Examen Segundo Interciclo Inteligencia Artificial Nombre:Roberto Pacho

Objetivo:

1. Desarrollar un juego (tema libre) empleando una de las 2 siguientes alternativas:

1.easyAl

2.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 jugar 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.

- 2. Dentro del juego el usuario puede registrar he ingresar los gustos de alguna área basadas en el lugar geográfico por ejemplo: comida, películas, lugares turísticos etc.
- 3. En base a la información proporcionada se deberá generar un sistema que permita mostrar lugares de interés, para ello tomar los datos de las tareas y pruebas dentro de una base de datos orientadas a grafos.
- 4. Realizar el sistema con una interfaz gráfica y almacenar los puntajes y datos de los usuarios o jugadores.

Código y documentos de entrega: Se deberá entregar un informe con el procesos dentro del mismo tener capturas del uso del juego y generar un documento en PDF de validación y pruebas. Finalmente subir todo al repositorio incluido los códigos fuentes

Desarrollo:

- Creamos la interfaz Login para guardar los datos del jugadores

● ● ≐		LOGIN IA
	LOGIN	
Usuario	RobertoPacho	
Edad:	26	
Pais:	cuenca/ecuador	
correo [roberto@gmail.com]
	Ingresar	

- Creamos una interfaz de opciones para el jugador como: Jugar/Recomendar



 Usamos la librería Gym para desarrollar el juego Instalamos Gym

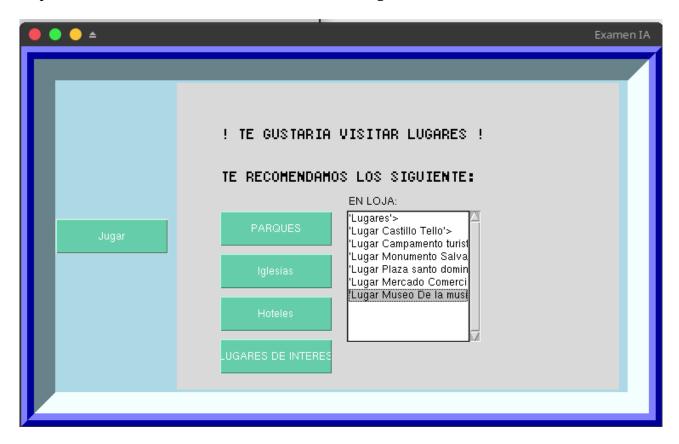


• Importamos la librería en el cuaderno de python

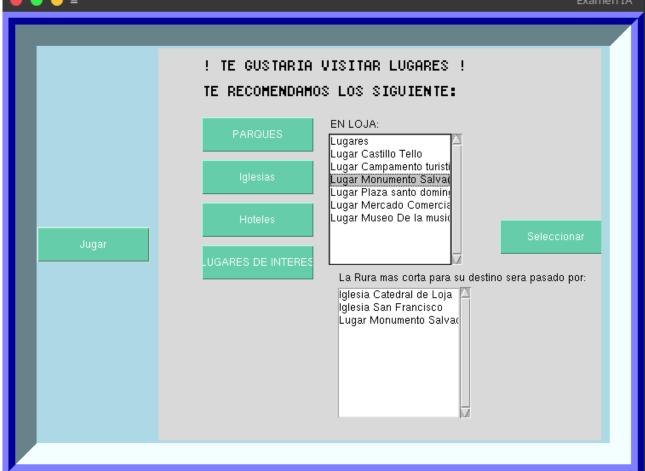
• Al presionar el botón jugar tendremos lo siguiente

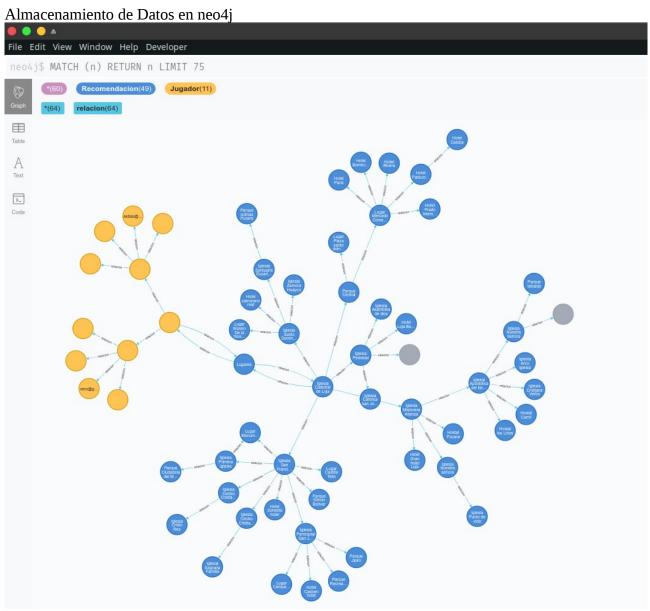


Al presionar el botón Recomendaciones tendremos la siguiente interfaz.



Seleccionamos el lugar y nos mostrara la ruta mas corta ! TE GUSTARIA VISITAR LUGARES !





```
from tkinter import *
import tkinter.font as tkFont
import gym
from neo4j import GraphDatabase
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'examen_ia')
    root1 = Tk()
root1.title("Examen IA")
    frame = Frame(root1, width=480, height=320)
frame.pack(fill='both', expand=1)
frame.config(cursor="pirate")
frame.config(bg="lightblue")
frame.config(bd=25)
     frame.config(relief="sunken")
     def juego():
         def OpenAI_Gym(game,num_episodios,num_inter):
             env= gym.make(game)
              for i_episodio in range(num_episodios):
                  for t in range(num_inter):
                       env.render()
                        recompensa += reward
                       if terminal:
                           recompensa += reward
                            print("recompensa obtenida: ",recompensa)
              with neo4j._driver.session() as session:
                  session.write_transaction(neo4j.crear_puntaje,puntos)
session.write_transaction(neo4j.crear_relacion_usuario_puntaje,usr,puntos)
         OpenAI_Gym(game= 'MsPacman-v0', num_episodios=5,num_inter= 1000)
    def recomendacion():
         vjuego = Frame(frame, width=480, height=320)
         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='! TE GUSTARIA VISITAR LUGARES !', font=fontStyle)
         tipo.place(x=50,y=50)
         etomo=Label(vjuego,text='TE RECOMENDAMOS LOS SIGUIENTE: ', font=fontStyle)
         etomo.place(x=50,y=100)
         def par(self):
              lug=str(self)
             neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'examen_ia')
with neo4j._driver.session() as session:
btn1 = Button(vjuego, text="PARQUES", bg="MediumAquamarine", fg="white", height = 2, width =
15,command=par("Parque"))
        btn2 = Button(vjuego, text="Iglesias", bg="MediumAquamarine", fg="white", height = 2, width =
15,command=par("Igle"))
         btn3 = Button(vjuego, text="Hoteles", bg="MediumAquamarine", fg="white", height = 2, width =
15,command=par("Ho")) btn4 = Button(vjuego, text="LUGARES DE INTERES", bg="MediumAquamarine", fg="white", height = 2, width = 15,command=par("Luga"))
         btn1.place(x=50, y=150)
         btn2.place(x=50, y=200)
         btn3.place(x=50, y=250)
         btn4.place(x=50, y=300)
    btn5 = Button(frame, text="Jugar", bg="MediumAquamarine", fg="white", height = 2, width =
15.command=iuego)
    btn6 = Button(frame, text="Recomendaciones", bg="MediumAquamarine", fg="white", height = 2, width =
15, command=recomendacion)
     btn5.grid(column=1, row=0)
    btn6.grid(column=2, row=0)
    root1.config(cursor="arrow")
     root1.config(bg="blue")
    root1.config(relief="ridge")
```

```
root = Tk()
root.title('LOGIN IA")
root.geometry('400x400')
root.restzable(width=0, height=0)
fontStyle = tkFont.Font(family="Lucida Grande", size=12)
fontStyle = tkFont.Font(family="Lucida Grande", size=30)
fontStyle2 = tkFont.Font(family="Lucida Grande", size=35)
eedad=Label(root,text='Edad: ', font=fontStyle)
eedad.place(x=50,y=150)
edad = Entry(root, fg='black', bg='white', justify='center')
edad.place(x=110, y=150, width=100)
econteu-Laber(vestor, text-correct)
correo.place(x=50,y=250)
correo.place(x=110, y=250, width=200)
class Neo4iService(object):
         def close(self):
    self._driver.close()
          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_pais(self, tx, usuario, pais):
    tx.run("MATCH (a:Jugador {nombre: $usuario}) "
        "MATCH (b:Jugador {pais: $pais}) "
        "MERGE (a)=[r:relacion]->(b)",
        usuario=usuario, pais=pais)
         def crear_relacion_usuario_correo(self, tx, usuario, correo):
    tx.run("MATCH (a:Jugador {nombre: $usuario}) "
        "MATCH (b:Jugador {correo: $correo}) "
        "MERGE (a)=[r:relacion]->(b)",
        usuario=usuario, correo=correo)
         def crear_relacton_usuario_puntaje(self, tx, usuario, puntaje):
    tx.run("MATCH (a:Jugador {nombre: $usuario}) "
        "MATCH (b:Jugador {puntaje: $puntaje}) "
        "MERGE (a)-[r:relaction]->(b)",
        usuario=usuario, puntaje=puntaje)
         def buscar(self,tx,vjuego,lug):
    result = tx,run("MATCH(p:Recomendacion) WHERE p.name STARTS WITH $p RETURN p.name", p=lug)
    etomo=label(vjuego,text="EN LOJA:")
    etomo.place(x=200,y=130)
                     mylist = Listbox(framel, yscrollcommand = scrollbar.set )
for line in result:
    mylist.insert(END, str(line))
                     mylist.pack( side = LEFT, fill = BOTH )
scrollbar.config( command = mylist.yview )
         cerrar():
neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'examen_ia')
usr=str(usuario.get())
ed=str(edd.get())
pa=str(pais.get())
mail=str(correo.get())
```