UNIVERSIDAD POLITECNICA SALESIANA

Alumno: Rayner Palta

Examen de IA:
Enunciado:

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

In [5]:

```
1
2 pip install pygds
3
```

```
Collecting pygds
  Downloading https://files.pythonhosted.org/packages/2e/a2/c0968f941d
0627293ca7b6f91d25a54804fab2c59d86d69a07cf2d01dc0f/pygds-0.2.0-py3-non e-
any.whl (https://files.pythonhosted.org/packages/2e/a2/c0968f941d062
7293ca7b6f91d25a54804fab2c59d86d69a07cf2d01dc0f/pygds-0.2.0-py3-none-a
ny.whl)
Requirement already satisfied: neo4j in /Users/rayner/opt/anaconda3/li
b/python3.7/site-packages (from pygds) (4.2.0)
Requirement already satisfied: pytz in /Users/rayner/opt/anaconda3/li
b/python3.7/site-packages (from neo4j->pygds) (2019.3)
Installing collected packages: pygds
Successfully installed pygds-0.2.0 Note: you may need to restart
the kernel to use updated packages.
```

```
from pygds import GDS
from tkinter import *
 from tkinter import messagebox
 from tkinter import ttk
 import tkinter as tk
 from py2neo import Graph
 from neo4j import GraphDatabase
 from neo4j import unit_of_work

 AUTH = ("neo4j", "neo4jj")
 username = ('neo4j')
 password = ('neo4jj')
 uri = "bolt://localhost:7687"
 driver = GraphDatabase.driver(uri, auth=(username, password))
 session2= driver.session(database="neo4j")
 session2= driver.session()
 #driver2 = GraphDatabase.driver("bolt://localhost:7687",  auth=("neo4j",
 "neo4j

tx = session2
root= Tk() 21
def main():

   app = ventanaPrincipal(root) 24
class ventanaPrincipal:
   def __init__(self, master): 26
self.master = master
         self.master.title('IA')
         self.master.geometry('00x500+0+0')
         self.master.config(bg = 'powder blue')
         self.frame =Frame(self.master, bg = 'powder blue')
         self.frame.pack()

       self.btnProducto = Button(self.frame,  bg='#eb5e0b', text='JUGAR', widt
       self.btnProducto.grid(padx=20, pady=25,row=7, column = 0)
         self.btnCliente = Button(self.frame, bg='#fff3e6',
         text='RECOMENDACIONE
         self.btnCliente.grid(padx=20, pady=25, row=2, column=0, columnspan=5)
         37        #grid(row=7, column = 2) 38    def ventanaNueva(self):
         self.ventanaNueva = Toplevel(self.master)
         self.ventanaProducto = Producto(self.ventanaNueva) 41    def
         ventanaNueva2(self):
         self.ventanaNueva2 = Toplevel(self.master)
         self.ventanaCliente = RecomendacionesR(self.ventanaNueva2) 44
         def ventanaNueva3(self):
         self.ventanaNueva3 = Toplevel(self.master)
         self.ventanaPedido = Pedido(self.ventanaNueva3)

         class RecomendacionesR():
         def __init__(self, window):
         self.wind = window
         self.wind.title('Recomendaciones')
         frame = LabelFrame(self.wind)
         self.wind.config(bg = '#c7ffd8')
         self.frame =Frame(self.wind, bg = '#c7ffd8')
