

SOFTWARE DE APLICACIÓN

PLANO CARTESIANO

Bienvenido a la Aplicación Educativa

PLANO CARTESIANO



CartesKids

MANUAL DE USUARIO

SOFTWARE DE APLICACIÓN

KartesKids v1.0: Software de dibujo de pares ordenados en un plano cartesiano

AUTORES:

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

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	6
3.1.	INSTALAR PYTHON	6
3.2.	VERIFICAR LA INSTALACIÓN.....	6
3.3.	EJECUTAR LA APLICACIÓN	6
3.4.	CONFIGURACIÓN OPCIONAL	7
3.5.	SOLUCIÓN DE PROBLEMA.....	7
3.6.	ACTUALIZACIÓN	7
4.	DESCRIPCIÓN DE LA APLICACIÓN CARTESKIDSV1.0	8
4.1.	MENÚ PRESENTACIÓN	8
4.3.	MENÚ AYUDA	12
4.4.	MENÚ ACERCA DE	13
5.	PREGUNTAS FRECUENTES (FAQS) SOBRE EL SOFTWARE	14
5.1.	¿QUÉ ES CARTESKIDS Y CUÁL ES SU PROPÓSITO?	14
5.2.	¿CUÁLES SON LOS REQUISITOS MÍNIMOS DEL SISTEMA PARA EJECUTAR?	14
5.3.	¿CÓMO INSTALO O CONFIGURO CARTESKIDS?	14
5.4.	¿QUÉ SE DEBE REALIZAR SI EL SOFTWARE NO FUNCIONA CORRECTAMENTE?	15
6.	ADVERTENCIAS Y SUGERENCIAS DEL SOFTWARE DE APLICACIÓN.	15
6.1.	ADVERTENCIAS.....	15
6.2.	SUGERENCIAS	15
7.	CÓDIGO	17

1. INTRODUCCIÓN

****CartesKids v1.0**** es un software educativo diseñado especialmente para apoyar el aprendizaje de los pares ordenados y el uso del plano cartesiano en estudiantes de educación primaria. A través de una interfaz amigable y colorida, los niños y niñas pueden visualizar y ubicar fácilmente puntos en el plano, relacionando conceptos matemáticos con experiencias interactivas y divertidas. Este recurso busca despertar el interés por las matemáticas desde edades tempranas, facilitando la comprensión de temas que pueden parecer abstractos al principio.

Una de las características principales de ****PlanoCartesiano v1.0**** es la posibilidad de ingresar pares ordenados (x, y) y observar en tiempo real cómo se ubican en el plano cartesiano. Además, el software utiliza colores llamativos, etiquetas claras y una estructura visual sencilla, ideal para captar la atención de los estudiantes y mantenerlos motivados. También se pueden, eliminar o agregar nuevos puntos, lo que permite practicar con libertad y reforzar el aprendizaje a través de la exploración.

El uso de ****CartesKids v1.0**** en el aula o en casa puede servir como complemento a las clases tradicionales, permitiendo a los docentes o padres acompañar el proceso educativo de forma más dinámica. Los alumnos pueden trabajar individualmente o en grupo, compartiendo estrategias y descubriendo patrones, lo cual promueve la participación activa y el pensamiento lógico. La herramienta también ofrece un espacio seguro donde cometer errores forma parte natural del aprendizaje.

En resumen, ****CartesKids v1.0**** es una herramienta didáctica valiosa que combina tecnología, matemáticas y creatividad para enriquecer el aprendizaje de los estudiantes de primaria. Su propósito es que los niños no solo aprendan a ubicar puntos en un plano, sino que también se sientan motivados a seguir explorando el mundo de las matemáticas con entusiasmo y confianza.

PROYECTO REGISTRADO CON URL:

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

2. REQUISITOS DEL SISTEMA

Para garantizar un rendimiento óptimo de CartesKids v1.0, el software educativo diseñado para ayudar a los estudiantes de educación primaria a comprender los conceptos de coordenadas y gráficos en el plano cartesiano es fundamental que tu sistema cumpla con los requisitos mínimos y recomendados. Asegúrate de que tanto el hardware como el software de tu equipo estén correctamente configurados antes de instalar y utilizar el programa.

