

# SOFTWARE DE APLICACIÓN

## LAS TORRES DE HANOI



# MANUAL DE USUARIO

## **SOFTWARE DE APLICACIÓN**

**TorresHanoi v1.0:** El juego tiene tres torres y varios discos de diferentes tamaños colocados en uno. Se deben mover todos los discos a otra torre, moviendo un disco a la vez y sin colocar un disco grande sobre un disco menor.

## **AUTORES:**

- LEONEL COYLA IDME
- ELQUI YEYE PARI CONDORI
- JUAN REYNALDO PAREDES QUISPE
- JOSÉ PÁNFILO TITO LIPA
- ALFREDO MAMANI CANQUI

## ÍNDICE

1.	INTRODUCCIÓN .....	4
2.	REQUISITOS DEL SISTEMA .....	5
2.1.	SISTEMA OPERATIVO.....	5
2.2.	HARDWARE .....	5
2.3.	SOFTWARE .....	5
3.	INSTALACIÓN Y CONFIGURACIÓN .....	5
3.1.	INSTALAR PYTHON .....	5
3.2.	EJECUTAR LA APLICACIÓN .....	5
3.3.	CONFIGURACIÓN OPCIONAL .....	6
3.4.	ACTUALIZACIÓN .....	6
4.	DESCRIPCIÓN DE LA APLICACIÓN TORRESHANOI v1.0 .....	7
4.1.	MENÚ PRESENTACIÓN .....	7
4.3.	MENÚ AYUDA .....	11
4.4.	MENÚ ACERCA DE .....	11
5.	PREGUNTAS FRECUENTES (FAQS) SOBRE EL SOFTWARE .....	13
5.1.	¿QUÉ ES TORRES DE HANOI Y CUÁL ES SU PROPÓSITO? .....	13
5.2.	¿CUÁLES SON LOS REQUISITOS MÍNIMOS DEL SISTEMA PARA EJECUTAR? .....	13
5.3.	¿CÓMO INSTALO O CONFIGURO TORRESHANOI? .....	13
6.	ADVERTENCIAS Y SUGERENCIAS DEL SOFTWARE DE APLICACIÓN. ....	14
6.1.	ADVERTENCIAS.....	14
6.2.	SUGERENCIAS .....	15
7.	CÓDIGO .....	16

## 1. INTRODUCCIÓN

Las Torres de Hanoi son un juego matemático clásico que fue inventado por el matemático francés Édouard Lucas en 1883. Está compuesto por tres torres y un grupo de discos de diferentes tamaños apilados desde un disco grande hasta un disco pequeño. El objetivo principal es trasladar todos los discos de la torre 1 a la torre 3, teniendo la torre 2 como auxiliar. Además, debe considerarse que ningún disco grande debe estar sobre un disco pequeño.

Se considera la siguiente regla desafiante: solo se puede mover un disco a la vez y no puede colocarse un disco grande sobre un disco pequeño. Entonces, el jugador debe pensar con lógica y planificar correctamente el movimiento del disco origen al disco destino. Si consideramos el incremento del número de discos, la dificultad del juego se incrementa.

PROYECTO REGISTRADO CON URL:

<https://github.com/leonelcoyla/TorresHanoi>

## 2. REQUISITOS DEL SISTEMA

Para garantizar un rendimiento óptimo de TorresHanoi v1.0, el software diseñado para jugar las Torres de Hanoi, Es importante que tu equipo cumpla con las especificaciones mínimas y recomendadas. Verifica que tanto el hardware como el software estén debidamente configurados antes de proceder con la instalación y uso del programa.

### 2.1. Sistema Operativo

Al aplicativo TorresHanoi v1.0 es compatible con los siguientes sistemas operativos:

- Windows: 7, 8, 10, 11 (recomendado Windows 10 o superior)
- macOS: 10.14 Mojave o superior
- Linux: Ubuntu 18.04 o superior, Debian, Fedora, etc.

### 2.2. Hardware

- Procesador: Intel/AMD 1 GHz o superior
- Memoria RAM: 2 GB mínimo (recomendado 4 GB)
- Espacio en disco: 100 MB libres
- Pantalla: 1024x768 píxeles o superior

### 2.3. Software

- Python: Versión 3.8 o superior
- Tkinter: Incluido por defecto en la instalación oficial de Python
- Editor sugerido: VSCode, Thonny, IDLE o cualquier IDE para Python

## 3. INSTALACIÓN Y CONFIGURACIÓN

Para poner en funcionamiento el software TorresHanoi v1.0, debes asegurarte de que los siguientes componentes estén correctamente instalados en tu equipo:

### 3.1. Instalar Python

- Ve a <https://www.python.org/downloads/>
- Descarga e instala Python para tu sistema operativo.
- Marca la opción “Add Python to PATH” durante la instalación.
- Verifica la instalación con el comando:

### 3.2. Ejecutar la Aplicación

- Guarde el código en un archivo llamado TorresHanoi.py.
- Abre una terminal o consola
- Dirígete a la carpeta donde está el archivo
- Ejecutar el programa

```
python TorresHanoi.py #código fuente
```

TorresHanoi.exe archivo ejecutable

mysetup.exe archivo instalador de la aplicación

CONTACTO DEL DESARROLLADOR: +51951679658

### 3.3. Configuración opcional

**TorresHanoi v1.0** Está diseñado para funcionar con una resolución mínima de 1024x768 píxeles. Si prefieres usar una resolución más alta, puedes ajustar el tamaño de la ventana directamente en el código.

