

Búsqueda del Tesoro

ALUMNA
CITLALI CUAHTEPITZI CUATLAPANTZI

PROFESOR
MI. MARCO ANTONIO MARTÍNEZ QUINTANA

ASIGNATURA
ESTRUCTURA DE DATOS Y ALGORITMOS I



FECHA
(13/AGOSTO/2021)

SEMESTRE 2021-2

Tabla de contenido

Primeros Videojuegos	3
Búsqueda de Tesoro Sonar.....	4
Código	6
Resultados	14
Recursos informáticos	24
Diagrama de Grantt.....	24
Costos.....	25
Video YouTube	25
Repositorio	25
Conclusiones	26
Referencias	26

Primeros Videojuegos

La programación, desarrollo y consumo de videojuegos parece acrecentarse rumbo a una sociedad orientada al consumo tecnológico, actualmente se encuentran presentes en cualquier dispositivo digital, toma elementos de distintos sectores del software, la computación gráfica, simulaciones físicas y la Inteligencia Artificial entre otras. Para Raph Koster, los videojuegos son puzzles a resolver que representan patrones abstractos del mundo ya sean reales o no y se basan principalmente en el conocimiento, el aprendizaje y el análisis de estos patrones por parte del jugador por lo que estos representan herramientas de aprendizaje.

Contrario a lo que podríamos pensar, los primeros videojuegos fueron creados en universidades con fines de investigación o experimentación, que constituyen un claro antecedente.

Entre estos primeros experimentos se encuentran:

- Cathode Ray Tube Amusement Device (1947)
Desarrollado por Thomas T. Goldsmith Jr. y Estle Ray Mann
Se controlaba por medio de un circuito analógico el brillo y la posición de un punto con un tubo de rayos catódicos intentando crear un simulador de misiles. Los objetivos debían ser superpuestos dado que no era posible dibujar gráficos todavía.
- Ta-te-ti (1952)
Programado por Alexander S. Douglas
Versión programada de Ta-te-ti para el EDSAC (La primer computadora con posibilidad de almacenar programas). Éste fue el primer videojuego en utilizar una pantalla grafica digital y al mismo tiempo el primer programa de juegos del mundo almacenado en una computadora.
- Tennis for Two (1958)
Por William Higinbotham
El objetivo del experimento era poder demostrar que era posible el control interactivo en pantalla. El juego fue programado sobre un osciloscopio.
- Computer Space(1971)
Creado por Nolan Bushnell y Ted Dabney
El jugador controla un cohete que puede disparar a los platillos voladores o incluso a sí mismo.

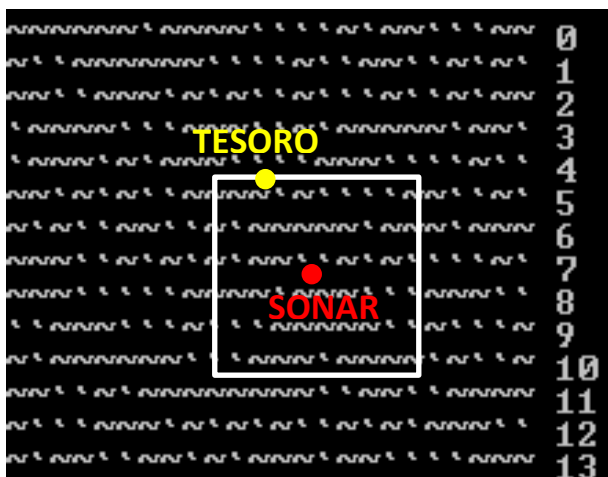
El desarrollo de los videojuegos con que nos encontramos hoy en día es fruto de un equipo y no de una sola persona, como todo comienzo, los videojuegos han recorrido un largo camino desde sus primeros antecedentes y es notable que el camino continúa.

Búsqueda del Tesoro Sonar

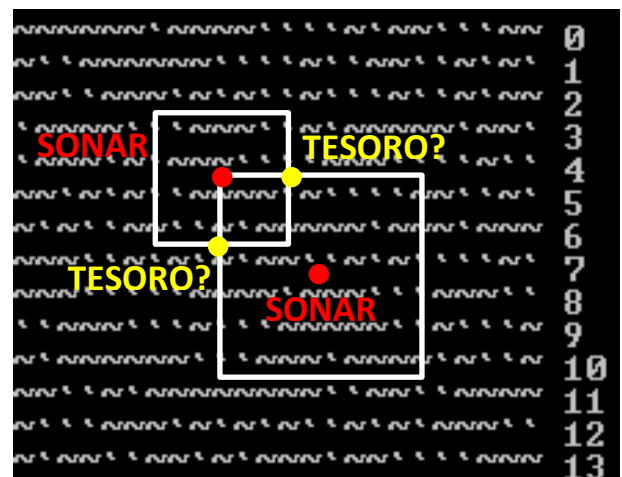
Eres el capitán de Simón, un buque cazador de tesoros. Tu misión actual es encontrar los tres cofres con tesoros perdidos que se hallan ocultos en la parte del océano en que te encuentras y recogerlos.