2.1. Sistema Operativo

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

- Windows 7, 8, 10 o superior
- macOS 10.13 High Sierra o superior
- Linux (distribuciones con entorno gráfico como Ubuntu, Mint, Fedora)

2.2. Hardware

- Procesador: Intel Core i3 o superior / AMD equivalente
- Memoria RAM: Mínimo 2 GB (recomendado 4 GB o más)
- Almacenamiento: Al menos 200 MB libres para dependencias de Python e imágenes
- Resolución de pantalla: 1024x768 o superior

2.3. Software

Es necesario contar con los siguientes programas y herramientas instalados antes de usar CartesKids v1.0

Python:

- Python: Versión 3.8 o superior
- Bibliotecas requeridas (pueden instalarse con pip):

`pip install matplotlib pillow`

- Estas bibliotecas se usan para:

- **tkinter:** Crear la interfaz gráfica
- **matplotlib:** Dibujar el gráfico en el plano cartesiano
- **Pillow:** Cargar y mostrar imágenes

- Archivos necesarios:

Archivo de imagen Puntos.png en la misma carpeta que el script.
(Este archivo se muestra en la presentación del software. Si no está presente, el programa seguirá funcionando, pero no se mostrará la imagen)

3. INSTALACIÓN Y CONFIGURACIÓN

Antes de poder ejecutar CartesKids v1.0, asegúrate de tener los siguientes requisitos instalados en tu sistema:

3.1. Instalar Python

- Descargue la última versión de Python desde
[python.org](https://www.python.org/downloads/).
- Continúa con las instrucciones de instalación adecuadas para su sistema operativo.

3.2. Verificar la Instalación

- Abra una terminal o línea de comandos y escriba:
`python --version`
- Debería ver la versión de Python instalada.

3.3. Ejecutar la Aplicación

- Guarde el código dado en un archivo con el nombre CartesKids.py.
- Abra la terminal, diríjase a la carpeta donde almacenó el archivo y ejecútelo.:
`python CartesKids.py`
- También es posible ejecutar el archivo CartesKids.exe directamente desde su ubicación.
- Otra opción es instalar el programa ejecutando mysetup.exe desde la carpeta donde se encuentra.

Archivo principal

CartesKids.py #código fuente

CartesKids.exe archivo ejecutable

mysetup.exe archivo instalador de la aplicación

CONTACTO DEL DESARROLLADOR: +51951679658

3.4. Configuración opcional

CartesKids v1.0 está configurado para funcionar con una resolución de pantalla mínima de **1024x768 px**. Si deseas usar una resolución mayor, puedes modificar el tamaño de la ventana en el código.

- Abre el archivo **plano_cartesiano.py** en un editor de texto.
- Busca la línea donde se establece el tamaño de la ventana (en el código proporcionado, es `ventana.geometry("650x850")`).
- Cambia las dimensiones a tu preferencia. Por ejemplo:

```
self.ventana.geometry("800x1000") # Ajusta el tamaño de la ventana
```

3.5. Solución de problema

Error al ejecutar el programa: Si obtienes un error que menciona que faltan bibliotecas, asegúrate de haber ejecutado `pip install matplotlib pillow` correctamente.

Imagen no cargada: Si no ves la imagen "Puntos.png" en la pantalla de presentación, asegúrate de que el archivo Puntos.png esté en la misma carpeta que el archivo `plano_cartesiano.py`.