- Abre el archivo **TorresHanoi.py** en un editor de texto.
- Busca la línea donde se establece el tamaño de la ventana (en el código proporcionado, es `root.geometry("1200x700")`)
- Cambia las dimensiones de acuerdo con tu necesidad.

### 3.4. Actualización

- Para actualizar TorresHanoi v.1.0
- Verificar si existe una nueva versión disponible en el sitio oficial o repositorio.
- Descargar la versión más reciente.
- Reemplazar los archivos antiguos.
- Ejecutar nuevamente la aplicación.

## 4. DESCRIPCIÓN DE LA APLICACIÓN TorresHanoi v1.0

El juego de las Torres de Hanói fue inventado por el matemático francés Édouard Lucas en el año de 1883.

El objetivo del juego de las Torres de Hanoi es mover todos los discos de la torre 1 (origen) a la torre 3 (destino) considerando la torre 2 como una torre auxiliar para mover los discos. Teniendo en cuenta las siguientes reglas: 1. Mover solo un disco a la vez. 2. No se puede ubicar un disco grande sobre un disco pequeño. 3. Solo puede usar la torre 2 como torre auxiliar para el desplazamiento de discos.

Las Torres de Hanoi es un juego de entrenamiento, generalmente se usa en la enseñanza de matemáticas e informática. Además, en programación se usa para ilustrar el algoritmo recursivo. El juego de las Torres de Hanoi necesita concentración y orden en la toma de decisiones.

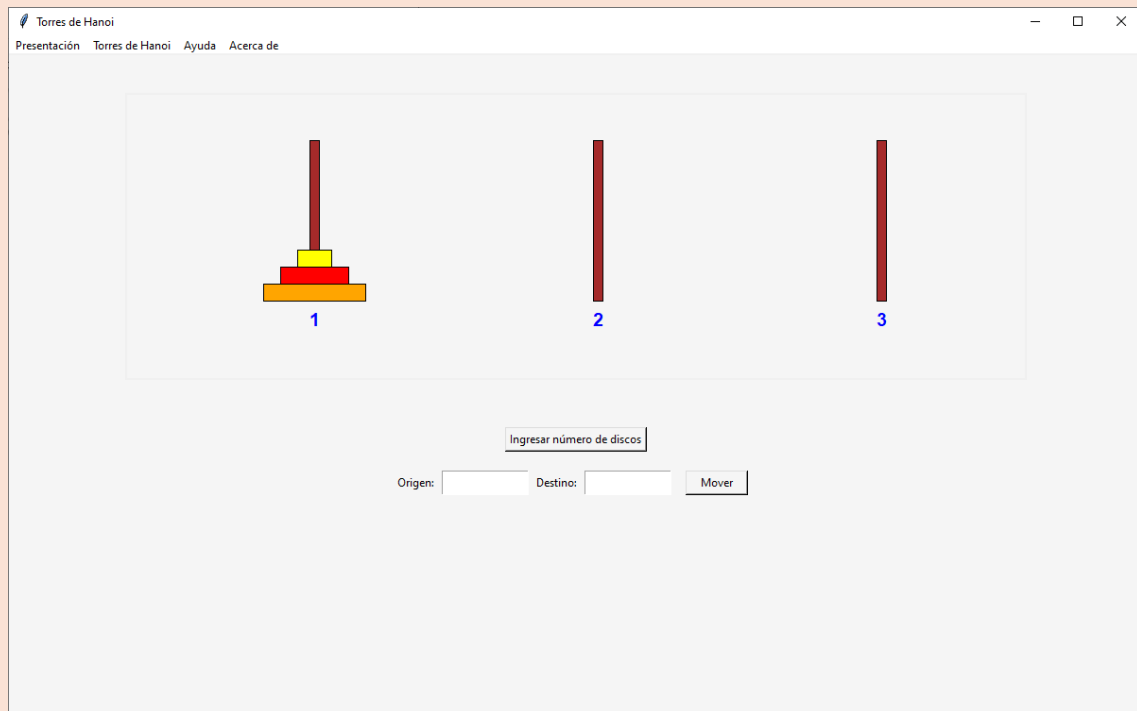
### 4.1. Menú Presentación

Al iniciar la aplicación de las TorresHanoi v1.0, los usuarios acceden a un menú de presentación que incluye un mensaje de bienvenida, el logotipo distintivo y el nombre destacado del programa. Esta sección tiene como objetivo familiarizar al usuario con la interfaz y con la finalidad principal del software, brindándole una visión general de sus funciones antes de comenzar a mover los discos.



