

UNIVERSIDAD POLITECNICA SALESIANA

Alumna: Narcisa Araujo

Examen Final IA

Enunciado:

1. Desarrollar un juego (tema libre) empleando una de las 2 siguientes alternativas:
2. easyAI
3. Universe + GYM El juego deberá implementar algún algoritmo de IA y de igual forma, generar un informe de movimientos, puntajes y quién gana la partida. Se debe tener un juego en donde se tenga un jugador humano y otro utilizando Inteligencia Artificial, finalmente no se puede repetir el juego por más de tres personas por lo que se debe publicar en el foro el juego seleccionado.

```

In [1]: from tkinter import *
import tkinter.font as tkFont
import gym
from neo4j import GraphDatabase

def opciones(self):
    usr=self
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')

    root1 = Tk()
    root1.title("Examen Final-IA")
    root1.resizable(1,1)

    frame = Frame(root1, width=480, height=320)
    frame.pack(fill='both', expand=1)
    frame.config(bd=50)
    frame.config(bg="pink")

    def juego():
        def OpenAI_Gym(game,num_episodios,num_inter):
            env = gym.make(game)
            puntos=""
            for i_episode in range(num_episodios):
                env.reset()
                recompensa=0
                for t in range(num_inter):
                    env.render()
                    action = env.action_space.sample()
                    observation,reward,done,info = env.step(action)
                    recompensa+=reward

                    if done:
                        recompensa += reward
                        print("puntaje obtenido:",recompensa)
                        break
            puntos=str(recompensa)
            env.close()
            with neo4j._driver.session() as session:
                session.write_transaction(neo4j.crear_puntaje,puntos)
                session.write_transaction(neo4j.crear_relacion_usuario_puntaje,usr)
        OpenAI_Gym(game= 'CartPole-v0', num_episodios=5,num_inter= 1000)

    def recomendacion():

        vjuego = Frame(frame, width=480, height=420)
        vjuego.grid(row=0, column=2, padx=10, pady=3, ipadx=20, ipady=20)
        fontStyle = tkFont.Font(family="Lucida Grande", size=12)

        tipo=Label(vjuego,text=' Lugares Turisticos Guayaquil ', font=fontStyle)
        tipo.place(x=50,y=10)

# Lugares turisticos recomendados

#Parques Guayaquil
def par1():
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')

```

```

        with neo4j._driver.session() as session:
            session.read_transaction(neo4j.buscar1,vjuego)
#Iglesias de Guayaquil
        def par2():
            neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')
            with neo4j._driver.session() as session:
                session.read_transaction(neo4j.buscar2,vjuego)

#Hoteles de Guayaquil
        def par3():
            neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')
            with neo4j._driver.session() as session:
                session.read_transaction(neo4j.buscar3,vjuego)

#Lugares Turisticos de Guayaquil
        def par4():
            neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')
            with neo4j._driver.session() as session:
                session.read_transaction(neo4j.buscar4,vjuego)

#Boton Parques recomendados
        btn1 = Button(vjuego, text="Parques", bg="salmon", fg="black", height = 2)
#Boton Iglesias recomendados
        btn2 = Button(vjuego, text="Iglesias", bg="salmon", fg="black", height = 2)
#Boton Hoteles recomendados
        btn3 = Button(vjuego, text="Hoteles", bg="salmon", fg="black", height = 2)
#Boton Lugares Turisticos recomendados
        btn4 = Button(vjuego, text="Lugares Turisticos", bg="salmon", fg="black", height = 2)

        btn1.place(x=50, y=80)
        btn2.place(x=50, y=130)
        btn3.place(x=50, y=180)
        btn4.place(x=50, y=230)

#Boton Iniciar juego
        btn5 = Button(frame, text="Jugar", bg="salmon", fg="black", height = 2, width = 10)
#Boton ir a recomendaciones
        btn6 = Button(frame, text="Recomendaciones", bg="salmon", fg="black", height = 2, width = 10)

        btn5.grid(column=1, row=0)
        btn6.grid(column=2, row=0)

        root1.config(bd=15)
        root1.config(bg="pink")
# Finalmente bucle de la aplicación
        root1.mainloop()