         frame.grid(row = 0, column = 0, columnspan = 3, pady = 20)
```

```
56          btnR = Button(window, bg='#16c79a', text = 'RESTAURANTS', command =
            sel 57          btnR.place(relx=0.5, rely=0.4, anchor=CENTER)
58          btnH = Button(window, bg='#f8dc81', text = 'SITIOS TURISTICOS',
            command
59          btnH.place(relx=0.5, rely=0.5, anchor=CENTER)
```

```
    def ventanaRestaurant(self):
newWindow = Toplevel(root)
newWindow.title("Selecciona gustos")
newWindow.geometry("200x200")          frame
= LabelFrame(newWindow)
newWindow.config(bg = '#161d6f')
        self.frame =Frame(newWindow, bg = '#161d6f')
                  btnR = Button(newWindow, bg='#ffee93', text = 'Carnes',
command = self.
        btnR.place(relx=0.5, rely=0.4, anchor=CENTER)         btnP =
Button(newWindow, bg='#f5d782', text = 'Pizza' , command = self.
btnP.place(relx=0.5, rely=0.5, anchor=CENTER)
        btnM = Button(newWindow, bg='#e97878', text = 'Mariscos', command =
sel         btnM.place(relx=0.5, rely=0.6, anchor=CENTER)
        btnt = Button(newWindow, bg='#e97878', text = 'Restaurants Cercanos',
c         btnt.place(relx=0.5, rely=0.8, anchor=CENTER)

        root.mainloop()     def
ventanaSitio(self):         newWindow =
Toplevel(root)
newWindow.title("Seleccione lugar")
newWindow.geometry("200x200")
frame = LabelFrame(newWindow)
newWindow.config(bg = '#161d6f')
        self.frame =Frame(newWindow, bg = '#161d6f')
                  btnR = Button(newWindow, bg='#ffee93', text = 'Museos',
command = self.         btnR.place(relx=0.5, rely=0.4, anchor=CENTER)
        btnP = Button(newWindow, bg='#f5d782', text = 'Parques' , command =
sel         btnP.place(relx=0.5, rely=0.5, anchor=CENTER)
        btnM = Button(newWindow, bg='#e97878', text = 'Otros', command =
self.o         btnM.place(relx=0.5, rely=0.6, anchor=CENTER)
        btnt = Button(newWindow, bg='#e97878', text = 'Sitios Cercanos',
comman         btnt.place(relx=0.5, rely=0.8, anchor=CENTER)

        root.mainloop()
@unit_of_work(timeout=2)
def obtenerCarne(self):
        #query = "[CALL gds.graph.create('k1Localidad',{Restaurant : {label:
'R         query =session2.run("Match (n:Restaurant) where n.comida='carne'
return         comida = []         for c in query:
comida.append(c)             comida2= []         for x in
range(0,len(comida)):
            #messagebox.showinfo(friends[x], 'Restaurants')  #shows warning
mes             comida2.append(comida[x])             #print (comida2[x])
        messagebox.showwarning('Carnes',comida2
)         return comida2
session2.close()         driver.close()
        def obtenerPizza(self):           query =session2.run("Match
(n:Restaurant) where n.comida='pizzeria' ret         pizza = []         for c
in query:
```

```
110                 pizza.append(c)
```