### 4.2. Torres de Hanoi

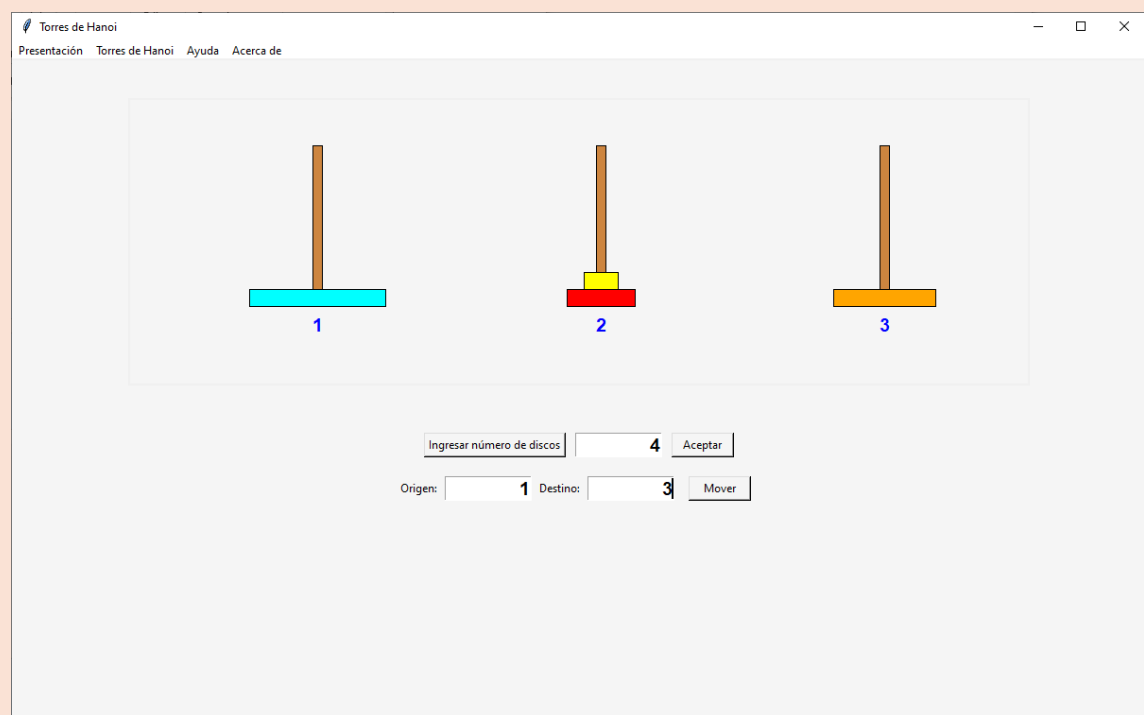
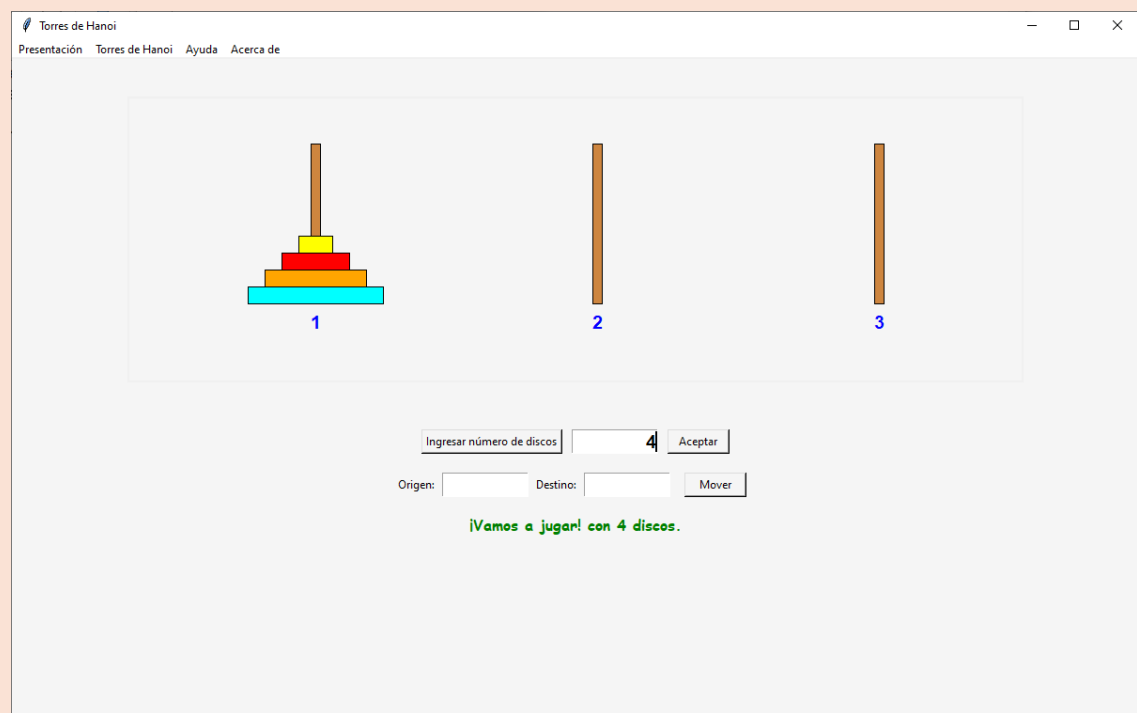
El menú de las "Torres de Hanoi" de *TorresHanoi v1.0* es la sección central de la aplicación, diseñada para interactuar con las tres torres y los discos. En primer lugar, el usuario debe ingresar el número de discos que desea mover el disco de la torre 1 a la torre 3, considerando que la torre 2 es una torre auxiliar para el desplazamiento de discos. Además, un disco grande no puede estar sobre un disco pequeño.