El sonar es una tecnología que utilizan los barcos para localizar objetos debajo del mar, normalmente estas áreas serían circulares, pero en este juego utilizaremos cuadrados para facilitar la programación. Los dispositivos de sonar en este juego le dirán al jugador que tan lejos están del tesoro más cercano, pero no en qué dirección, al colocar diversos sonares, el jugador puede deducir donde se encuentra el tesoro.

Para jugar, se ingresan las coordenadas del punto del océano en donde se quiere colocar un dispositivo sonar, el sonar puede detectar cuál es la distancia al cofre más cercano.



Al colocar un sonar la línea blanca representa las posibles respuestas
En este caso sonar=3



Al colocar otro sonar, en este caso sonar=2
Las intersecciones son las posibles respuestas

Los cofres del tesoro no se mueven, los dispositivos sonar pueden detectar cofres hasta una distancia 9. Si todos los cofres están fuera del alcance, el punto se indicará con un 0.

Si un dispositivo es colocado directamente sobre un cofre del tesoro, se ha descubierto la ubicación del cofre, y este será recogido. El dispositivo sonar permanecerá allí.

Se realizó la programación en Python, el juego hace uso de las coordenadas cartesianas, lo que la convierte en una herramienta interactiva para desarrollar el análisis e interpretación del plano, por lo que su distribución al estudiante que comience con materias relacionadas, puede serle útil para divertirse aprendiendo.

Este programa puede ser proporcionado a instituciones educativas donde la integración de los recursos tecnológicos hoy en día ya es una necesidad con un costo muy accesible.



Código

BÚSQUEDA DEL TESORO SONAR

```
import random
import sys

def dibujarTablero(tablero):
    # Dibuja la estructura de datos del tablero.

    líneah = ' ' #Espacio inicial para los números a lo largo del lado izquierdo del tablero
    for i in range(1, 6):
        líneah += (' ' * 9) + str(i)

    #Imprimir las coordenadas superiores
    print(líneah)
    print(' ' + ('0123456789' * 6))
    print()

    #Imprimir cada una de las 15 filas
    for i in range(15):
        #Los números de una sola cifra deben ser precedidos por un espacio extra
        if i < 10:
            espacioExtra = ' '
        else:
            espacioExtra = ''
        print('%s%s %s %s' % (espacioExtra, i, obtenerFila(tablero, i), i))

    #Imprime las coordenadas inferiores
    print()
    print(' ' + ('0123456789' * 6))
    print(líneah)

def obtenerFila(tablero, fila):
    # Devuelve la cadena de números de las columnas
    filaTablero = ''
    for i in range(60):
        filaTablero += tablero[i][fila]
    return filaTablero

def obtenerNuevoTablero():
    # Crear una nueva estructura de datos para un tablero de 60x15.
    tablero = []
    for x in range(60): # la lista principal es una lista de 60 listas
        tablero.append([])
```

```

        for y in range(15): # cada lista en la lista principal tiene 15 cadenas de un solo caracter
            # usar diferentes caracteres para el océano para hacerlo más fácil de leer.
            if random.randint(0, 1) == 0:
                tablero[x].append('~')
            else:
                tablero[x].append(' ')
    return tablero

def obtenerCofresAleatorios(númCofres):
    # Crear una lista de estructuras de datos cofre (listas de dos ítems con coordenadas x, y)
    cofres = []
    for i in range(númCofres):
        cofres.append([random.randint(0, 59), random.randint(0, 14)]) # Agrega a la lista cofres
    números aleatorios en el rango
    return cofres

def esMovidaVálida(x, y):
    # Devuelve True si las coordenadas pertenecen al tablero, de lo contrario False.
    return x >= 0 and x <= 59 and y >= 0 and y <= 14

def realizarMovida(tablero, cofres, x, y):
    # Cambia la estructura de datos del tablero agregando un caracter de dispositivo sonar.
    Elimina los cofres
    # de la lista de cofres a medida que son encontrados. Devuelve False si la movida no es
    válida.
    # En caso contrario, devuelve una cadena con el resultado de esa movida.
    if not esMovidaVálida(x, y): # llama a la función def esMovidaVálida(x, y)
        return False

    menorDistancia = 100 # cualquier cofre estará a una distancia menor que 100.
    for cx, cy in cofres:
        if abs(cx - x) > abs(cy - y):
            distancia = abs(cx - x)
        else:
            distancia = abs(cy - y)

        if distancia < menorDistancia: # Para encontrar el cofre más cercano.
            menorDistancia = distancia

    if menorDistancia == 0:
        # El sonar cae sobre el cofre
        cofres.remove([x, y]) # Remueve el cofre
        return "\n\nHas encontrado un cofre del tesoro hundido!"
    else:
        if menorDistancia < 10:
            tablero[x][y] = str(menorDistancia)
            return "\n\nTesoro detectado a una distancia %s del dispositivo sonar." %
(menorDistancia)
        else:

```

```

        tablero[x][y] = 'O'
        return "\n\nEl sonar no ha detectado nada. Todos los cofres están fuera del
alcançe del dispositivo."

```

```

def ingresarMovidaJugador():
    # Permite al jugador teclear su movida. Devuelve una lista de dos ítems con coordenadas xy.
    print("\n\n¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)")
    while True:
        movida = input()
        if movida.lower() == "salir":
            print("\n\n¡Gracias por jugar!")
            sys.exit()

        movida = movida.split()
        if len(movida) == 2 and movida[0].isdigit() and movida[1].isdigit() and
esMovidaVálida(int(movida[0]), int(movida[1])):
            return [int(movida[0]), int(movida[1])]
        print('Ingresa un número de 0 a 59, un espacio, y luego un número de 0 a 14.')

```

```

def jugarDeNuevo():
    # Esta función devuelve True si el jugador quiere jugar de nuevo, de lo contrario devuelve
False.
    print("\n\n¿Quieres jugar de nuevo? (sí o no)")
    return input().lower().startswith('s') # Si respuesta comienza con s

```

```

def mostrarInstrucciones():
    print("\n\nInstrucciones:

```

Eres el capitán de Simón, un buque cazador de tesoros. Tu misión es encontrar los tres cofres con tesoros perdidos que se hallan ocultos en el océano.

Para jugar, ingresa las coordenadas del punto del océano en que quieres colocar un dispositivo sonar. El sonar puede detectar cuál es la distancia al cofre más cercano.

```

Pulsa enter para continuar...")
    input()

    print("

```

Por ejemplo, aquí hay un cofre ubicado a una distancia 2 del dispositivo sonar

```

22222    c = cofre

```



```

c 2 d =sonar
2 d 2
2 2
22222

```

Los dispositivos pueden detectar cofres hasta una distancia de 9, si todos los cofres están fuera del alcance, el punto se indicará con un O.

Si un dispositivo es colocado directamente sobre un cofre del tesoro, has descubierto la ubicación del cofre, y este será recogido. El dispositivo sonar permanecerá allí.

Cuando recojas un cofre, los dispositivos sonar se actualizarán para localizar el próximo cofre más cercano.

Pulsa enter para continuar...\n\n")

```

input()
print()

```

```

import os
os.system("cls")

```

```

print("\n\t\t\t\tBÚSQUEDA DEL TESORO SONAR\n\n")
print()
print("\t\t¿ Te gustaría ver las instrucciones? (sí/no)\n\n")
if input().lower().startswith('s'):#Si respuesta comienza con s
    mostrarInstrucciones()

```

while True:

```

    # configuración del juego
    dispositivosSonar = 16
    elTablero = obtenerNuevoTablero()
    losCofres = obtenerCofresAleatorios(3)
    dibujarTablero(elTablero)
    movidasPrevias = []

```

while dispositivosSonar > 0:

Comienzo de un turno:

mostrar el estado de los dispositivos sonar / cofres

if dispositivosSonar > 1: extraSsonar = 's'

else: extraSsonar = "

if len(losCofres) > 1: extraScofre = 's'

else: extraScofre = "

print("\n\n\nAún tienes %s dispositivos%s sonar. Falta encontrar %s cofre%s." %

(dispositivosSonar, extraSsonar, len(losCofres), extraScofre))

```
x, y = ingresarMovidaJugador()
movidasPrevias.append([x, y]) # debemos registrar todas las movidas para que los
dispositivos sonar puedan ser actualizados.
```

```
resultadoMovida = realizarMovida(elTablero, losCofres, x, y)
if resultadoMovida == False:
    continue
else:
    if resultadoMovida == "\n\n¡Has encontrado uno de los cofres del tesoro!":
        # actualizar todos los dispositivos sonar presentes en el mapa.
        for x, y in movidasPrevias:
            realizarMovida(elTablero, losCofres, x, y)
        dibujarTablero(elTablero)
        print(resultadoMovida)

    if len(losCofres) == 0:
        print("\n\n¡Has encontrado todos los cofres del tesoro! ¡Ganaste :)!")
        break

    dispositivosSonar -= 1

if dispositivosSonar == 0:
    print("\n\n¡Ya no tienes dispositivos sonar!")
    print("\n\nPerdiste :c   Juego terminado.")
    print("\n\nLos cofres restantes estaban aquí:")
    for x, y in losCofres:
        print(' %s, %s' % (x, y))

if not jugarDeNuevo():
    sys.exit()
```

```

1  # BÚSQUEDA DEL TESORO SONAR
2
3  import random
4  import sys
5
6  def dibujarTablero(tablero):
7      # Dibuja la estructura de datos del tablero.
8
9      líneah = ' ' #Espacio inicial para los números a lo largo del lado izquierdo del tablero
10     for i in range(1, 6):
11         líneah += (' ' * 9) + str(i)
12
13     #Imprimir las coordenadas superiores
14     print(líneah)
15     print(' ' + ('0123456789' * 6))
16     print()
17
18     #Imprimir cada una de las 15 filas
19     for i in range(15):
20         #Los números de una sola cifra deben ser precedidos por un espacio extra
21         if i < 10:
22             espacioExtra = ' '
23         else:
24             espacioExtra = ''
25         print('%s%s %s %s' % (espacioExtra, i, obtenerFila(tablero, i), i))
26
27     #Imprime las coordenadas inferiores
28     print()
29     print(' ' + ('0123456789' * 6))
30     print(líneah)
31
32
33 def obtenerFila(tablero, fila):
34     # Devuelve la cadena de números de las columnas
35     filaTablero = ''
36     for i in range(60):
37         filaTablero += tablero[i][fila]
38     return filaTablero
39
40 def obtenerNuevoTablero():
41     # Crear una nueva estructura de datos para un tablero de 60x15.
42     tablero = []
43     for x in range(60): # la lista principal es una lista de 60 listas
44         tablero.append([])
45         for y in range(15): # cada lista en la lista principal tiene 15 cadenas de un solo caracter
46             # usar diferentes caracteres para el océano para hacerlo más fácil de leer.
47             if random.randint(0, 1) == 0:
48                 tablero[x].append('~')
49             else:
50                 tablero[x].append(' ')
51     return tablero
52
53 def obtenerCofresAleatorios(númCofres):
54     # Crear una lista de estructuras de datos cofre (listas de dos ítems con coordenadas x, y)
55     cofres = []
56     for i in range(númCofres):
57         cofres.append([random.randint(0, 59), random.randint(0, 14)]) #Agrega a la lista cofres números aleatorios en el rango
58     return cofres
59
60 def esMovidaVálida(x, y):
61     # Devuelve True si las coordenadas pertenecen al tablero, de lo contrario False.
62     return x >= 0 and x <= 59 and y >= 0 and y <= 14
63
64 def realizarMovida(tablero, cofres, x, y):
65     # Cambia la estructura de datos del tablero agregando un caracter de dispositivo sonar. Elimina los cofres
66     # de la lista de cofres a medida que son encontrados. Devuelve False si la movida no es válida.
67     # En caso contrario, devuelve una cadena con el resultado de esa movida.
68     if not esMovidaVálida(x, y): # llama a la función def esMovidaVálida(x, y)
69         return False
70
71     menorDistancia = 100 # cualquier cofre estará a una distancia menor que 100.
72     for cx, cy in cofres:
73         if abs(cx - x) > abs(cy - y):
74             distancia = abs(cx - x)
75         else:
76             distancia = abs(cy - y)
77
78     if distancia < menorDistancia: #Para encontrar el cofre más cercano.
79         menorDistancia = distancia
80

```

```

81     if menorDistancia == 0:
82         #El sonar cae sobre el cofre
83         cofres.remove([x, y])#Remueve el cofre
84         return "\n\nHas encontrado un cofre del tesoro hundido!"
85     else:
86         if menorDistancia < 10:
87             tablero[x][y] = str(menorDistancia)
88             return "\n\nTesoro detectado a una distancia %s del dispositivo sonar." % (menorDistancia)
89         else:
90             tablero[x][y] = '0'
91             return "\n\nEl sonar no ha detectado nada. Todos los cofres están fuera del alcance del dispositivo."
92
93
94 def ingresarMovidaJugador():
95     # Permite al jugador teclear su movida. Devuelve una lista de dos ítems con coordenadas xy.
96     print("\n\n¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)")
97     while True:
98         movida = input()
99         if movida.lower() == "salir":
100             print("\n\n¡Gracias por jugar!")
101             sys.exit()
102
103         movida = movida.split()
104         if len(movida) == 2 and movida[0].isdigit() and movida[1].isdigit() and esMovidaVálida(int(movida[0]), int(movida[1])):
105             return [int(movida[0]), int(movida[1])]
106         print('Ingresa un número de 0 a 59, un espacio, y luego un número de 0 a 14.')
107
108
109 def jugarDeNuevo():
110     # Esta función devuelve True si el jugador quiere jugar de nuevo, de lo contrario devuelve False.
111     print("\n\n¿Quieres jugar de nuevo? (sí o no)")
112     return input().lower().startswith('s')#Si respuesta comienza con s
113
114
115 def mostrarInstrucciones():
116     print('\n\nInstrucciones:
117
118
119     Eres el capitán de Simón, un buque cazador de tesoros. Tu misión es encontrar los
120     tres cofres con tesoros perdidos que se hallan ocultos en el océano.
121
122     Para jugar, ingresa las coordenadas del punto del océano en que quieres colocar
123     un dispositivo sonar. El sonar puede detectar cuál es la distancia al cofre más
124     cercano.
125
126     ...
127     -Pulsa enter para continuar...'')
128     input()
129
130     print('
131
132     Por ejemplo, aquí hay un cofre ubicado a una distancia 2 del dispositivo sonar
133
134
135         22222      c = cofre
136         c 2      d =sonar
137         2 d 2
138         2   2
139         22222
140
141
142     Los dispositivos pueden detectar cofres hasta una distancia de 9, si todos
143     los cofres están fuera del alcance, el punto se indicará con un 0.
144
145     Si un dispositivo es colocado directamente sobre un cofre del tesoro, has des-
146     cubierto la ubicación del cofre, y este será recogido. El dispositivo sonar
147     permanecerá allí.
148
149     Cuando recojas un cofre, los dispositivos sonar se actualizarán para localizar
150     el próximo cofre más cercano.
151
152     ...
153     -Pulsa enter para continuar...\n\n'')
154     input()
155     print()
156
157     import os
158     os.system("cls")
159
160     print("\n\t\tBÚSQUEDA DEL TESORO SONAR\n\n")
161     print()
162     print("\t\t¿Te gustaría ver las instrucciones? (sí/no)\n\n")
163     if input().lower().startswith('s'):#Si respuesta comienza con s
164         mostrarInstrucciones()
165

```

```

166 while True:
167     # configuración del juego
168     dispositivosSonar = 16
169     elTablero = obtenerNuevoTablero()
170     losCofres = obtenerCofresAleatorios(3)
171     dibujarTablero(elTablero)
172     movidasPrevias = []
173
174     while dispositivosSonar > 0:
175         # Comienzo de un turno:
176
177         # mostrar el estado de los dispositivos sonar / cofres
178         if dispositivosSonar > 1: extraSsonar = 's'
179         else: extraSsonar = ''
180         if len(losCofres) > 1: extraScofre = 's'
181         else: extraScofre = ''
182         print("\n\n\nAún tienes %s dispositivos%s sonar. Falta encontrar %s cofre%s." % (dispositivosSonar, extraSsonar, len(losCofres), extraScofre))
183
184         x, y = ingresarMovidaJugador()
185         movidasPrevias.append([x, y]) # debemos registrar todas las movidas para que los dispositivos sonar puedan ser actualizados.
186
187         resultadoMovida = realizarMovida(elTablero, losCofres, x, y)
188         if resultadoMovida == False:
189             continue
190         else:
191             if resultadoMovida == "\n\n;Has encontrado uno de los cofres del tesoro!":
192                 # actualizar todos los dispositivos sonar presentes en el mapa.
193                 for x, y in movidasPrevias:
194                     realizarMovida(elTablero, losCofres, x, y)
195             dibujarTablero(elTablero)
196             print(resultadoMovida)
197
198         if len(losCofres) == 0:
199             print("\n\n;Has encontrado todos los cofres del tesoro!;Ganaste :)!")
200             break
201
202         dispositivosSonar -= 1
203
204     if dispositivosSonar == 0:
205         print("\n\n;Ya no tienes dispositivos sonar!")
206         print("\n\nPerdiste :c Juego terminado.")
207         print("\n\nLos cofres restantes estaban aquí:")
208         for x, y in losCofres:
209             print(' %s, %s' % (x, y))
210
211     if not jugarDeNuevo():
212         sys.exit()
213
214
215
216
217
218
219

```

Python file

length: 7,190 lines: 248

Ln: 248 Col: 5 Pos: 7,191

Windows (CR LF)

UTF-8

INS

RESULTADOS

```
C:\Windows\system32\cmd.exe - python sonar.py

BÚSQUEDA DEL TESORO SONAR

¿Te gustaría ver las instrucciones? <sí/no>

si

Instrucciones:

Eres el capitán de Simón, un buque cazador de tesoros. Tu misión es encontrar los tres cofres con tesoros perdidos que se hallan ocultos en el océano.

Para jugar, ingresa las coordenadas del punto del océano en que quieres colocar un dispositivo sonar. El sonar puede detectar cuál es la distancia al cofre más cercano.

Pulsa enter para continuar...

Por ejemplo, aquí hay un cofre ubicado a una distancia 2 del dispositivo sonar

      22222      c = cofre
      c  2      d =sonar
      2 d 2
      2  2
      22222

Los dispositivos pueden detectar cofres hasta una distancia de 9, si todos los cofres están fuera del alcance, el punto se indicará con un 0.

Si un dispositivo es colocado directamente sobre un cofre del tesoro, has descubierto la ubicación del cofre, y este será recogido. El dispositivo sonar permanecerá allí.

Cuando recojas un cofre, los dispositivos sonar se actualizarán para localizar el próximo cofre más cercano.

Pulsa enter para continuar...
```


Aún tienes 15 dispositivoss sonar. Falta encontrar 3 cofres.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? <0-59 0-14> <o teclea salir>
34 7

	1	2	3	4	5	
	01234567890	12345678901	23456789012	34567890123	45678901234	567890123456789
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14

012345678901234567890123456789012345678901234567890123456789

1 2 3 4 5

Tesoro detectado a una distancia 7 del dispositivo sonar.

Aún tienes 14 dispositivoss sonar. Falta encontrar 3 cofres.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? <0-59 0-14> <o teclea salir>
27 7

	1	2	3	4	5	
	01234567890	12345678901	23456789012	34567890123	45678901234	567890123456789
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14

012345678901234567890123456789012345678901234567890123456789

1 2 3 4 5

Tesoro detectado a una distancia 7 del dispositivo sonar.

Aún tienes 13 dispositivos sonar. Falta encontrar 3 cofres.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)

27 0

	1	2	3	4	5
0	01234567890123456789012345678901234567890123456789				
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

01234567890123456789012345678901234567890123456789

1 2 3 4 5

El sonar no ha detectado nada. Todos los cofres están fuera del alcance del dispositivo.

Aún tienes 12 dispositivos sonar. Falta encontrar 3 cofres.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)

27 13

	1	2	3	4	5
0	01234567890123456789012345678901234567890123456789				
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

01234567890123456789012345678901234567890123456789

1 2 3 4 5

Tesoro detectado a una distancia 6 del dispositivo sonar.

Aún tienes 11 dispositivoss sonar. Falta encontrar 3 cofres.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? <0-59 0-14> <o teclea salir>

33 13

	1	2	3	4	5	
	01234567890	12345678901	23456789012	34567890123	45678901234	567890123456789
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

01234567890123456789012345678901234567890123456789

1 2 3 4 5

Tesoro detectado a una distancia 1 del dispositivo sonar.

Aún tienes 10 dispositivoss sonar. Falta encontrar 3 cofres.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? <0-59 0-14> <o teclea salir>

21 13

	1	2	3	4	5	
	01234567890	12345678901	23456789012	34567890123	45678901234	567890123456789
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

01234567890123456789012345678901234567890123456789

1 2 3 4 5

Tesoro detectado a una distancia 1 del dispositivo sonar.

```
¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea
salir)
20 13
```

Tesoro detectado a una distancia 1 del dispositivo sonar.

```
¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea
salir)
20 12
```

Tesoro detectado a una distancia 2 del dispositivo sonar.

Aún tienes 7 dispositivoss sonar. Falta encontrar 3 cofres.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)

21 14

```

      1      2      3      4      5
01234567890123456789012345678901234567890123456789
0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0
1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1
2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2
3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3
4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4
5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5
6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6
7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7
8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8
9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9
10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10
11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11
12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12
13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13
14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14

      1      2      3      4      5
01234567890123456789012345678901234567890123456789
i
```

Has encontrado un cofre del tesoro hundido!

Aún tienes 6 dispositivoss sonar. Falta encontrar 2 cofres.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)

34 13

```

      1      2      3      4      5
01234567890123456789012345678901234567890123456789
0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0 ~~~~~ 0
1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1 ~~~~~ 1
2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2 ~~~~~ 2
3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3 ~~~~~ 3
4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4 ~~~~~ 4
5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5 ~~~~~ 5
6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6 ~~~~~ 6
7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7 ~~~~~ 7
8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8 ~~~~~ 8
9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9 ~~~~~ 9
10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10 ~~~~~ 10
11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11 ~~~~~ 11
12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12 ~~~~~ 12
13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13 ~~~~~ 13
14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14 ~~~~~ 14

      1      2      3      4      5
01234567890123456789012345678901234567890123456789
      1      2      3      4      5
```

Tesoro detectado a una distancia 1 del dispositivo sonar.

Aún tienes 5 dispositivos sonar. Falta encontrar 2 cofres.

```
¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea
salir)
```

34 14

[illegible]

Has encontrado un cofre del tesoro hundido?

Aún tienes 4 dispositivos sonar. Falta encontrar 1 cofre.

```
¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)
```

59 8

[illegible]

El sonar no ha detectado nada. Todos los cofres están fuera del alcance del dispositivo.

Aún tienes 3 dispositivos sonar. Falta encontrar 1 cofre.

```
¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)
```

45 7

[illegible]

Tesoro detectado a una distancia 4 del dispositivo sonar.

Aún tienes 2 dispositivos sonar. Falta encontrar 1 cofre.

```
¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)
```

41 11

[illegible]

Tesoro detectado a una distancia 1 del dispositivo sonar.

Aún tienes 1 dispositivos sonar. Falta encontrar 1 cofre.

¿Dónde quieres dejar caer el siguiente dispositivo sonar? (0-59 0-14) (o teclea salir)

42 11

```

      1      2      3      4      5
01234567890123456789012345678901234567890123456789
0 ~~~~~ 0
1 ~~~~~ 1
2 ~~~~~ 2
3 ~~~~~ 3
4 ~~~~~ 4
5 ~~~~~ 5
6 ~~~~~ 6
7 ~~~~~ 7
8 ~~~~~ 8
9 ~~~~~ 9
10 ~~~~~ 10
11 ~~~~~ 11
12 ~~~~~ 12
13 ~~~~~ 13
14 ~~~~~ 14
```

```

      1      2      3      4      5
01234567890123456789012345678901234567890123456789
```

¡

Has encontrado un cofre del tesoro hundido!

¡Has encontrado todos los cofres del tesoro! ¡Ganaste :>!

¿Quieres jugar de nuevo? (sí o no)

```

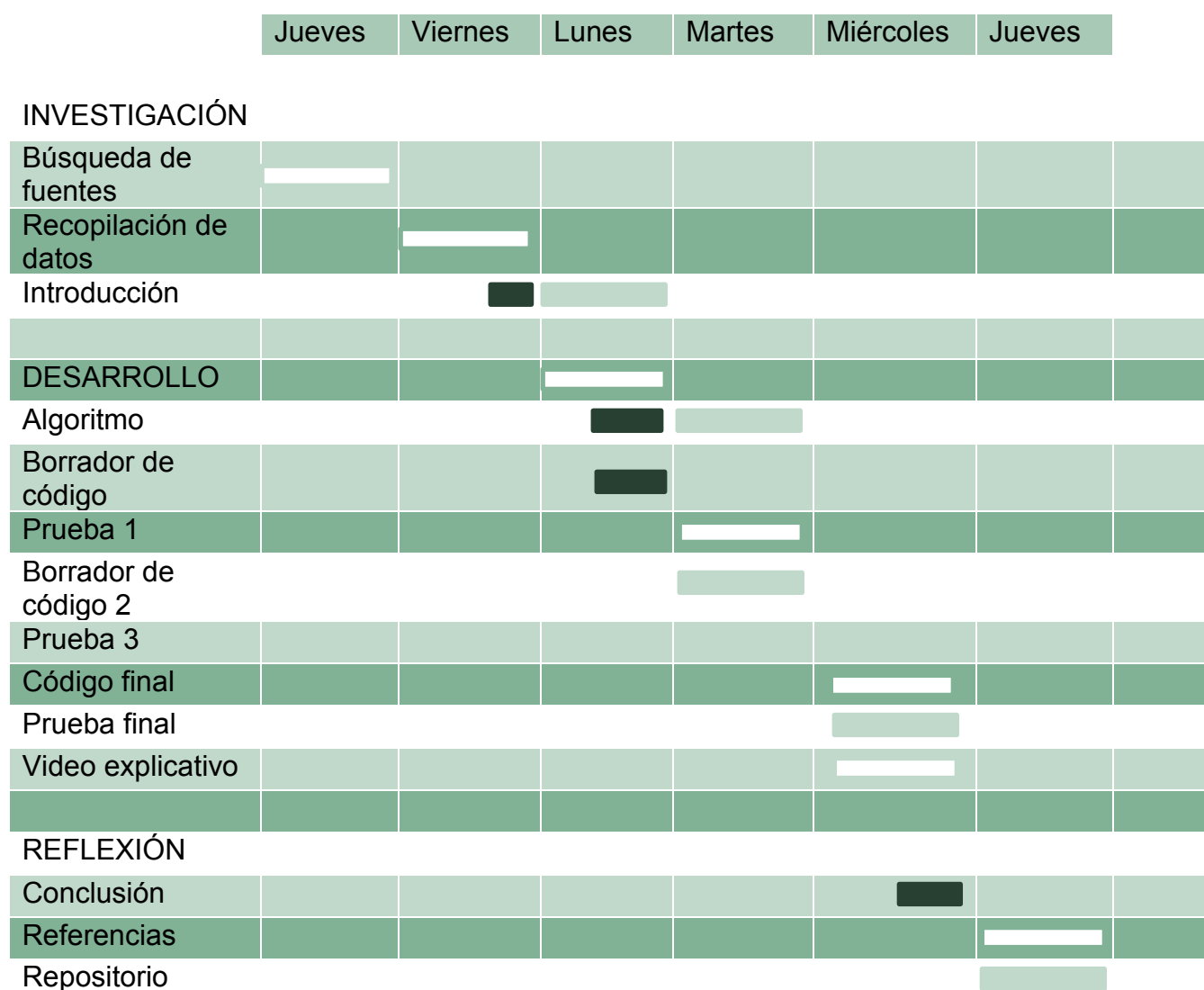
      1      2      3      4      5
01234567890123456789012345678901234567890123456789
0 ~~~~~ 0
1 ~~~~~ 1
2 ~~~~~ 2
3 ~~~~~ 3
4 ~~~~~ 4
5 ~~~~~ 5
6 ~~~~~ 6
7 ~~~~~ 7
8 ~~~~~ 8
9 ~~~~~ 9
10 ~~~~~ 10
11 ~~~~~ 11
12 ~~~~~ 12
13 ~~~~~ 13
14 ~~~~~ 14
```

Recursos informáticos

Configuración de nuestro equipo de cómputo Herramientas a utilizar en laptop o PC

- Editor de Texto Plano -Notepad++ Windows
- Terminal -Intérprete de Python

Diagrama de Gantt



Costos

El juego “Búsqueda de tesoro Sonar” es una opción creativa y amigable con cualquier sector de la población que disfrute el análisis. Al ser este tipo de juego su costo para su distribución es acorde a los conocimientos que se requirieron para su elaboración. Los costos podrían aumentar si se añade alguna otra función a las ya presentadas o se añade algún tipo de representación gráfica llamativa

	COSTO
Proceso de investigación	\$ 50.00
Elaboración de algoritmo y diagrama de flujo	\$ 200.00
Elaboración de código	\$ 500.00
Pruebas y modificaciones	\$ 200.00
Pruebas y modificaciones finales	\$ 100.00
TOTAL	\$1050.00

Video YouTube

<https://youtu.be/HizAoNXMS78>

Repositorio

<https://github.com/CitlaliCuahtepitziCuatlapantzi/EDA-I/tree/main/Proyecto%20Final>

Conclusiones

El uso de algoritmos y estructura de datos se presenta en cada actividad de nuestro día a día, desde una edad temprana comenzamos a seguir instrucciones en base a los posibles resultados considerando distintos escenarios, todo esto con principal objetivo, resolver problemas. A estas instrucciones son lo que ahora conocemos como algoritmos orientados a la programación, son estos la base de todo lo que desarrollamos, es por eso de su gran importancia realizar algoritmos correctos.

El juego presentado, a pesar de que fue capaz de jugarse, no se libra de la falta de consideración de posibles escenarios, es por eso que el desarrollo de cualquier página, juego, aplicación etc. Requiere de un gran equipo y múltiples pruebas.

Las estructuras de datos se refieren al uso de variables complejas como las listas en listas, que permiten, como su nombre indica, estructurar datos relacionados entre sí, para realizar con ellos operaciones ya sea en conjunto o de forma individual, esto es de gran utilidad para el desarrollo de programas con un nivel mayor de complejidad, que nos facilita y ahorra código. Su buen uso optimiza cualquier programa, será capaz de comprenderse con mayor facilidad e incluso tendrá una presentación gráfica con mayor calidad. Combinar buenos algoritmos y estructuras de datos resultan en brindar una herramienta que como ya se mencionó, resolverá problemas por nosotros, de manera más rápida y siempre con la respuesta correcta.

Realizar este proyecto de un juego en el lenguaje Python me ha posicionado en lo que en su tiempo, fueron los primeros videojuegos, el comienzo de esta actividad recreativa que rodea al mundo e incluso reflexionar sobre el proceso fructuoso que se ha venido acrecentado exponencialmente. Los videojuegos en la actualidad son mucho más elaborados, llenos de distintos componentes, herramientas, y ramas de estudio que se hacen imposibles de imaginar. En el curso de Fundamentos de Programación aprendí mi primer lenguaje de programación, C, en este Curso de Algoritmos y Estructura de Datos I, comienzo con otro, Python, el cual fue más fácil comprender puesto que traía las principales bases, pero también adopte nuevos conceptos en este lenguaje, conceptos que desarrollan habilidades de estructuración y que me permitieron realizar este pequeño pero interesante juego.

Referencias

Iglesias, A. A., & Blanque, J. (2011). Desarrollo de videojuegos.

Recuperado de: http://tesis.blanque.com.ar/Home_files/Tesis_Alejandro_Adrian_Iglesias.pdf

Sweigart, A. (2008). Inventa sus propios juegos de computadora con Python 3rd Edition.

Recuperado de: http://inventwithpython.com/inventwithpython_early3rded.pdf

Documentación de Python 3.9.6: <https://docs.python.org/3/>