```

```

In [2]: root = Tk()
root.title("ExamenFinal-IA")
root.geometry('450x400')
root.resizable(width=0, height=0)
fontStyle = tkFont.Font(family="Lucida Grande", size=12)
fontStyle1 = tkFont.Font(family="Lucida Grande", size=25)
fontStyle2 = tkFont.Font(family="Lucida Grande", size=15)

eusuario=Label(root,text='Registro Usuarios', font=fontStyle1)
eusuario.place(x=150,y=20)

eusuario=Label(root,text='Usuario', font=fontStyle)
eusuario.place(x=50,y=100)
usuario = Entry(root, fg='black', bg='white', justify='center')
usuario.place(x=110, y=100, width=200)

edad=Label(root,text='Edad: ', font=fontStyle)
edad.place(x=50,y=150)
edad = Entry(root, fg='black', bg='white', justify='center')
edad.place(x=110, y=150, width=100)

class Neo4jService(object):

    def __init__(self, uri, user, password):
        self._driver = GraphDatabase.driver(uri, auth=(user, password))

    def close(self):
        self._driver.close()

    def crear_usuario(self, tx, nombre):
        tx.run("CREATE (Usuario:Jugador {usuario: $nombre})", nombre=nombre)

    def crear_usuarioj(self, tx, usr):
        tx.run("CREATE (Usr:Jugador {nombre: $usr})", usr=usr)

    def crear_edad(self, tx, edad):
        tx.run("CREATE (Edad:Jugador {edad: $edad})", edad=edad)

    def crear_puntaje(self, tx, puntaje):
        tx.run("CREATE (Puntaje:Jugador {puntaje: $puntaje})", puntaje=puntaje)

    def crear_relacion_usuario_usr(self, tx, usuario, usr):
        tx.run("MATCH (a:Jugador {usuario: $usuario}) "
               "MATCH (b:Jugador {nombre: $usr}) "
               "MERGE (a)-[r:relacion]->(b)",
               usuario=usuario, usr=usr)

    def crear_relacion_usuario_edad(self, tx, usuario, edad):
        tx.run("MATCH (a:Jugador {nombre: $usuario}) "
               "MATCH (b:Jugador {edad: $edad}) "
               "MERGE (a)-[r:relacion]->(b)",
               usuario=usuario, edad=edad)

    def crear_relacion_usuario_puntaje(self, tx, usuario, puntaje):

```

```

tx.run("MATCH (a:Jugador {nombre: $usuario}) "
      "MATCH (b:Jugador {puntaje: $puntaje}) "
      "MERGE (a)-[r:relacion]->(b)",
      usuario=usuario, puntaje=puntaje)

```

ALGORITMO DE BUSQUEDA DE LA RUTA MAS CORTA

```

def ruta_mas_corta(self,tx,a,vjuego):
    result = tx.run("MATCH (start:Station {name: 'Iglesia la Merced'}), (end:
                    \"CALL gds.alpha.shortestPath.stream({\n\"
                    \"nodeProjection: 'Station',\n\"
                    \"relationshipProjection: {\n\"
                    \"CONNECTION: {\n\"
                    \"type: 'CONNECTION',\n\"
                    \"properties: 'time',\n\"
                    \"orientation: 'UNDIRECTED'\n\"
                    \"}\n\"
                    \"},\n\"
                    \"startNode: start,\n\"
                    \"endNode: end,\n\"
                    \"relationshipWeightProperty: 'time'\n\"
                    \"})\n\"
                    \"YIELD nodeId\n\"
                    \"RETURN gds.util.asNode(nodeId).name AS name\",destino=a)

etomof=Label(vjuego,text="La Ruta mas corta es:")
etomof.place(x=210,y=260)

```

```

frame1 = Frame(vjuego)
frame1.place(x=210,y=280)

```

```

scrollbar = Scrollbar(frame1)
scrollbar.pack( side = RIGHT, fill = Y )