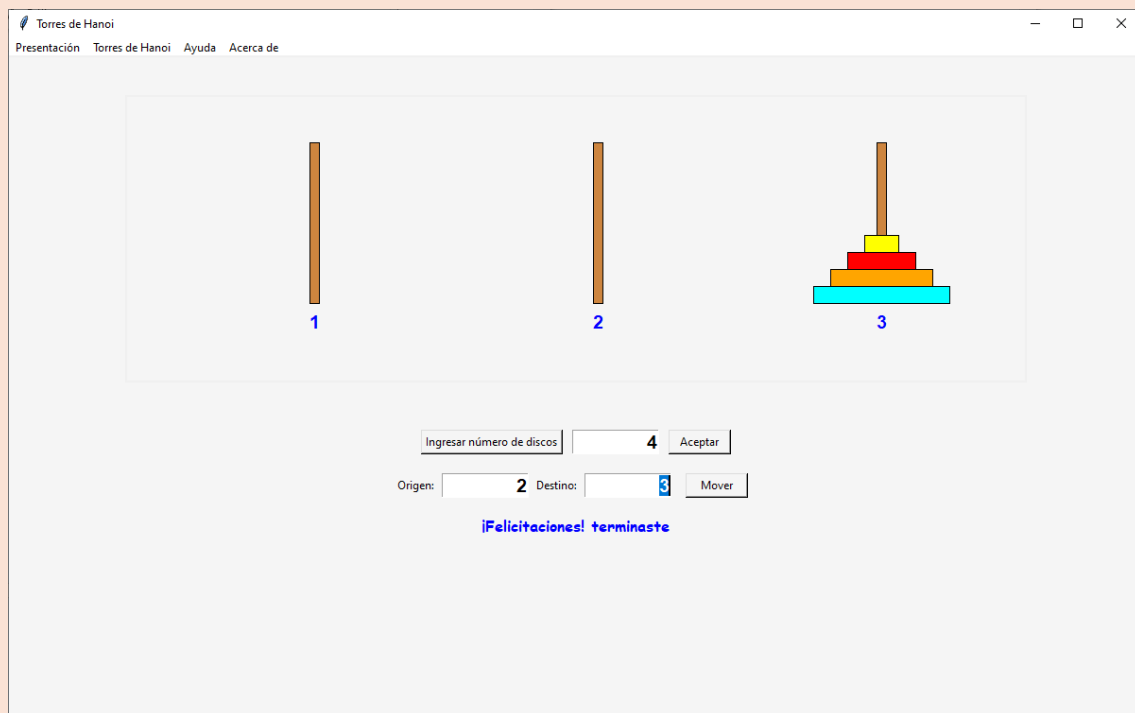


Al ingresar número de discos (escribir 4), haga clic en el botón 'Aceptar'. Esto generará en la primera torre los cuatros discos de mayor a menor. Luego podrá realizar desplazamiento de discos de la torre 1 a la torre 3 considerando la torre 2 como auxiliar para los movimientos. Además, debe considerar que un disco de mayor tamaño no puede estar sobre un disco pequeño,

Realizar lo desplazamientos de los discos considerando el origen la torre 1 y destino la torre 3.





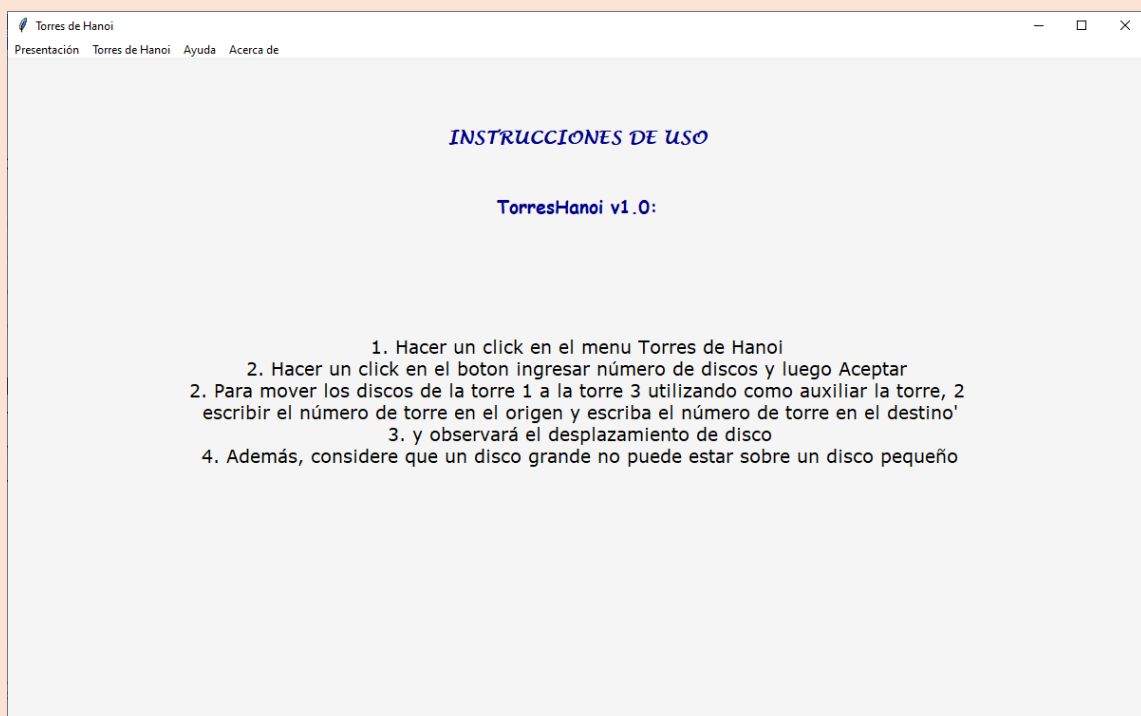


Al finalizar el desplazamiento de los discos queda como la muestra de la torre 3 y se realizó el desplazamiento con cuatro discos. El sistema solo considera 8 discos para el desplazamiento.