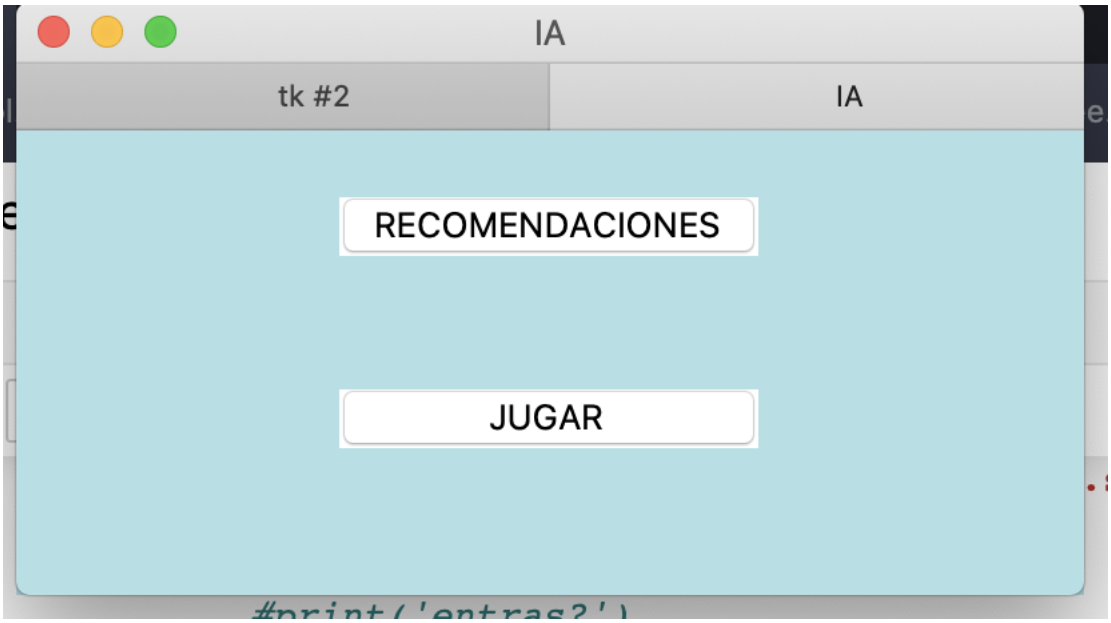
```
pizza2=[]          for x in
range(0,len(pizza)):
pizza2.append(pizza[x])
#messagebox.showinfo(friends[x],
'Restaurants')  #shows warning mes
            #print (pizza[x])
        messagebox.showwarning('Pizzerias',pizza2
)          return pizza2          session2.close()
driver.close()
      def
obtenerMariscos(self):
        #query = "[CALL gds.graph.create('k1Localidad',{Restaurant : {label:
'R          query2 =session2.run("Match (n:Restaurant) where n.comida='marisco'
ret          marisco = []          for c in query2:
            marisco.append(c)
marisco2 =[]          for x in
range(0,len(marisco)):
marisco2.append(marisco[x])

        messagebox.showwarning('Restaurants',marisco2)          return marisco2
session2.close()          driver.close()      @unit_of_work(timeout=5)      def
crearMapa():          query2 = session2.run ("""  CALL
gds.graph.create('k1Localidad',{Restaur          return query2
session2.close()          driver.close()      def
obtenerSimilitudRestaurant(self):          query =session2.run ("""CALL
gds.beta.knn.stream('k1Localidad', {topK:          friends = []          for c
in query:
            print('entras?')
friends.append(c)          nuevaL = []
for x in range(0,len(friends)):
nuevaL.append(friends[x])
        messagebox.showwarning('Restaurants',nuevaL)          return friends
session2.close()          driver.close()        @unit_of_work(timeout=5)
def crearMapa2():          query2 = session2.run ("""  CALL
gds.graph.create('k1LocalidadSitios',{S          return query2
session2.close()          driver.close()
     def
obtenerMuseo(self):
        #query = "[CALL gds.graph.create('k1Localidad',{Restaurant : {label:
'R          query =session2.run("Match (n:Sitios) where n.tipo='museo' return
n.nom          museo = []          for c in query:              museo.append(c)
museo2= []          for x in range(0,len(museo)):
            #messagebox.showinfo(friends[x], 'Restaurants')  #shows warning
mes              museo2.append(museo[x])              #print (comida2[x])
        messagebox.showwarning('Museos',museo2
)          return museo2
session2.close()
```

```
        driver.close()

    def obtenerParque(self):
    query =session2.run("Match (n:Sitios) where n.tipo='parques' return n.n 186
    parque = [] 187         for c in query:
            parque.append(c)
            parque2=[]
            for x in range(0,len(parque)):
            parque2.append(parque[x])
            #messagebox.showinfo(friends[x], 'Restaurants')  #shows warning mes
            #print (pizza[x])
            messagebox.showwarning('Parques ',parque2 )
            return parque2
            session2.close()
            driver.close()

    def obtenerOtros(self):
    #query = "[CALL gds.graph.create('k1Localidad',{Restaurant : {label: 'R 201
    query2 =session2.run("Match (n:Sitios) where n.tipo='otros' return n.no 202
    otros = []
            for c in query2:
            otros.append(c)
            otros2 =[]
            for x in range(0,len(otros)):
            otros2.append(otros[x])

        messagebox.showwarning('Otros',otros)
        return otros
        session2.close()
        driver.close()
        def obtenerSimilitudSitios(self):
        query =session2.run ("""CALL gds.beta.knn.stream('k1LocalidadSitios', {
        215         sitios = [] 216         for c in query:
            #print('entras?')
            sitios.append(c)
            nuevaL = []
            for x in range(0,len(sitios)):
            nuevaL.append(sitios[x])
            messagebox.showwarning('Sitios Cercanos',nuevaL)
            return nuevaL
            session2.close()
            driver.close()

    if __name__ == '__main__':
    main()
    window = Tk()
    window.geometry('400x200')
    #application = Cliente(window)
    window.mainloop()
```

```
entras? entras?
entras? entras?
entras? entras?
entras? entras?
entras? entras?
entras? entras?
entras? entras?
```