```

```

mylist = Listbox(frame1, yscrollcommand = scrollbar.set )
for line in result:
    texto=str(line)
    tam=len(texto)-2
    mensajea = texto[14:tam]
    mylist.insert(END, mensajea)

```

```

mylist.pack( side = LEFT, fill = BOTH )
scrollbar.config( command = mylist.yview )

```

#Consulta de parques por su nombre

```

def buscar1(self,tx,vjuego):
    result = tx.run("MATCH(p:Station) WHERE p.name STARTS WITH $p RETURN p.name")
    etomo=Label(vjuego,text="Lugares Turisticos Guayaquil:")
    etomo.place(x=200,y=80)

```

```

frame1 = Frame(vjuego)
frame1.place(x=200,y=100)

```

```

scrollbar = Scrollbar(frame1)
scrollbar.pack( side = RIGHT, fill = Y )

```

```

mylist = Listbox(frame1, yscrollcommand = scrollbar.set )
for line in result:
    mensaje=str(line)
    tam=len(mensaje)-2
    mensajea = mensaje[16:tam]
    mylist.insert(END, mensajea)

mylist.pack( side = LEFT, fill = BOTH )
scrollbar.config( command = mylist.yview )
def select():
    a = str(mylist.get(str(mylist.curselection()[0])))
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')
    with neo4j._driver.session() as session:
        session.read_transaction(neo4j.ruta_mas_corta,a,vjuego)

btn7 = Button(vjuego, text="Seleccionar", bg="hotpink", fg="white", height=30)
btn7.place(x=400, y=200)

```

#Consulta de iglesias por su nombre

```

def buscar2(self,tx,vjuego):
    result = tx.run("MATCH(p:Station) WHERE p.name STARTS WITH $p RETURN p.name")
    etomo=Label(vjuego,text="Lugares Turisticos Guayaquil:")
    etomo.place(x=200,y=80)

```

```

frame1 = Frame(vjuego)
frame1.place(x=200,y=100)

```

```

scrollbar = Scrollbar(frame1)
scrollbar.pack( side = RIGHT, fill = Y )

```

```

mylist = Listbox(frame1, yscrollcommand = scrollbar.set )
for line in result:
    mensaje=str(line)
    tam=len(mensaje)-2
    mensajea = mensaje[16:tam]
    mylist.insert(END, mensajea)

```

```

mylist.pack( side = LEFT, fill = BOTH )
scrollbar.config( command = mylist.yview )

```

```

def select():
    a = str(mylist.get(str(mylist.curselection()[0])))
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')
    with neo4j._driver.session() as session:
        session.read_transaction(neo4j.ruta_mas_corta,a,vjuego)

```

```

btn7 = Button(vjuego, text="Seleccionar", bg="hotpink", fg="white", height=30)
btn7.place(x=400, y=200)

```

#Consulta de hoteles por su nombre

```

def buscar3(self,tx,vjuego):
    result = tx.run("MATCH(p:Station) WHERE p.name STARTS WITH $p RETURN p.name")
    etomo=Label(vjuego,text="Lugares Turisticos Guayaquil:")
    etomo.place(x=200,y=80)

```

```

frame1 = Frame(vjuego)
frame1.place(x=200,y=100)

```

```

scrollbar = Scrollbar(frame1)
scrollbar.pack( side = RIGHT, fill = Y )

mylist = Listbox(frame1, yscrollcommand = scrollbar.set )
for line in result:
    mensaje=str(line)
    tam=len(mensaje)-2
    mensajea = mensaje[16:tam]
    mylist.insert(END, mensajea)

mylist.pack( side = LEFT, fill = BOTH )
scrollbar.config( command = mylist.yview )

def select():
    a = str(mylist.get(str(mylist.curselection()[0])))
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')
    with neo4j._driver.session() as session:
        session.read_transaction(neo4j.ruta_mas_corta,a,vjuego)

btn7 = Button(vjuego, text="Seleccionar", bg="hotpink", fg="white", height=30)
btn7.place(x=400, y=200)

```

#Consulta de Lugares turisticos por su nombre

```

def buscar4(self,tx,vjuego):
    result = tx.run("MATCH(p:Station) WHERE p.name STARTS WITH $p RETURN p.name")
    etomo=Label(vjuego,text="Lugares Turisticos Guayaquil:")
    etomo.place(x=200,y=80)