### 4.3. Menú Ayuda

El apartado "Ayuda" del software TorresHanoi v1.0 ha sido creado como una herramienta accesible y práctica, pensada para asistir a los estudiantes en la comprensión de su funcionamiento. Esta guía proporciona explicaciones claras y directas, con el fin de facilitar el aprendizaje y el manejo adecuado del programa.

Su principal finalidad es ofrecer orientación de manera sencilla, permitiendo que los usuarios se familiaricen con las funciones esenciales de la aplicación. Gracias a esto, los estudiantes pueden utilizar de forma efectiva las distintas herramientas que ofrece el software y aprovechar al máximo sus beneficios.

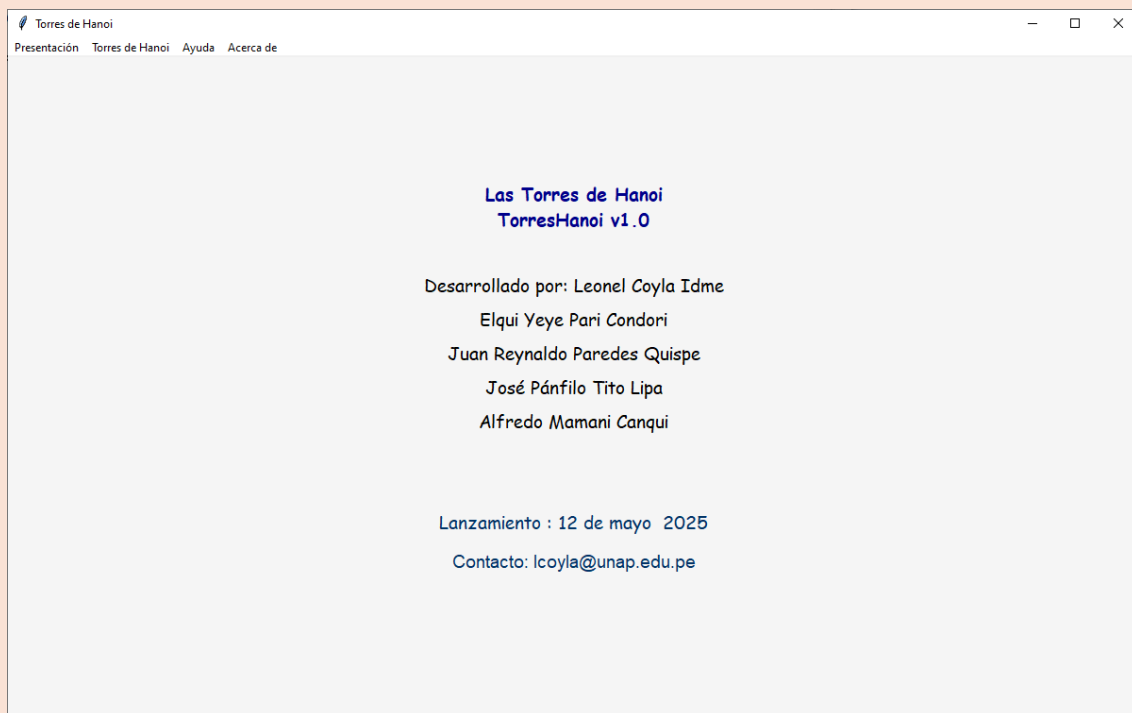


### 4.4. Menú Acerca de

En la sección "Acerca de", los usuarios pueden acceder a información esencial sobre TorresHanoi v1.0, una aplicación educativa creada con el propósito de mover discos de la torre 1 a la torre 3 utilizando como torre temporal la torre 2. Además, hay que considerar que un disco grande no puede estar sobre un disco pequeño. Esta área del programa presenta datos como el nombre completo del software, la versión actual, el desarrollador principal identificado como Leonel Coyla Idme, la fecha de lanzamiento y los medios de contacto disponibles para ofrecer soporte, recibir sugerencias o atender consultas.

El propósito de esta sección es ofrecer claridad sobre la autoría y el origen del programa, lo que contribuye a fortalecer la confianza de los usuarios al momento de utilizar la herramienta. Al proporcionar estos datos, se busca reforzar la transparencia del proyecto y demostrar su carácter confiable y educativo.

Además, este espacio establece un vínculo directo con el desarrollador, permitiendo que los usuarios puedan resolver inquietudes, enviar comentarios o realizar preguntas de manera sencilla. De este modo, estudiantes, docentes y familias pueden conocer con precisión el origen de la aplicación y asegurarse de que están utilizando una plataforma confiable, creada específicamente para apoyar procesos de aprendizaje.



## **5. PREGUNTAS FRECUENTES (FAQS) SOBRE EL SOFTWARE**

### **5.1. ¿Qué es Torres de Hanoi y cuál es su propósito?**

La Torres de Hanoi es un problema matemático clásico de tipo recursivo que fue propuesto por el matemático francés Édouard Lucas en 1883. Consiste en mover una serie de discos de diferentes tamaños desde una torre 1 (origen) hasta una torre 3 (destino), utilizando una torre 2 como auxiliar, siguiendo tres reglas fundamentales:

- Solo se puede mover un disco a la vez.
- Solo se puede mover el disco superior de una torre.
- Nunca se puede colocar un disco más grande sobre uno más pequeño.

