
   1. **Sentencia o consulta de creación del Nodo(s) y Relaciones**

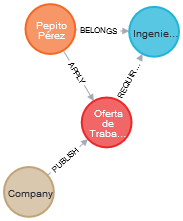
***Observación****: Escriba el código en Neo4J para crear el (los) nodo(s) y relaciones que corresponda(n) con su conjunto de datos específico. Debe contener al menos 4 relaciones y 3 nodos.*

create (Pepito:User {id: 0, name: 'Pepito Pérez', tDocument: 'CC', document: '1025983341', email: 'peperez@unal.edu.co', phone: '3106923304', address: 'Cra 56B No. 80-47', dateBirthday: '1995-12-14', sex: 'Masculine', nacionality: 'Colombian', type: 'Egresado'}), (Sistemas:Academic\_Program {id: 0, code: 5820, name: 'Ingeniería de Sistemas e Informática'}), (Offer:Offer {id: 0, name: 'Oferta de Trabajo', limitDate: '2020-11-15', startDate: '2020-12-01', perfil: 'Se solicita ingeniero de sistemas para programar en Python', vacancies: 2, observations: 'Debe saber python y conocer la metodología SCRUM', placeWork: 'Medellín', knowledge: 'Python', yearsExperience: 0, contractType: 'Indefinido', contact: 'companyX@cp.com.co', remuneration: 2500000, information: 'Se pagan prestaciones', functions: 'Desarrollar, Prestar apoyo al área técnica, Hacer pruebas unitarias de funcionalidades implementadas' }), (Company:Bidder {id: 0, name: 'Company X', phone: '4440089' }) ,(Pepito)-[:BELONGS]->(Sistemas), (Pepito)-[:APPLY]->(Offer), (Offer)-[:REQUIRES]->(Sistemas),(Company)-[:PUBLISH]->(Offer)

* 1. **Grafo generado**

***Observación****: inserte la toma de pantalla del grafo generado.*

******

******

1. **Análisis de lectura**

***Observación****: Considerando el artículo: “A Framework for Migrating Relational Datasets to NoSQL” de Leonardo Rocha et al. (2015). Analice y responda concretamente:*

1. ¿Cuáles son las limitaciones, que se pueden inferir de la lectura, para migrar los conjuntos de datos relacionales a NoSQL?
2. ¿Cuáles limitaciones considera que se deben considerar, a parte de las mencionadas en el artículo?
3. ¿Cuáles son las razones (criterios) que se deben considerar para migrar un conjunto de datos relacionados a NoSQL?

Para llevar a cabo la migración de datos relacionales, a una estructura no relacional, se debe tener en cuenta la cantidad de datos que se maneja y el acceso eficiente y óptimo que se tiene a cada registro que hace parte de ese conjunto de datos, es decir, se debe tener en cuenta el volumen de datos a migrar, que exista la necesidad de datos redundantes para mejorar los tiempo de consulta a los datos y que pueden estar disponibles para ser usados en cualquier momento. Todo ello permite que las aplicaciones se vuelvan escalables y cuenten con una alta disponibilidad para hacer uso de los diferentes datos.