3.6. Actualización

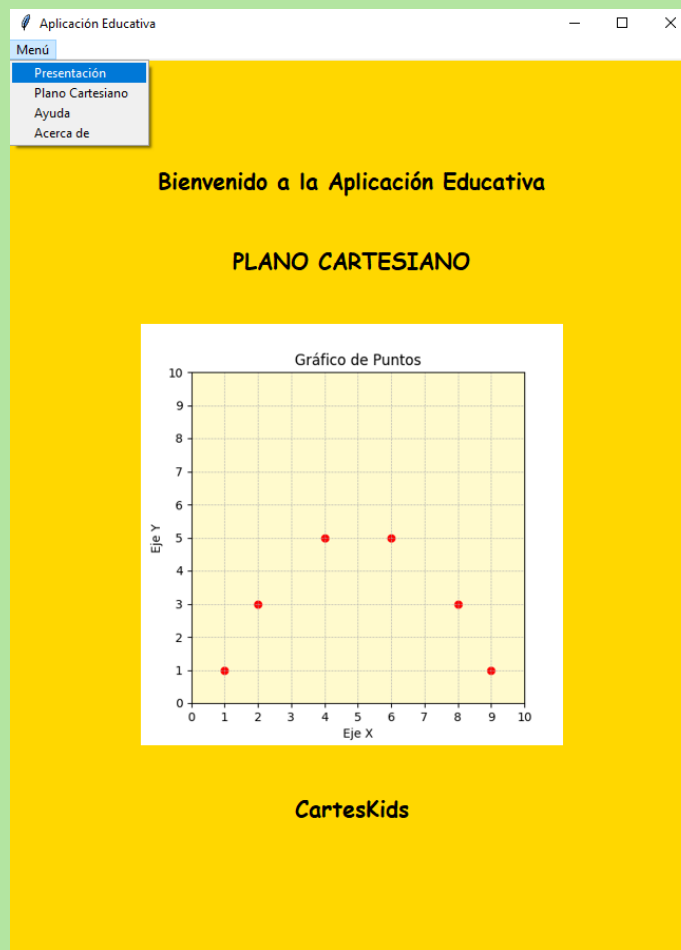
Para futuras actualizaciones de CartesKids v1.0, simplemente descarga la nueva versión de los archivos y reemplaza los existentes.

4. DESCRIPCIÓN DE LA APLICACIÓN CartesKidsv1.0

CartesKids v1.0 es una herramienta educativa diseñada para estudiantes de educación primaria que facilita la representación y comprensión de los pares ordenados en un plano cartesiano, mejorando así el aprendizaje de conceptos matemáticos de manera visual e interactiva.

4.1. Menú Presentación

Al iniciar CartesKids v1.0, los usuarios son recibidos en la pestaña de presentación, donde encontrarán una cálida bienvenida, el logo distintivo de la aplicación y su nombre destacado. Este espacio está diseñado para introducir al usuario a la interfaz y el propósito del software, permitiéndole familiarizarse con sus funciones antes de comenzar a graficar puntos en el plano cartesiano.

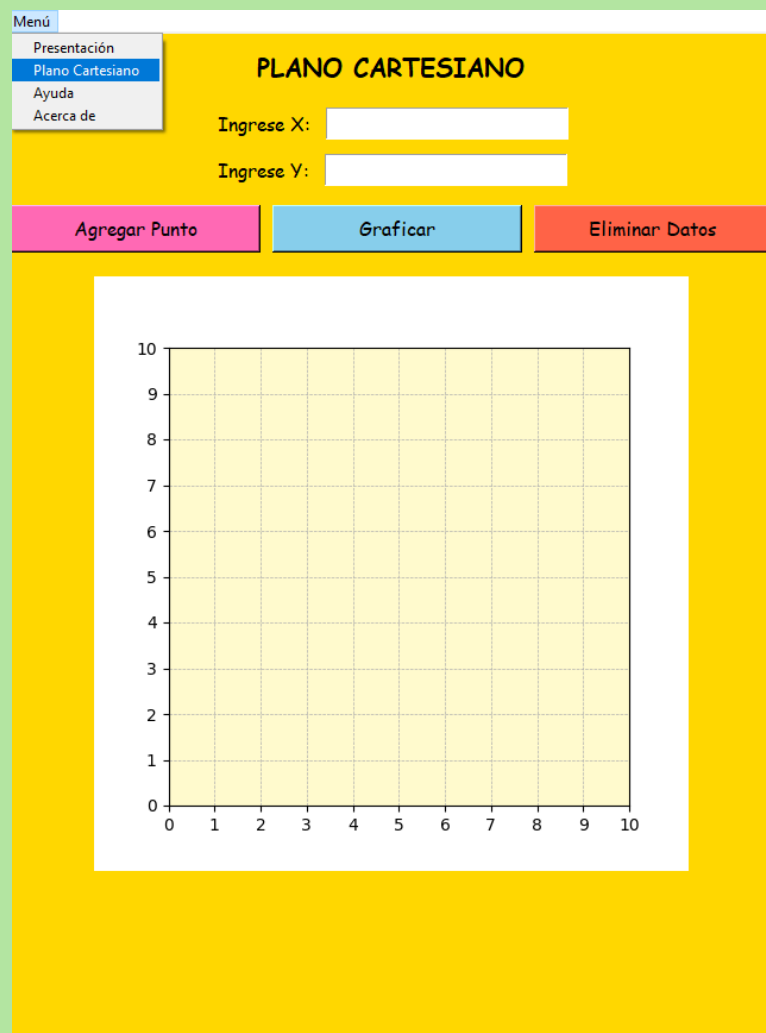


4.2. CartesKids v1.0

El menú Plano Cartesiano de CartesKids v1.0 permite a los estudiantes de educación primaria explorar y comprender la representación de puntos en un sistema de coordenadas de manera interactiva y sencilla.

En esta sección, los usuarios pueden ingresar valores para las coordenadas X e Y, agregar puntos al plano y visualizar su ubicación exacta en una cuadrícula. Además, el software ofrece opciones para graficar, eliminar datos y ajustar la visualización para un mejor aprendizaje.

El diseño colorido y la interfaz intuitiva facilitan la navegación y el uso del programa, fomentando un aprendizaje dinámico y atractivo. CartesKids v1.0 es una excelente herramienta para desarrollar habilidades matemáticas a través de la exploración y la práctica visual.

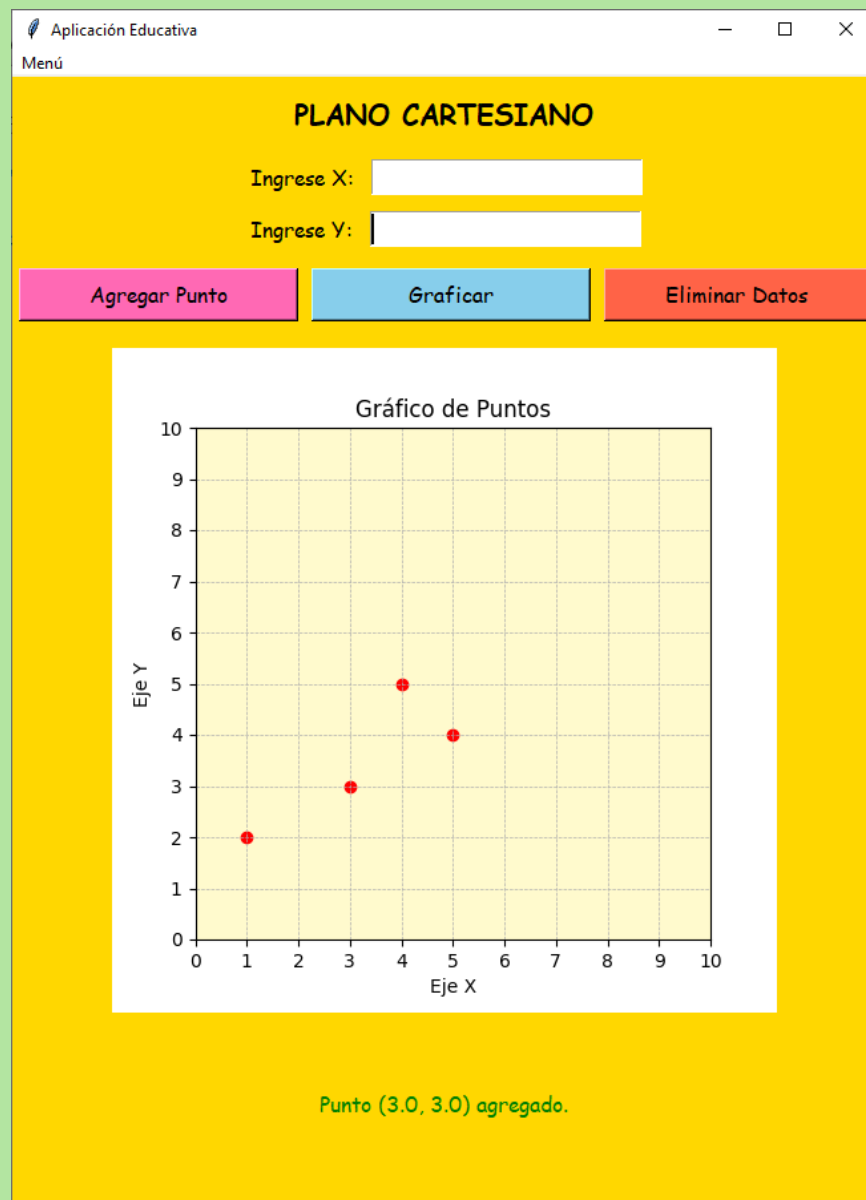


Ingreso de Datos

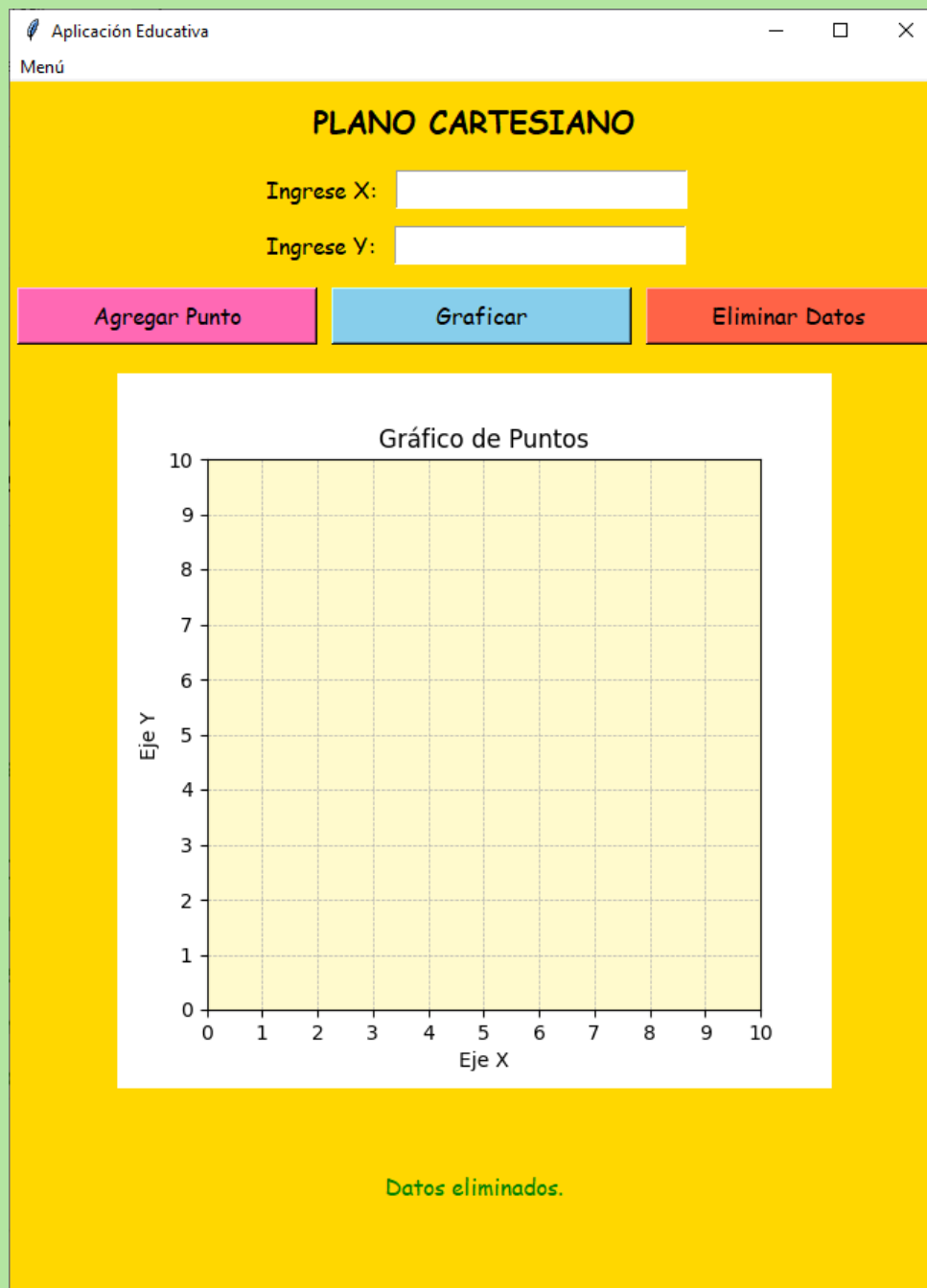
En esta sección, el estudiante podrá interactuar con el plano cartesiano de forma práctica y divertida. Para comenzar, debe ingresar un valor numérico en el campo "Ingrese X", seguido del valor correspondiente en "Ingrese Y", que representan las coordenadas de un punto en el plano. Los datos ingresados deben ser de 0 a 10 para los dos ejes.