El propósito del problema es trasladar todos los discos desde la torre de origen a la torre destino respetando las reglas mencionadas. Aunque parece sencillo, el número de movimientos necesarios crece exponencialmente con la cantidad de discos, lo que lo convierte en un excelente ejercicio para enseñar y practicar pensamiento recursivo, lógica algorítmica y estructura de datos. Es ampliamente utilizado en la enseñanza de programación y algoritmos.

### **5.2. ¿Cuáles son los requisitos mínimos del sistema para ejecutar?**

El aplicativo es compatible con diversos sistemas operativos, incluyendo Windows (versiones 7, 8, 10 y 11, siendo recomendable Windows 10 o superior), macOS (desde la versión 10.14 Mojave en adelante) y distribuciones de Linux como Ubuntu 18.04 o superior, Debian, Fedora, entre otras. En cuanto a los requisitos de hardware, se necesita un procesador Intel o AMD de al menos 1 GHz, un mínimo de 2 GB de memoria RAM (recomendado 4 GB), 100 MB de espacio libre en disco y una pantalla con resolución de 1024x768 píxeles o superior. Para su correcto funcionamiento, también se requiere tener instalado Python en su versión 3.8 o superior, junto con Tkinter (incluido por defecto en la instalación oficial de Python), y se sugiere utilizar editores como VSCode, Thonny, IDLE u otro entorno de desarrollo compatible con Python.

### **5.3. ¿Cómo instalo o configuro TorresHanoi?**

El software TorresHanoi v1.0 está optimizado para funcionar con una resolución mínima de 1024x768 píxeles, lo cual garantiza su correcto desempeño en la mayoría de las pantallas. Sin embargo, si el usuario prefiere trabajar con una resolución más alta, el tamaño de la ventana puede modificarse fácilmente accediendo al código del programa.

Para realizar este cambio, se debe abrir el archivo TorresHanoi.py en un editor de texto y buscar la línea donde se establece el tamaño de la interfaz, que normalmente aparece como `root.geometry("1200x700")`. A partir de allí, es posible personalizar las dimensiones de la ventana según las necesidades del usuario.

#### **5.4. ¿Qué se debe realizar si el software no funciona correctamente?**

En caso de que TorresHanoi v1.0 presente fallos en su funcionamiento, lo primero que se debe verificar es que todas las bibliotecas necesarias —como `tkinter`, `pillow`— estén correctamente instaladas, y que se esté utilizando Python en su versión 3.8 o superior. Además, es fundamental asegurarse de que no haya errores de sintaxis en el ingreso de datos y que el archivo de imagen `torresHanoi.png` se encuentre en la misma carpeta donde se ejecuta el programa.

Si luego de estas comprobaciones el problema persiste, se recomienda reiniciar tanto la aplicación como el sistema operativo. En caso de que los errores continúen, es aconsejable contactar al desarrollador principal a través del correo electrónico indicado en la sección "Acerca de" del programa ([lcoyla@unap.edu.pe](mailto:lcoyla@unap.edu.pe)), detallando el problema, incluyendo capturas de pantalla o mensajes de error que puedan ayudar a identificar la causa del inconveniente.

### **6. ADVERTENCIAS Y SUGERENCIAS DEL SOFTWARE DE APLICACIÓN.**

#### **6.1. Advertencias**

Es fundamental considerar que TaorresHanoi v1.0 ha sido desarrollado con un enfoque exclusivamente educativo, por lo tanto, el software de las Torres de Hanoi es un juego educativo que se utiliza para razonar los desplazamientos de los discos de la torre 1 a la torre 3 usando la torre 2 como auxiliar. Además, hay que considerar que no puede estar un disco grande sobre uno pequeño.

Además, se aconseja no eliminar ni renombrar el archivo de imagen `TorresHanoi.png`, ya que esto podría afectar la visualización adecuada dentro de la sección de presentación del software. Para garantizar un funcionamiento

óptimo, se recomienda ejecutar la aplicación en sistemas operativos compatibles y mantener actualizadas todas las bibliotecas requeridas.

## **6.2. Sugerencias**

Se sugiere que los usuarios experimenten y practiquen con diversos números de discos para aplicar el desplazamiento de discos de la torre 1 (origen) a la torre 3 (destino) teniendo como torre auxiliar la torre 2. Además, debe considerar que un disco grande no puede estar sobre un disco de tamaño pequeño. Esta práctica no solo consolida los conocimientos teóricos, sino que también fomenta el desarrollo de habilidades.