```

```

frame1 = Frame(vjuego)
frame1.place(x=200,y=100)

```

```

scrollbar = Scrollbar(frame1)
scrollbar.pack( side = RIGHT, fill = Y )

```

```

mylist = Listbox(frame1, yscrollcommand = scrollbar.set )
for line in result:
    mensaje=str(line)
    tam=len(mensaje)-2
    mensajea = mensaje[16:tam]
    mylist.insert(END, mensajea)

```

```

mylist.pack( side = LEFT, fill = BOTH )
scrollbar.config( command = mylist.yview )

```

```

def select():
    a = str(mylist.get(str(mylist.curselection()[0])))
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')
    with neo4j._driver.session() as session:
        session.read_transaction(neo4j.ruta_mas_corta,a,vjuego)

```

```

btn7 = Button(vjuego, text="Seleccionar", bg="hotpink", fg="white", height=30)
btn7.place(x=400, y=200)

```

```

neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')

```

```

with neo4j._driver.session() as session:
    #session.write_transaction(neo4j.crear_usuario , "Usuario")

def cerrar():
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'egfa2209')
    usr=str(usuario.get())
    ed=str(edad.get())

    with neo4j._driver.session() as session:

        session.write_transaction(neo4j.crear_usuarioj , usr)
        session.write_transaction(neo4j.crear_edad , ed)

        session.write_transaction(neo4j.crear_relacion_usuario_usr,"Usuario",usr)
        session.write_transaction(neo4j.crear_relacion_usuario_edad,usr,ed)

    #root.destroy()
    opciones(usr)

# Boton de registro de usuario
btn1 = Button(root, text="Registro", bg="hotpink", fg="white", height = 2, width
btn1.place(x=120, y=300)

root.mainloop()

```

puntaje obtenido: 15.0
puntaje obtenido: 21.0
puntaje obtenido: 17.0
puntaje obtenido: 45.0
puntaje obtenido: 40.0

Registro Usuario

ExamenFinal-IA

Registro Usuarios

Usuario

Edad:

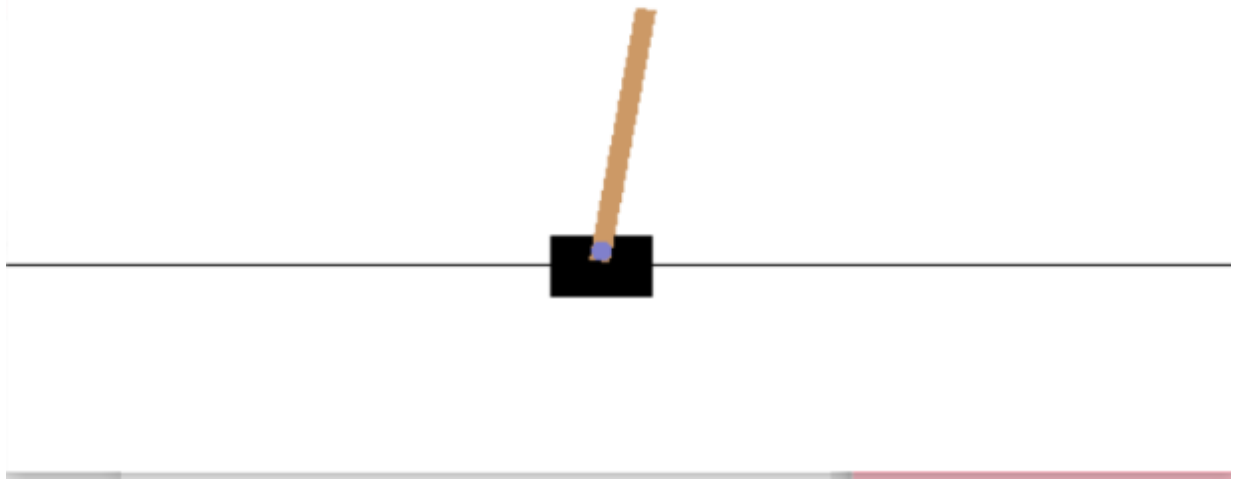
Registro

Ingreso a juego y recomendaciones

Examen Final-IA

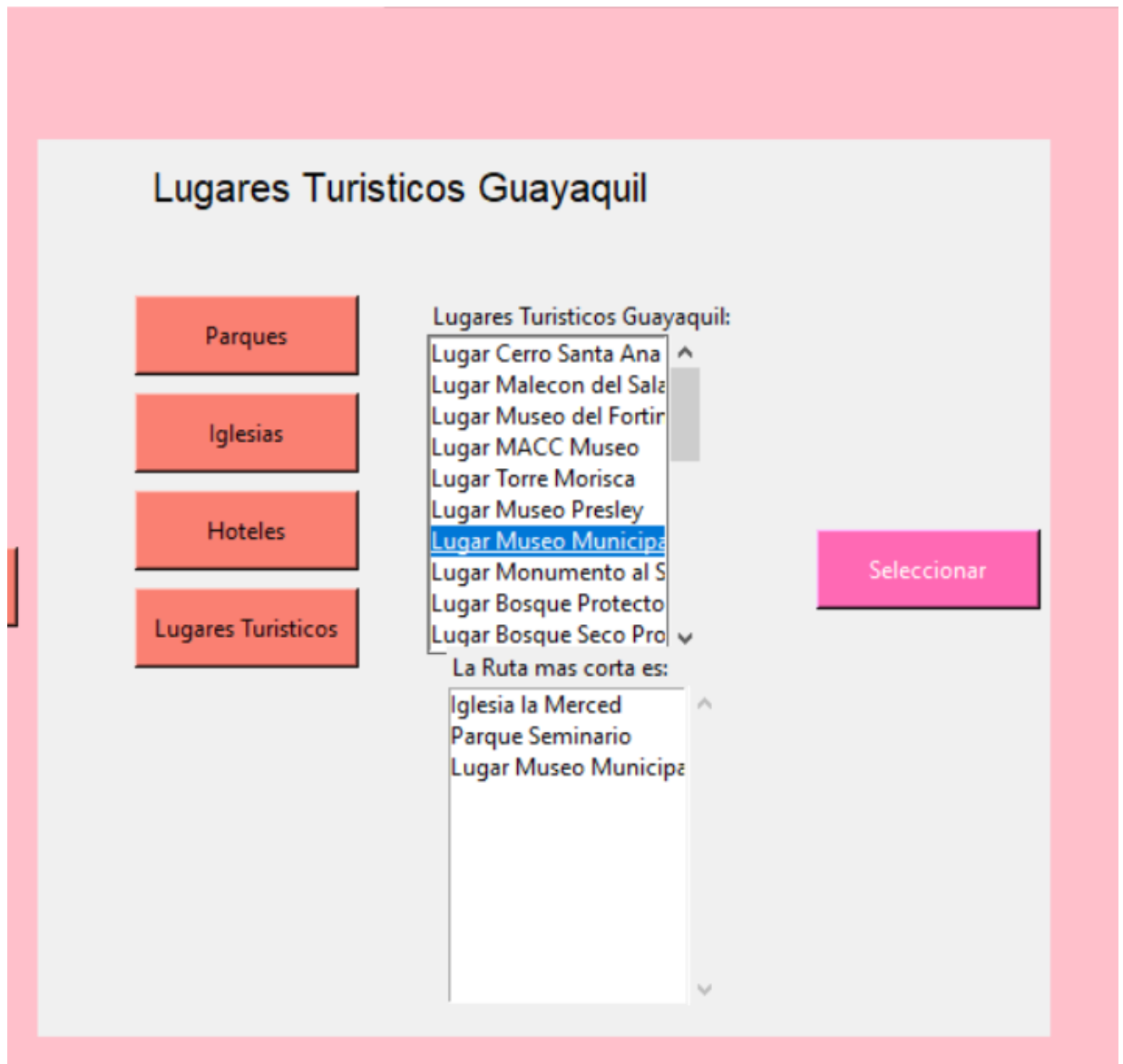
Jugar Recomendaciones

Juego gym

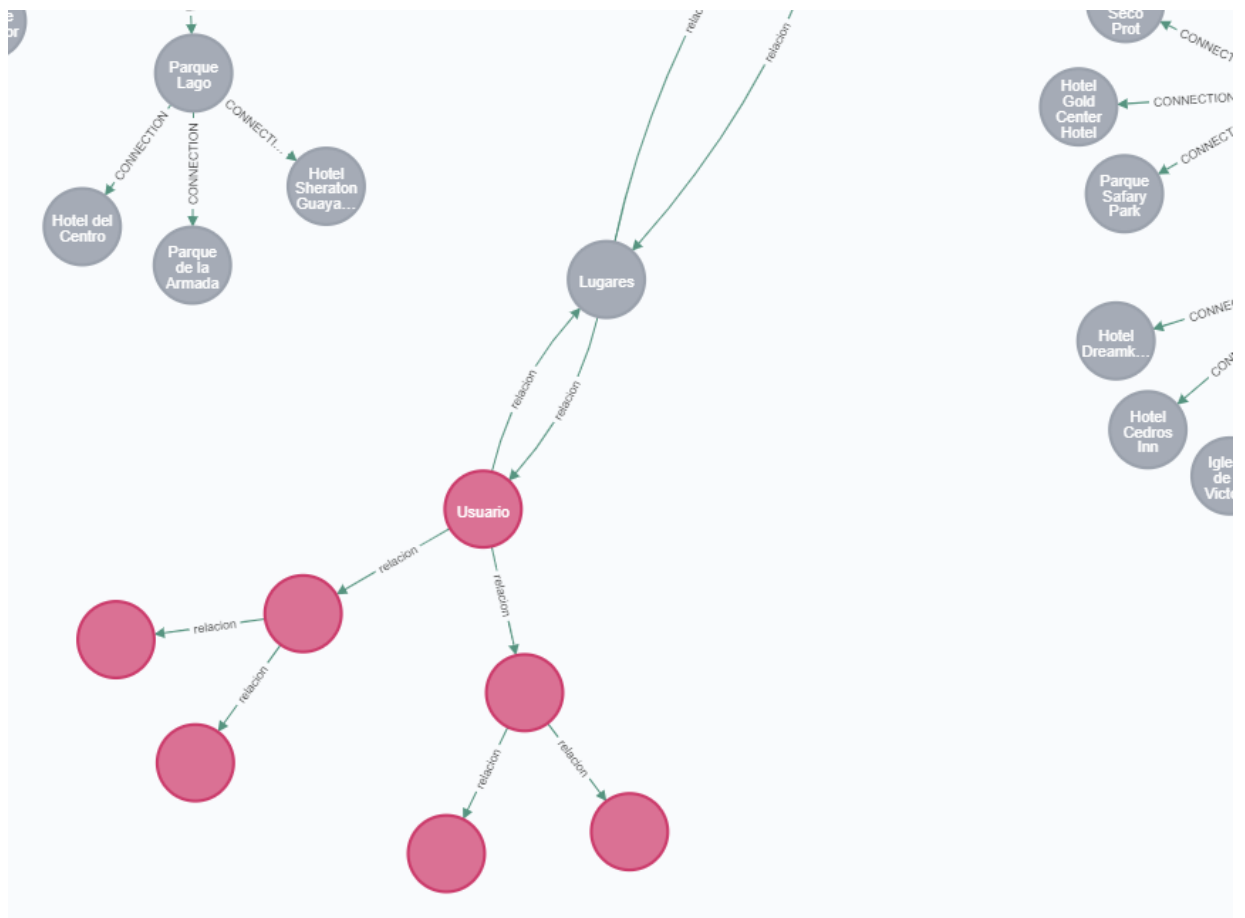


Recomendaciones de Lugares Turisticos de Guayaquil

Realizado en base a los lugares turisticos,hoteles, iglesias y parques almacenados anteriormente



Almacenamiento de usuario, edad, puntuacion de base Neo4j



In []: