UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ingeniería

Buscaminas

Pineda Cruz Tania

Profesor: M.I. Marco Antonio Martínez Quintana

2021-2

Estructura de Datos y Algoritmos I

10 de agosto de 2021

Resumen

En el presente documento se abordará sobre la planificación y desarrollo que se llevó a cabo para la realización de un programa que te permite realizar un clásico juego llamado “Buscaminas” en lenguaje de programación Python.

Hay que recordar que Python es un lenguaje de programación interpretado, multiparadigma y multiplataforma usado, principalmente, en Big Data, Inteligencia Artificial, Data Science y desarrollo web. Esto lo convierte en un lenguaje de propósito general de gran nivel debido a su extensa biblioteca, cuya colección ofrece una amplia gama de instalaciones. El lenguaje Python se caracteriza por ser simple, rápido y tener una curva de aprendizaje amigable y corta. Está desarrollado bajo una licencia de código abierto, por lo que es de libre uso y distribución.

Existen lenguajes de programación que dificultan a los programadores documentar todo un proyecto. En ese sentido, la construcción de un proyecto puede ser extremadamente abrumadora cuando se usa otro lenguaje de programación.

Sin embargo, cuando optamos por el lenguaje de programación Python no hay problemas que puedan detener ese proceso. Python tiene una comunidad solidaria que facilita la obtención de tutoriales. Además, se pueden encontrar varias guías y documentos en línea y fuera de línea para los nuevos programadores.

Pero, ¿por qué es importante la planeación para el desarrollo de cualquier proyecto? Un proyecto consta de múltiples facetas y disciplinas las cuales intervienen directa o indirectamente obligándonos desde el principio a identificar la manera correcta en que debemos organizar cada una de las actividades involucradas y su nivel de importancia. La correcta planificación nos ayuda a establecer la prioridad de cada una de las actividades y a tener un mejor control del tiempo para ejecutar un proyecto con la calidad deseada y con éxito.

Así mismo, debemos conocer los recursos que se requieren para la realización del programa y de igual manera, saber con cuáles nos encontramos familiarizados; averiguar de qué es lo que se está haciendo en realidad: ¿cómo se verá?, ¿qué características incluirá?, ¿qué características no incluirá?

Introducción

Hace años, varias empresas consiguieron ver la tecnología como una forma de hacer entretenimiento. La velocidad con que se implementan los nuevos juegos y las nuevas tecnologías impactan en la imagen de los padres. Además de reducir el interés por otras actividades y contribuir al sedentarismo, debido al escaso movimiento físico, la mayoría de los videojuegos promueve la satisfacción inmediata y la omnipotencia, lo que genera escasa tolerancia a la frustración.

Sin embargo, estudios recientes señalan las ventajas que tienen los juegos en términos cognitivos. Permiten que los niños logren desarrollar una serie de destrezas que los ayudarán a enfrentar de forma más efectiva las exigencias escolares a nivel cognitivo. Mejorar su razonamiento, coordinación visomotora, capacidad de atención y concentración.

El juego es una herramienta indispensable para lograr que el niño desarrolle ese pensamiento lógico, este no sólo recrea, sino que ayuda a que, dentro de esta diversión, la persona alcance un conocimiento relacionando los objetos mediante la manipulación, la reflexión, la comparación y la actuación en diferentes escenarios.

Entre los distintos juegos destinados a potenciar las habilidades mentales, los que requieren el uso de la lógica para resolverlos son los más entretenidos y a la vez beneficiosos para ejercitar todas las formas de razonamiento. Generalmente no es necesario tener amplios conocimientos de matemáticas o de otras disciplinas para hallar las soluciones a estos problemas o juegos de lógica, basta con razonar de forma inductiva y deductiva y saber relacionar conceptos. No hace falta tener una determinada edad para divertirse con estos juegos.

El famoso juego Buscaminas ha sido uno de los juegos más representativos en la sociedad, es un sistema relativamente básico en el que hay que evitar tocar las minas o bombas escondidas en las celdas. El juego fue creado por Curt Johnson.