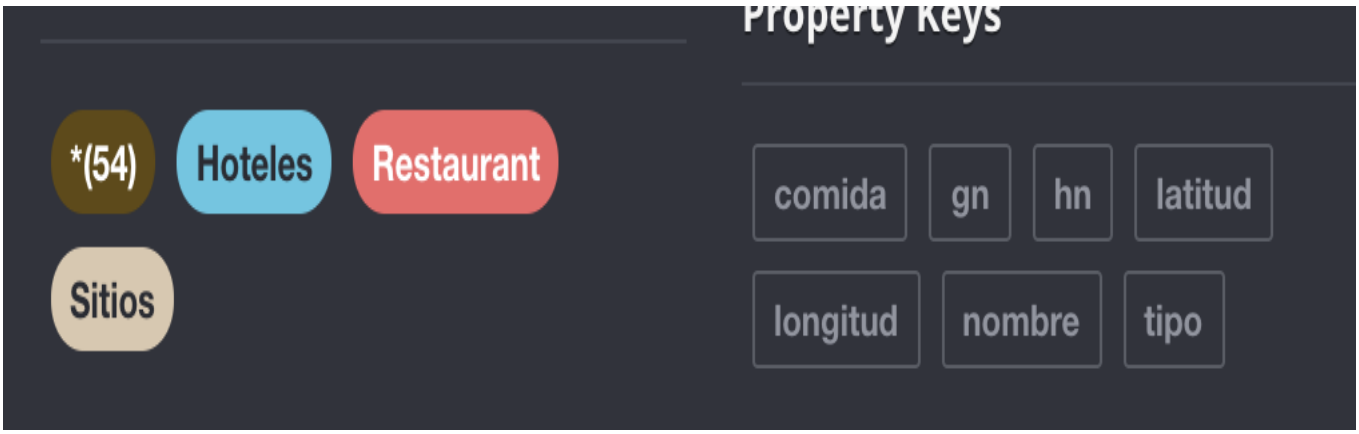
```
entras? entras?
entras? entras?
entras? In [ ]:
```

```
1
```




Carnes
Pizza
Mariscos
Restaurants Cercanos
#print('entras?')


IA
Recomen...
Seleccion...
Video to animated...
Carnes
Pizza
Restaurants C
#print('entras?
sitios.append(c)
nuevaL = []
for x in range(0,ler
nuevaL.append(si
messagebox.showwarni
return nuevaL
Carnes
Nombre {{Parrilladas el Guacho}} marcelo's {{Ali's Parrillada}} {{Alamo chalet}} {{alis }} {{Candies al FOGU}} {{Candies restaurant}} {{los nietos carnes & pizza}} {{la chorizada}} {{El lince Restaurant Ficoa}} {{Restaurant Media Noche Grill}} {{Azar Parrilla Argentina}} {{Tortas del chavo }} {{villa parrilla}} {{Parrilla 63}} {{The Gru}} {{The Grand Restaurant}}
OK


Property Keys
*(54)
Hoteles
Restaurant
Sitios
comida
gn
hn
latitud
longitud
nombre
tipo