

---

# JUEGO DEL AHORCADO

Patricia Aguayo Escudero

---

# ÍNDICE

1

ENTIDAD  
RELACIÓN

2

NORMALIZACIÓN

3

BASE DE  
DATOS

4

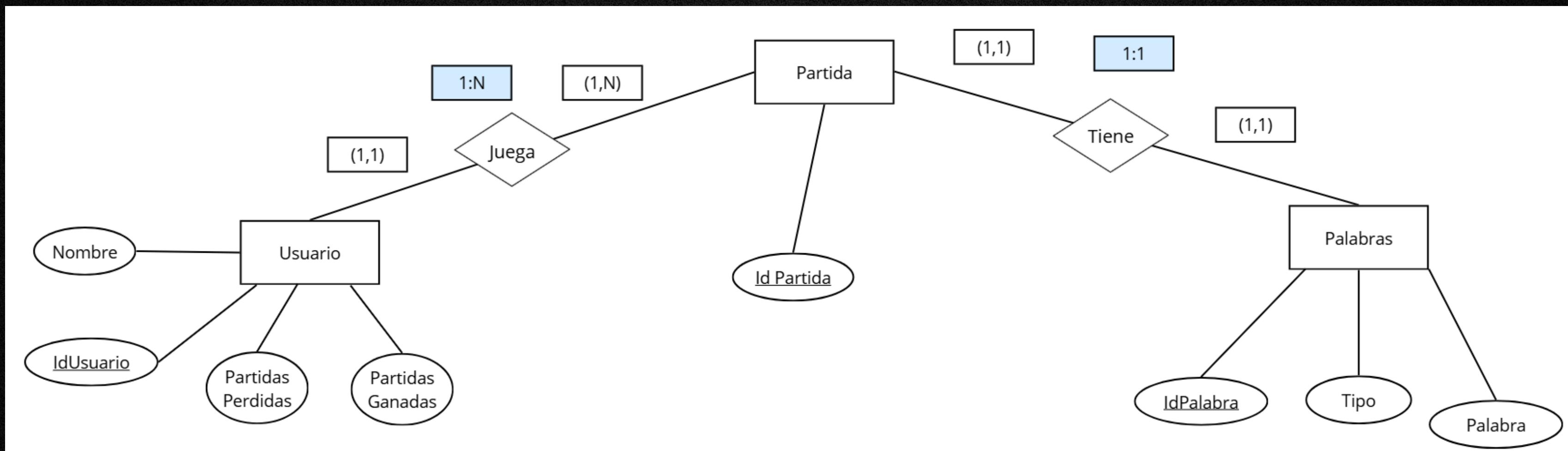
CÓDIGO

5

REPOSITORIO  
GIT HUB



# 1. ENTIDAD RELACIÓN



### 3. BASE DE DATOS

```
# Función para crear la base de datos
Codeium: Refactor | Explain | Docstring | X
def crear_base_datos(): 1 usage ▲ Patri
    conexión = sqlite3.connect("ahorcado.db")
    cursor = conexión.cursor()

    # Crear tabla usuarios
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS usuarios (
            idUsuario INTEGER PRIMARY KEY AUTOINCREMENT,
            nombre TEXT NOT NULL,
            Ganadas INTEGER DEFAULT 0,
            Perdidas INTEGER DEFAULT 0
        );
    """)

    # Crear tabla tematicas
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS palabras (
            idPalabra INTEGER PRIMARY KEY AUTOINCREMENT,
            palabra TEXT NOT NULL UNIQUE,
            tipo TEXT NOT NULL
        );
    """)
```

```
datos_tematicas = [
    ('carla', 'Personas'),
    ('alberto', 'Personas'),
    ('andrea', 'Personas'),
    ('mariano', 'Personas'),
    ('martin', 'Personas'),
    ('naranja', 'Frutas'),
    ('pera', 'Frutas'),
    ('manzana', 'Frutas'),
    ('uva', 'Frutas'),
    ('platano', 'Frutas'),
    ('pantalla', 'Conceptos_Informaticos'),
    ('intezfaz', 'Conceptos_Informaticos'),
    ('metodo', 'Conceptos_Informaticos'),
    ('variable', 'Conceptos_Informaticos'),
    ('clase', 'Conceptos_Informaticos'),
```

```
cursor.executemany(sql: """
    INSERT OR IGNORE INTO palabras (palabra, tipo)
    VALUES (?, ?);
""", datos_tematicas)

cursor.execute("""
CREATE TABLE IF NOT EXISTS partidas (
    idPartida INTEGER PRIMARY KEY AUTOINCREMENT,
    idUsuario INTEGER NOT NULL,
    idPalabra INTEGER NOT NULL,
    FOREIGN KEY (idUsuario) REFERENCES usuarios (idUsuario),
    FOREIGN KEY (idPalabra) REFERENCES palabras (idPalabra)
);
""")

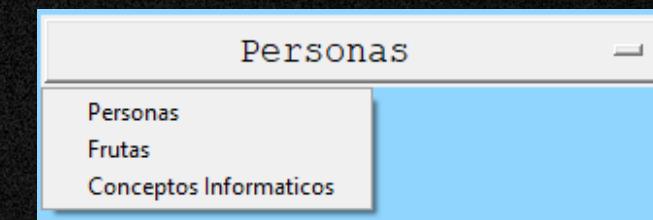
conexión.commit()
conexión.close()
print("Base de datos creada exitosamente.")
```

```
def conectar_bd(): 8 usages ▲ Patri
    return sqlite3.connect("ahorcado.db")

Codeium: Refactor | Explain | Docstring | X
def obtener_usuario(nombre): 2 usages ▲ Patri
    conn = conectar_bd()
    cursor = conn.cursor()
    cursor.execute(sql: "SELECT idUsuario, Ganadas, Perdidas FROM usuarios WHERE nombre = ?", parameters: (nombre,))
    usuario = cursor.fetchone()
    conn.close()
    return usuario
```

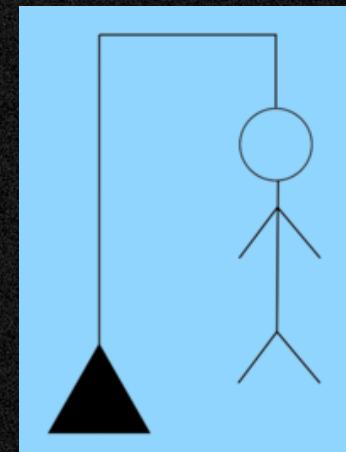
## 4. CÓDIGO

```
Tipos = tk.StringVar(value="Personas")
menu_temas = tk.OptionMenu(juego, Tipos, value: "Personas", *values: "Frutas", "Conceptos Informaticos")
menu_temas.place(x=470, y=450)
menu_temas.config(font=("Courier New", 14),width=25, bg="#fbffcc8")
```



```
imagenes = [
    tk.PhotoImage(file="../Resources/Reducidas/Vida1.png"),
    tk.PhotoImage(file="../Resources/Reducidas/Vida2.png"),
    tk.PhotoImage(file="../Resources/Reducidas/Vida3.png"),
    tk.PhotoImage(file="../Resources/Reducidas/Vida4.png"),
    tk.PhotoImage(file="../Resources/Reducidas/Vida5.png"),
    tk.PhotoImage(file="../Resources/Reducidas/Vida6.png"),
    tk.PhotoImage(file="../Resources/Reducidas/Vida7.png"),
]
```

```
imagenLabel = tk.Label(juego, image=imagenes[0], bg=juego.cget("bg"))
imagenLabel.place(x=100, y=200)
```



## 4. CÓDIGO

```
Codeium: Refactor | Explain | Docstring | X
def reiniciar_juego():
    global palabra, palabra_guiones, letras_incorrectas, intentos_restantes
    palabra = ""
    palabra_guiones = []
    letras_incorrectas = set()
    intentos_restantes = 6

    # Reiniciar campos en la interfaz gráfica

    palabraSecreta.config(text="")
    entradaLetra.delete( first: 0, tk.END)
    entradaLetra.config(state=tk.DISABLED)
    Pganadas.config(text="Ganadas: 0")
    Pperdidas.config(text="Perdidas: 0")
    intentos.config(text="Intentos restantes: 6")
    incorrectas.config(text="Letras incorrectas: ")
    entradaNombre.delete( first: 0, tk.END)
    entradaNombre.config(state=tk.NORMAL)
    Tipos.set("Personas")
    imagenLabel.config(image=imagenes[0])
```

```
# Variables globales
palabra = ""
palabra_guiones = []
letras_incorrectas = set()
intentos_restantes = 6
usuario_id = None
ganadas = 0
perdidas = 0
nombre = ""
```

```
def actualizar_pantalla():
    palabraSecreta.config(text=" ".join(palabra_guiones))
    intentos.config(text=f"Intentos restantes: {intentos_restantes}")
    incorrectas.config(text="Letras incorrectas: ")
    Pganadas.config(text=f"Ganadas: {ganadas}")
    Pperdidas.config(text=f"Perdidas: {perdidas}")
    entradaLetra.config(state=tk.NORMAL)
```

```
palabra = random.choice(palabras)

palabra_id = obtener_id_palabra(palabra)
insertar_partida(usuario_id, palabra_id)

palabra_guiones = ["_"] * len(palabra)
letras_incorrectas = set()
intentos_restantes = 6

actualizar_pantalla()
```

## 5. GIT HUB



Enlace al repositorio del juego  
del ahorcado: [Git Patri](#)

FIN