La lógica del juego es sencilla. El jugador se enfrenta a campos de minas y debe, con la información que aparece en el tablero, evitar que todo explote.

Descripción general

Los videojuegos han dejado de lado a mercados tan fuertes y tradicionales como el cine o la televisión, ya que las opciones de realidad aumentada, los dispositivos como las gafas de realidad virtual y la constante evolución gráfica y narrativa de los videojuegos, han puesto a esta industria a la cabeza del sector de entretenimiento audiovisual.

El desarrollo de videojuegos (sobre todo en plataformas móviles) es un gran impulso a la economía global, ya que la industria ofrece cada vez más posibilidades de desarrollo e ingresos para los programadores de videojuegos y los desarrolladores de tecnologías móviles, en un mercado donde los principales ganadores son los que disfrutan.

El mercado de los videojuegos es cada vez más valioso y su crecimiento es exponencial. Forman parte de uno de los sectores de mayor crecimiento. Los videojuegos están en la categoría de “industrias creativas”, donde el capital fundamental es la creatividad humana. A primera vista los videojuegos pueden considerarse una distracción para pasar el tiempo, pero si atravesamos esta noción primaria, descubrimos que pueden optimizar la experiencia de las personas, mejorando el aprendizaje, la resolución de conflictos y el relacionamiento en equipos. A su vez, fomentan la educación y la preservación de las culturas. La demanda del sector fue creciendo en el tiempo, muchos pensaban que los videojuegos iban a correr la suerte de una moda pasajera.

Este proyecto consiste en la elaboración y desarrollo de un programa que te permite crear un tablero en el cual se podrán representar las casillas y bombas que un buscaminas tiene. El objetivo principal es volver a implementar los juegos de estrategia y destreza que en tiempos atrás se utilizaban mucho para el entretenimiento y de esta forma también poder adquirir mayor conocimiento en el mundo de la programación. Es un juego sencillo de realizar y se puede adaptar a las posibilidades que el jugador requiera, así mismo se pueden implementar nuevas estrategias para su uso.

Algoritmo

Problema: seguir el algoritmo para crear un Buscaminas.

Entrada: lenguaje de programación Python / Teclado.

Salida: ejecución del juego / Teclado / Pantalla.

1. Determinar el tamaño del tablero que se desea presentar en la pantalla.
2. Crear tablero en relación con el tamaño antes descrito. Colocar un separador por cada hilera y columna.
3. Colocar el número que le corresponde a cada hilera y columna. Todas las funciones reciben el tablero.
4. Colocar las minas o bombas de manera aleatoria. Se puede cambiar el número de estas y así agregar dificultar al juego. Lo que se hace es obtener un número aleatorio tanto para la hilera como para la columna.
5. Abrir casilla o tablero. Esta función es la que invoca el usuario al momento de jugar. Se trata de abrir una casilla y dependiendo de lo que haya en la misma, se devuelve un estado.
6. Obtener minas cercanas. Se visitan las casillas que están junto a la casilla de la coordenada indicada. No importa si está en una esquina o en un borde.
7. El usuario gana cuando ya no hay espacios por descubrir. Por lo tanto, lo único que hay que hacer es comprobar que ya no haya ninguna casilla con esa característica, recorriendo el tablero.
8. Para el juego hacemos un ciclo infinito que se romperá cuando el jugador pierda o gane. En el mismo, solicitar las coordenadas e indicar el estado.

Código fuente

```
import os
os.system("cls")
#Proyecto: Juego de buscaminas
import random
import re

#Mensaje de Bienvenida
print("\n\n\t\t\tBuscaminas ¿Estás listo para jugar?\n\n\t\t\t")
print("\t\t\t\t\tBy Tania\n\n\t\t\t")

#Crear tablero para representar al juego
class Tablero:
    def __init__(self, dim_size, num_bombas):
        self.dim_size = dim_size
        self.num_bombas = num_bombas
        self.tablero = self.hacer_nuevo_tablero()
        self.asignar_valores_tablero()
        self.excavar = set()

    def hacer_nuevo_tablero(self):
        #Generar un nuevo tablero
        tablero = [[None for _ in range(self.dim_size)] for _ in range(self.dim_size)]

        #Colocar las bombas
        bombas_colocadas = 0
        while bombas_colocadas < self.num_bombas:
            lugar = random.randint(0, self.dim_size**2 - 1)
            hilera = lugar // self.dim_size
            columna = lugar % self.dim_size
```

```
        if tablero[hilera][columna] == '*':
            continue

        tablero[hilera][columna] = '*'
        bombas_colocadas += 1

    return tablero

def asignar_valores_tablero(self):
    for h in range(self.dim_size):
        for c in range(self.dim_size):
            if self.tablero[h][c] == '*':
                continue
            self.tablero[h][c] = self.conseguir_num_bombas_vecinas(h, c)

def conseguir_num_bombas_vecinas(self, hilera, columna):

    num_bombas_vecinas = 0
    for h in range(max(0, hilera-1), min(self.dim_size-1, hilera+1)+1):
        for c in range(max(0, columna-1), min(self.dim_size-1, columna+1)+1):
            if h == hilera and c == columna:
                continue
            if self.tablero[h][c] == '*':
                num_bombas_vecinas += 1

    return num_bombas_vecinas

def dig(self, hilera, columna):
```

```
        self.excavar.add((hilera, columna))

        if self.tablero[hilera][columna] == '*':
            return False
        elif self.tablero[hilera][columna] > 0:
            return True

        for h in range(max(0, hilera-1), min(self.dim_size-1, hilera+1)+1):
            for c in range(max(0, columna-1), min(self.dim_size-1, columna+1)+1):
                if (h, c) in self.excavar:
                    continue
                self.dig(h, c)
        return True

    def __str__(self):

        tablero_visible = [[None for _ in range(self.dim_size)] for _ in range(self.dim_size)]
        for hilera in range(self.dim_size):
            for columna in range(self.dim_size):
                if (hilera,columna) in self.excavar:
                    tablero_visible[hilera][columna] = str(self.tablero[hilera][columna])
                else:
                    tablero_visible[hilera][columna] = ' '

        string_rep = ''
        amplitud = []
        for idx in range(self.dim_size):
            columnas = map(lambda x: x[idx], tablero_visible)
            amplitud.append(
                len(
```

```
            max(columnas, key = len)
        )
    )

    indices = [i for i in range(self.dim_size)]
    indices_hilera = '   '
    celdas = []
    for idx, columna in enumerate(indices):
        format = '%-' + str(amplitud[idx]) + "s"
        celdas.append(format % (columna))
    indices_hilera += '  '.join(celdas)
    indices_hilera += '  \n'

    for i in range(len(tablero_visible)):
        hilera = tablero_visible[i]
        string_rep += f'{i} |'
        celdas = []
        for idx, columna in enumerate(hilera):
            format = '%-' + str(amplitud[idx]) + "s"
            celdas.append(format % (columna))
        string_rep += ' |'.join(celdas)
        string_rep += ' |\n'

    str_len = int(len(string_rep) / self.dim_size)
    string_rep = indices_hilera + '-'*str_len + '\n' + string_rep + '-'*str_len

    return string_rep

#Jugar
def play(dim_size=10, num_bombas=10):
```

```
    #Paso 1: Crear el tablero
    tablero = Tablero(dim_size, num_bombas)
    #Paso 2: Mostrar el tablero y preguntar dónde quiere excavar
    #Paso 3: Si la ubicación es una bomba, mostrar el mensaje de "Game Over"
    #Paso 4: Si la ubicación no es una bomba, volver a excavar
    #Paso 5: Repetir el paso 2 y 3 hasta que no haya más lugares
    safe = True

    while len(tablero.excavar) < tablero.dim_size ** 2 - num_bombas:
        print(tablero)
        user_input = re.split(',(\\s)*', input("\n\n\t\t¿Dónde te gustaría excavar? Fila, columna: "))  # '0,
3'
        hilera, columna = int(user_input[0]), int(user_input[-1])
        if hilera < 0 or hilera >= tablero.dim_size or columna < 0 or columna >= dim_size:
            print("Invalid location. Try again.")
            continue

        safe = tablero.dig(hilera, columna)
        if not safe:
            break

    if safe:
        print("\n\n\t\t\t\t¡¡¡¡FELICIDADES!!!! ¡GANASTE!\n\n\t\t")
    else:
        print("\n\n\t\t\t\tGAME OVER :(\n\n\t\t")
        tablero.excavar = [(h,c) for h in range(tablero.dim_size) for c in range(tablero.dim_size)]
        print(tablero)

if __name__ == '__main__':
    play()
```