## 7. Código

```
1 import tkinter as tk
2 from PIL import ImageTk
3 from PIL import Image
4
5 class torresHanoi:
6     def __init__(self, root, contenido):
7         self.root = root
8         self.contenido = contenido
9
10        # Limpiar el frame
11        for widget in self.contenido.winfo_children():
12            widget.destroy()
13
14        self.nDiscos = 3 #Inicia con 3 discos
15        self.torres = [list(reversed(range(1, self.nDiscos+1))), [], []]
16
17        self.canvas = tk.Canvas(self.contenido, width=950, height=300, bg="White Smoke") #area de dibujo
18        self.canvas.pack(pady=40, anchor = "center")
19
20
21        # Frame para número de discos (antes que origen/destino)
22        self.frameDiscos = tk.Frame(self.contenido, bg="White Smoke")
23        self.frameDiscos.pack(pady=10)
24
25        tk.Button(self.frameDiscos, text="Ingresar número de discos",
26                bg="White Smoke", command=self.entrarDiscos).pack(side="left", padx=5)
27        self.entryDiscos = None # Se crea al presionar el botón
28
29        # Botones de movimiento (origen, destino, mover)
30        self.botones = tk.Frame(self.contenido, bg="White Smoke")
31        self.botones.pack(pady=10)
32
33        self.origenVar = tk.StringVar()
34        self.destinoVar = tk.StringVar()
35
36        tk.Label(self.botones, text="Origen:", bg="White Smoke").grid(row=0, column=0)
37        tk.Entry(self.botones, textvariable=self.origenVar, width=8, justify='right',
38                font=('Arial', 14, 'bold')).grid(row=0, column=1, padx=5)
39
40        tk.Label(self.botones, text="Destino:", bg="White Smoke").grid(row=0, column=2)
41        tk.Entry(self.botones, textvariable=self.destinoVar, width=8, justify='right',
42                font=('Arial', 14, 'bold')).grid(row=0, column=3, padx=5)
43
44        tk.Button(self.botones, text="Mover", bg="White Smoke",
45                width=8, command=self.mover).grid(row=0, column=4, padx=10)
46
47        self.mensaje_label = tk.Label(self.contenido, text="", bg="White Smoke",
48                fg="red", font=("Arial", 12))
49        self.mensaje_label.pack(pady=5)
50
51        self.dibujarTorres()
52
53        def entrarDiscos(self):
54            # Evitar múltiples entradas y botones
55            for widget in self.frameDiscos.winfo_children():
56                if type(widget) in [tk.Entry, tk.Button]:
57                    if isinstance(widget, tk.Entry) or widget.cget("text") == "Aceptar":
58                        widget.destroy()
59
60            self.entryDiscos = tk.Entry(self.frameDiscos, width=8, justify='right', font=('Arial', 14, 'bold'))
```



```

61 self.entryDiscos.pack(side="left", padx=5)
62
63 tk.Button(self.frameDiscos, text="Aceptar", bg="White Smoke",
64           width=8, command=self.aceptarDiscos).pack(side="left", padx=5)
65
66 def aceptarDiscos(self):
67     try:
68         n = int(self.entryDiscos.get())
69         if n < 1 or n > 8:
70             raise ValueError("El número debe estar entre 1 y 8")
71
72         self.nDiscos = n
73         self.torres = [list(reversed(range(1, n + 1))), [], []]
74         self.dibujarTorres()
75         self.mostrarMensaje(f"¡Vamos a jugar! con {n} discos.", "green")
76     except ValueError:
77         self.mostrarMensaje("Ingrese un número válido entre 1 y 8 discos.")
78
79 def dibujarTorres(self):
80     self.canvas.delete("all")
81
82     alturaDisco = 18
83     distanciaEntreTorres = [200, 500, 800]
84     self.colores = ["yellow", "red", "orange", "cyan", "blue", "Green Yellow", "magenta", "Lime"]
85
86     for t, torre in enumerate(self.torres):
87         x = distanciaEntreTorres[t]
88         self.canvas.create_rectangle(x - 5, 50, x + 5, 220, fill="peru")
89         y = 220
90         for disco in torre:
91             anchoDisco = disco * 18
92             color = self.colores[(disco-1)%len(self.colores)]
93             self.canvas.create_rectangle(x - anchoDisco, y - alturaDisco, x + anchoDisco, y,
94                                         fill=color, outline="black")
95             y = y - alturaDisco
96             self.canvas.create_text(x, 240, text=str(t + 1), font=("Helvetica", 14, "bold"), fill="blue")
97
98 def algoritmoHanoi(self, n, origen, destino, temporal):
99     if n == 1:
100         self.moverDisco(origen, destino)
101     else:
102         self.algoritmoHanoi(n - 1, origen, temporal, destino) #Algoritmo recursivo torres de Hanoi
103         self.moverDisco(origen, destino)
104         self.algoritmoHanoi(n - 1, temporal, destino, origen) #Algoritmo recursivo torres de Hanoi
105
106 def moverDisco(self, origen, destino):
107     if self.torres[origen]:
108         disco = self.torres[origen][-1]
109         if not self.torres[destino] or self.torres[destino][-1] > disco:
110             self.torres[origen].pop()
111             self.torres[destino].append(disco)
112             self.dibujarTorres()
113             if len(self.torres[2]) == self.nDiscos:
114                 self.mostrarMensaje("¡Felicitaciones! terminaste", "blue")
115             else:
116                 self.mostrarMensaje("Movimiento no permitido: No puede estar un disco grande sobre uno más")
117         else:
118             self.mostrarMensaje("Movimiento incorrecto: Torre de origen vacía.")
119
120