Resultados

C:\Users\super_000\Desktop\Python\pro.py - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

fun.py | listas.py | calificaciones.py | archivos.py | usuario.py | examen.py | pro.py

```python
import os
os.system("cls")
#Proyecto: Juego de buscaminas
import random
import re

#Mensaje de Bienvenida
print("\n\n\t\t\tBuscaminas ¿Estás listo para jugar?\n\n\t\t\t")
print("\t\t\t\t\tBy Tania\n\n\t\t\t")

#Crear tablero para representar al juego
class Tablero:
    def __init__(self, dim_size, num_bombas):
        self.dim_size = dim_size
        self.num_bombas = num_bombas
        self.tablero = self.hacer_nuevo_tablero()
        self.asignar_valores_tablero()
        self.excavar = set()

    def hacer_nuevo_tablero(self):
        #Generar un nuevo tablero
        tablero = [[None for _ in range(self.dim_size)] for _ in range(self.dim_size)]

        #Colocar las bombas
        bombas_colocadas = 0
        while bombas_colocadas < self.num_bombas:
            lugar = random.randint(0, self.dim_size**2 - 1)
            hilera = lugar // self.dim_size
            columna = lugar % self.dim_size

            if tablero[hilera][columna] == '*':
                continue

            tablero[hilera][columna] = '*'
            bombas_colocadas += 1
```

```python
        return tablero

    def asignar_valores_tablero(self):
        for h in range(self.dim_size):
            for c in range(self.dim_size):
                if self.tablero[h][c] == '*':
                    continue
                self.tablero[h][c] = self.conseguir_num_bombas_vecinas(h, c)

    def conseguir_num_bombas_vecinas(self, hilera, columna):

        num_bombas_vecinas = 0
        for h in range(max(0, hilera-1), min(self.dim_size-1, hilera+1)+1):
            for c in range(max(0, columna-1), min(self.dim_size-1, columna+1)+1):
                if h == hilera and c == columna:
                    continue
                if self.tablero[h][c] == '*':
                    num_bombas_vecinas += 1

        return num_bombas_vecinas

    def dig(self, hilera, columna):

        self.excavar.add((hilera, columna))

        if self.tablero[hilera][columna] == '*':
            return False
        elif self.tablero[hilera][columna] > 0:
            return True

        for h in range(max(0, hilera-1), min(self.dim_size-1, hilera+1)+1):
            for c in range(max(0, columna-1), min(self.dim_size-1, columna+1)+1):
```

```python
                if (h, c) in self.excavar:
                    continue
                self.dig(h, c)
        return True

    def __str__(self):

        tablero_visible = [[None for _ in range(self.dim_size)] for _ in range(self.dim_size)]
        for hilera in range(self.dim_size):
            for columna in range(self.dim_size):
                if (hilera,columna) in self.excavar:
                    tablero_visible[hilera][columna] = str(self.tablero[hilera][columna])
                else:
                    tablero_visible[hilera][columna] = ' '

        string_rep = ''
        amplitud = []
        for idx in range(self.dim_size):
            columnas = map(lambda x: x[idx], tablero_visible)
            amplitud.append(
                len(
                    max(columnas, key = len)
                )
            )

        indices = [i for i in range(self.dim_size)]
        indices_hilera = '   '
        celdas = []
        for idx, columna in enumerate(indices):
            format = '%-' + str(amplitud[idx]) + "s"
            celdas.append(format % (columna))
        indices_hilera += '  '.join(celdas)
        indices_hilera += '  \n'
```

```python

        for i in range(len(tablero_visible)):
            hilera = tablero_visible[i]
            string_rep += f'{i} |'
            celdas = []
            for idx, columna in enumerate(hilera):
                format = '%-' + str(amplitud[idx]) + "s"
                celdas.append(format % (columna))
            string_rep += ' |'.join(celdas)
            string_rep += ' |\n'

        str_len = int(len(string_rep) / self.dim_size)
        string_rep = indices_hilera + '-'*str_len + '\n' + string_rep + '-'*str_len

        return string_rep

#Jugar
def play(dim_size=10, num_bombas=10):
    #Paso 1: Crear el tablero
    tablero = Tablero(dim_size, num_bombas)
    #Paso 2: Mostrar el tablero y preguntar dónde quiere excavar
    #Paso 3: Si la ubicación es una bomba, mostrar el mensaje de "Game Over"
    #Paso 4: Si la ubicación no es una bomba, volver a excavar
    #Paso 5: Repetir el paso 2 y 3 hasta que no haya más lugares
    safe = True

    while len(tablero.excavar) < tablero.dim_size ** 2 - num_bombas:
        print(tablero)
        user_input = re.split(',(\\s)*', input("\n\n\t\t¿Dónde te gustaría excavar? Fila, columna: "))  # '0, 3'
        hilera, columna = int(user_input[0]), int(user_input[-1])
        if hilera < 0 or hilera >= tablero.dim_size or columna < 0 or columna >= dim_size:
            print("Invalid location. Try again.")
            continue
```

```python

        safe = tablero.dig(hilera, columna)
        if not safe:
            break

    if safe:
        print("\n\n\t\t\t\t¡¡¡FELICIDADES!!!! ¡GANASTE!\n\n\t\t")
    else:
        print("\n\n\t\t\t\tGAME OVER :(\n\n\t\t")
        tablero.excavar = [(h,c) for h in range(tablero.dim_size) for c in range(tablero.dim_size)]
        print(tablero)

if __name__ == '__main__':
    play()
```

Python file | length : 5,327 lines : 148 | Ln : 114 Col : Símbolo del si

```
                        Buscaminas ¿Estás listo para jugar?


                                          By Tania


   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |  |  |  |  |  |  |  |  |  |  |
1 |  |  |  |  |  |  |  |  |  |  |
2 |  |  |  |  |  |  |  |  |  |  |
3 |  |  |  |  |  |  |  |  |  |  |
4 |  |  |  |  |  |  |  |  |  |  |
5 |  |  |  |  |  |  |  |  |  |  |
6 |  |  |  |  |  |  |  |  |  |  |
7 |  |  |  |  |  |  |  |  |  |  |
8 |  |  |  |  |  |  |  |  |  |  |
9 |  |  |  |  |  |  |  |  |  |  |
---------------------------------


                ¿Dónde te gustaría excavar? Fila, columna: 2,5
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |  |  |  |  |1 |0 |0 |1 |  |  |
1 |  |  |  |  |1 |0 |0 |1 |  |  |
2 |  |  |  |  |2 |0 |0 |1 |1 |1 |
3 |  |  |  |  |2 |0 |0 |0 |0 |0 |
4 |  |  |  |  |2 |0 |0 |0 |0 |0 |
5 |  |  |  |  |2 |0 |0 |0 |0 |0 |
6 |  |  |  |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |  |1 |0 |
---------------------------------
```

```
                ¿Dónde te gustaría excavar? Fila, columna: 1,9
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |  |  |  |  |1 |0 |0 |1 |  |  |
1 |  |  |  |  |1 |0 |0 |1 |  |1 |
2 |  |  |  |  |2 |0 |0 |1 |1 |1 |
3 |  |  |  |  |2 |0 |0 |0 |0 |0 |
4 |  |  |  |  |2 |0 |0 |0 |0 |0 |
5 |  |  |  |  |2 |0 |0 |0 |0 |0 |
6 |  |  |  |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |  |1 |0 |
---------------------------------


                ¿Dónde te gustaría excavar? Fila, columna: 0,9
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |  |  |  |  |1 |0 |0 |1 |  |1 |
1 |  |  |  |  |1 |0 |0 |1 |  |1 |
2 |  |  |  |  |2 |0 |0 |1 |1 |1 |
3 |  |  |  |  |2 |0 |0 |0 |0 |0 |
4 |  |  |  |  |2 |0 |0 |0 |0 |0 |
5 |  |  |  |  |2 |0 |0 |0 |0 |0 |
6 |  |  |  |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |  |1 |0 |
---------------------------------
```

```
                ¿Dónde te gustaría excavar? Fila, columna: 0,8
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |  |  |  |  |1 |0 |0 |1 |1 |1 |
1 |  |  |  |  |1 |0 |0 |1 |  |1 |
2 |  |  |  |  |2 |0 |0 |1 |1 |1 |
3 |  |  |  |  |2 |0 |0 |0 |0 |0 |
4 |  |  |  |  |2 |0 |0 |0 |0 |0 |
5 |  |  |  |  |2 |0 |0 |0 |0 |0 |
6 |  |  |  |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |  |1 |0 |
---------------------------------


                ¿Dónde te gustaría excavar? Fila, columna: 9,7
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |  |  |  |  |1 |0 |0 |1 |1 |1 |
1 |  |  |  |  |1 |0 |0 |1 |  |1 |
2 |  |  |  |  |2 |0 |0 |1 |1 |1 |
3 |  |  |  |  |2 |0 |0 |0 |0 |0 |
4 |  |  |  |  |2 |0 |0 |0 |0 |0 |
5 |  |  |  |  |2 |0 |0 |0 |0 |0 |
6 |  |  |  |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |1 |1 |0 |
---------------------------------
```

```
                ¿Dónde te gustaría excavar? Fila, columna: 5,3
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |  |  |  |  |1 |0 |0 |1 |1 |1 |
1 |  |  |  |  |1 |0 |0 |1 |  |1 |
2 |  |  |  |  |2 |0 |0 |1 |1 |1 |
3 |  |  |  |  |2 |0 |0 |0 |0 |0 |
4 |  |  |  |  |2 |0 |0 |0 |0 |0 |
5 |  |  |  |3 |2 |0 |0 |0 |0 |0 |
6 |  |  |  |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |1 |1 |0 |
---------------------------------


                ¿Dónde te gustaría excavar? Fila, columna: 2, 1
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |0 |0 |1 |  |1 |0 |0 |1 |1 |1 |
1 |0 |0 |1 |  |1 |0 |0 |1 |  |1 |
2 |0 |0 |2 |  |2 |0 |0 |1 |1 |1 |
3 |0 |0 |2 |  |2 |0 |0 |0 |0 |0 |
4 |0 |1 |3 |  |2 |0 |0 |0 |0 |0 |
5 |0 |1 |  |3 |2 |0 |0 |0 |0 |0 |
6 |1 |3 |  |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |1 |1 |0 |
---------------------------------
```

```
                ¿Dónde te gustaría excavar? Fila, columna: 6,2
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |0 |0 |1 |  |1 |0 |0 |1 |1 |1 |
1 |0 |0 |1 |  |1 |0 |0 |1 |  |1 |
2 |0 |0 |2 |  |2 |0 |0 |1 |1 |1 |
3 |0 |0 |2 |  |2 |0 |0 |0 |0 |0 |
4 |0 |1 |3 |  |2 |0 |0 |0 |0 |0 |
5 |0 |1 |  |3 |2 |0 |0 |0 |0 |0 |
6 |1 |3 |3 |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |1 |1 |0 |
---------------------------------


                ¿Dónde te gustaría excavar? Fila, columna: 2,3
   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |0 |0 |1 |  |1 |0 |0 |1 |1 |1 |
1 |0 |0 |1 |  |1 |0 |0 |1 |  |1 |
2 |0 |0 |2 |2 |2 |0 |0 |1 |1 |1 |
3 |0 |0 |2 |  |2 |0 |0 |0 |0 |0 |
4 |0 |1 |3 |  |2 |0 |0 |0 |0 |0 |
5 |0 |1 |  |3 |2 |0 |0 |0 |0 |0 |
6 |1 |3 |3 |  |1 |0 |0 |0 |0 |0 |
7 |  |  |  |2 |1 |0 |1 |1 |1 |0 |
8 |  |  |2 |1 |0 |0 |1 |  |1 |0 |
9 |  |  |1 |0 |0 |0 |1 |1 |1 |0 |
---------------------------------
```

```
                ¿Dónde te gustaría excavar? Fila, columna: 7,2


                              GAME OVER :(


   0  1  2  3  4  5  6  7  8  9
---------------------------------
0 |0 |0 |1 |1 |1 |0 |0 |1 |1 |1 |
1 |0 |0 |1 |* |1 |0 |0 |1 |* |1 |
2 |0 |0 |2 |2 |2 |0 |0 |1 |1 |1 |
3 |0 |0 |2 |* |2 |0 |0 |0 |0 |0 |
4 |0 |1 |3 |* |2 |0 |0 |0 |0 |0 |
5 |0 |1 |* |3 |2 |0 |0 |0 |0 |0 |
6 |1 |3 |3 |* |1 |0 |0 |0 |0 |0 |
7 |* |3 |* |2 |1 |0 |1 |1 |1 |0 |
8 |2 |* |2 |1 |0 |0 |1 |* |1 |0 |
9 |1 |1 |1 |0 |0 |0 |1 |1 |1 |0 |
---------------------------------

C:\Users\super_000\Desktop\Python>pro.py
```

Recursos informáticos [software y hardware]

Agosto 2021

Software	Hardware
Sistema operativo Windows	Ordenador
Navegador	Memoria RAM
Editor de texto	Teclado (dispositivo de entrada)
Compilador	Monitor/Pantalla (dispositivo de salida)
Lenguaje de programación Python	---------

Costos asociados

Agosto 2021

¿Cuánto estarías dispuesto a pagar por que alguien desarrolle tu proyecto?	$150.00
¿Cuánto cobrarías si tú lo hicieras?	$250.00
¿Cuánto gastarás en tu proyecto?	$0.00
¿Cuánto tiempo te llevaría desarrollarlo?	3 días

Diagrama de Gantt

Agosto 2021				
Etapas	07/08/21	08/08/21	09/08/21	10/08/21
Idea general del proyecto				
Recursos informáticos por utilizar				
Algoritmo				
Desarrollo del código				
Compilación y ejecución				
Proyecto escrito				
Presentación del proyecto				

Canal de YouTube

Enlace: https://youtu.be/StpsB20ap1g

Nombre del canal: Pineda Cruz Tania

Repositorio de GitHub

Enlace: https://github.com/PinedaCruzTania34/EDA-I-Proyecto-Final

Conclusiones

Un algoritmo es un conjunto de instrucciones definidas, ordenadas y acotadas para resolver un problema o realizar una tarea. Primero debemos encontrar la solución al problema (definir el algoritmo), para luego, a través del código, poder indicarle a la máquina qué acciones queremos que lleve a cabo. De este modo, un programa no sería más que un conjunto de algoritmos ordenados y codificados en un lenguaje de programación para poder ser ejecutados en un ordenador.

Los algoritmos son muy importantes a la hora de programar ya que son los pasos exactos para resolver un problema. Esto quiere decir que es un procedimiento computacional bien definido que requiere de datos de entrada y produce un valor como salida.

No obstante, los algoritmos no son algo exclusivo de los ámbitos de las matemáticas, la lógica y la computación. Utilizamos numerosos algoritmos para resolver problemas en nuestra vida cotidiana. Algunos de los ejemplos más habituales son los manuales de instrucciones o las recetas de cocina.

Las estructuras de datos son parte fundamental del proceso evolutivo de la programación, son la base para el desarrollo de programas complejos que se diseñan cuidadosamente desde el punto en que se requiere no solo facilitar sino también aumentar la eficiencia. Inicialmente los programas no eran estructurados porque carecían de organización, estructuras de control y datos complejos; estos fueron introducidos a medida que evolucionaban los lenguajes, lo que permitió crear datos compuestos a partir de la agrupación de datos simples. Cuando hablamos de programación, la estructura de datos está representada por la forma determinada que se tiene de organizar los datos de un grupo para que podamos utilizarlos de la manera más efectiva posible.

Con estas estructuras tenemos la posibilidad de administrar todo tipo de datos sin ningún tipo de obstáculo, algo que en la actualidad s usa en la red para poder llevar a cabo, por ejemplo, los sistemas de manejo de contenidos, jugando así un papel importante en la creación de los mejores algoritmos.

Este proyecto fue muy diferente en todos los aspectos en relación con el proyecto de la materia de Fundamentos de Programación ya que, el tiempo de la creación del proyecto fue muy distinto, es decir, en el proyecto anterior tuve más tiempo de poder realizar una mejor organización y en este el tiempo fue más corto, sin embargo, en las dos situaciones me adapté para poder logar un resultado exitoso.

Con los dos proyectos tuve experiencias nuevas y sobre todo aprendí de nuevas cosas para poder implementarlas en algún futuro como ingeniera, este proyecto me permitió conocer más sobre los videojuegos e identificar el impacto que existe en la sociedad.

El crecimiento en los ingresos de los videojuegos para dispositivos móviles va en aumento. La penetración de esta tecnología en mercados emergentes y superpoblados, hacen que este tipo de dispositivos sean el futuro para gran parte del sector. El potencial demográfico, junto con el desarrollo tecnológico de estos dispositivos, plantea un crecimiento de los ingresos del sector de los videojuegos muy alto en los próximos años.

En comparación con la industria global aún queda mucho camino por andar, pero el camino para crecer y acortar esta brecha es la educación. Actualmente la formación académica es universitaria pero el foco debería ser impulsar la formación en programación, arte ilustrativo, composición musical y diseño de productos desde etapas más tempranas. “Tenemos que pensar en dónde la Inteligencia Artificial no podrá reemplazar al humano y la respuesta es: en industrias creativas”.

Referencias

(2018). Buscaminas: la historia de uno de los juegos más adictivos. Agosto 09, 2021, de Cromo Sitio web: https://www.elobservador.com.uy/nota/buscaminas-la-historia-de-uno-de-los-juegos-mas-adictivos-201821911290

(2019). Los videojuegos, el espejo donde se mira y mejora la sociedad. Agosto 08, 2021, de Portaltic Sitio web: https://www.europapress.es/portaltic/videojuegos/noticia-videojuegos-espejo-donde-mira-mejora-sociedad-20191009090739.html

Machuca, F. (2021). ¿Qué es Python? El lenguaje de programación más popular para aprender en 2021. Agosto 07, 2021, de Crehana Sitio web: https://www.crehana.com/mx/blog/desarrollo-web/que-es-python/

Vázquez, M. (2009). Potenciar las habilidades mentales mediante el juego. Agosto 08, 2021, de Consumer Sitio web: https://www.consumer.es/educacion/potenciar-las-habilidades-mentales-mediante-el-juego.html