```

```

121 def mover(self):
122     self.mostrarMensaje("") # Limpiar mensaje anterior
123     try:
124         origen = int(self.origenVar.get()) - 1
125         destino = int(self.destinoVar.get()) - 1
126         if origen not in [0, 1, 2] or destino not in [0, 1, 2]:
127             raise ValueError
128         self.moverDisco(origen, destino)
129     except ValueError:
130         self.mostrarMensaje("Ingrese números válidos (1, 2 o 3).")
131
132
133 def mostrarMensaje(self, texto, color="red"):
134     self.mensaje_label.config(
135         text=texto,
136         fg=color,
137         font=("Comic Sans MS", 12, "bold")
138     )
139
140
141
142 # Aplicación principal con menú
143 class App:
144     def __init__(self, root):
145         self.root = root
146         self.root.title("Torres de Hanoi")
147         self.root.geometry("1200x700")
148
149         self.barra_menu = tk.Menu(self.root)
150         self.root.config(menu=self.barra_menu)
151
152         self.barra_menu.add_command(label="Presentación", command=self.presentacion)
153         self.barra_menu.add_command(label="Torres de Hanoi", command=self.torresHanoi)
154         self.barra_menu.add_command(label="Ayuda", command=self.ayuda)
155         self.barra_menu.add_command(label="Acerca de ", command=self.acercade)
156
157         # Frame central
158         self.contenido = tk.Frame(self.root, bg="White Smoke")
159         self.contenido.pack(fill="both", expand=True)
160
161     def limpiarFrame():
162         for widget in contenido.winfo_children():
163             widget.destroy()
164
165     def presentacion(self):
166
167         for widget in self.contenido.winfo_children():
168             widget.destroy()
169
170         tk.Label(self.contenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="White Smoke").pack()
171         tk.Label(self.contenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="White Smoke").pack()
172         tk.Label(self.contenido, text="BIENVENIDOS A LA APLICACIÓN",
173             font=("Comic Sans MS", 18, "bold"), fg="#00008B", bg="White Smoke").pack()
174         tk.Label(self.contenido, text="LAS TORRES DE HANOI",
175             font=("Arial", 14, "bold"), fg="#00008B", bg="White Smoke").pack()
176         tk.Label(self.contenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="White Smoke").pack()
177         tk.Label(self.contenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="White Smoke").pack()
178
179         try:
180             image = Image.open("TorresHanoi.png")

```

```

181         image = image.resize((500,350))
182         imageTk = ImageTk.PhotoImage(image)
183         imageLabel = tk.Label(self.contenido, image=imageTk, bg="White Smoke")
184         imageLabel.image = imageTk
185         imageLabel.pack(pady=20)
186     except Exception as e:
187         print(f"Error al cargar la imagen: {e}")
188
189     tk.Label(self.contenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="White Smoke").pack()
190     tk.Label(self.contenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="White Smoke").pack()
191     tk.Label(self.contenido, text="TorresHanoi v1.0", font=("Comic Sans MS", 20, "bold"),
192             fg="#00008B", bg="White Smoke").pack()
193
194     def torresHanoi(self):
195         torresHanoi(self.root, self.contenido)
196
197     def ayuda(self):
198
199         for widget in self.contenido.winfo_children():
200             widget.destroy()
201
202         tk.Label(self.contenido, text="\n\nINSTRUCCIONES DE USO", bg="White Smoke",
203                 font=("Lucida Handwriting", 14, "bold"), fg="#00008B").pack(pady=20)
204         tk.Label(self.contenido, text="TorresHanoi v1.0:\n\n", bg="White Smoke",
205                 font=("Comic Sans MS", 14, "bold"), fg="#00008B").pack(pady=20)
206         tk.Label(self.contenido, text="\n1. Hacer un click en el menu Torres de Hanoi"
207                 f"\n2. Hacer un click en el boton ingresar número de discos y luego Aceptar\n"
208                 f"2. Para mover los discos de la torre 1 a la torre 3 utilizando como auxiliar la torre, 2\n"
209                 f"escribir el número de torre en el origen y escriba el número de torre en el destino"
210                 f"\n3. y observará el desplazamiento de disco"
211                 f"\n4. Además, considere que un disco grande no puede estar sobre un disco pequeño",
212                 bg="White Smoke",
213                 font=("Verdana", 14)).pack(pady=20)
214
215     def acercade(self):
216         for widget in self.contenido.winfo_children():
217             widget.destroy()
218
219
220         frameTextos = tk.Frame(self.contenido, bg="White Smoke")
221         frameTextos.pack(pady=20)
222
223         # Título
224         tk.Label(frameTextos, text="\n\n\n\nJuego de las Torres de Hanoi\nTorresHanoi v1.0",
225                 bg="White Smoke", font=("Comic Sans MS", 14, "bold"),
226                 fg="#00008B").pack(pady=(0, 10))
227
228         # Lista de autores
229         autores = [
230             "\nDesarrollado por: Leonel Coyla Idme",
231             "Elqui Yeye Pari Condori",
232             "Juan Reynaldo Paredes Quispe",
233             "José Pánfilo Tito Lipa",
234             "Alfredo Mamani Canqui",
235         ]
236
237         # Etiqueta por autor
238         for autor in autores:
239             tk.Label(frameTextos, text=autor, bg="White Smoke",
240                     font=("Comic Sans MS", 14)).pack(pady=20)

```

```

241
242     labelAcercade = tk.Label(self.contenido, text= "\n\nLanzamiento : 12 de mayo 2025",
243                             font=("Comic Sans MS", 14),fg="#003366",bg="White Smoke")
244     labelAcercade.pack(pady=(1,10))
245     labelAcercade = tk.Label(self.contenido, text= "Contacto: lcoyla@unap.edu.pe",
246                             font=("Comic Sans MS", 14),fg="#003366",bg="White Smoke")
247     labelAcercade.pack(pady=(1,10))
248
249
250 # Ejecutar aplicación
251 def main():
252     root = tk.Tk()
253     app = App(root)
254     root.mainloop()
255 if __name__ == "__main__":
256     main()
257

```