Una vez escritos ambos valores, se debe presionar el botón "Agregar Punto", el cual almacena el par ordenado (X, Y) en la memoria del programa.

Cuando se han ingresado uno o varios puntos, el botón "Graficar" permite visualizar todos los puntos en el plano cartesiano, destacándolos con color rojo para facilitar su identificación.



Si se desea reiniciar la actividad o comenzar con nuevos datos, se puede utilizar el botón "Eliminar Datos", que borra todos los puntos ingresados y limpia el gráfico, permitiendo comenzar nuevamente.



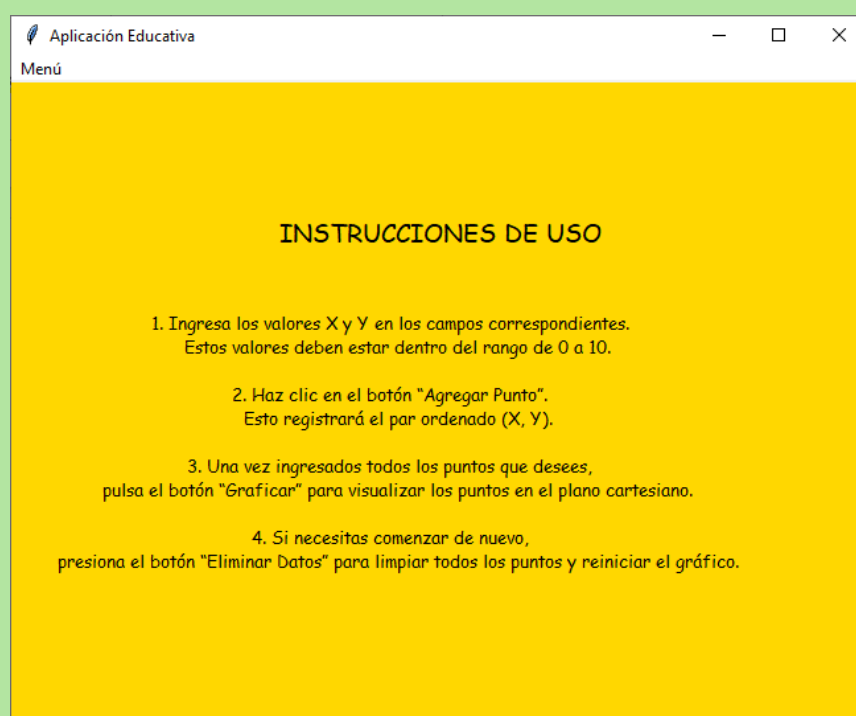
Esta funcionalidad ofrece a los estudiantes la posibilidad de experimentar y reforzar sus conocimientos matemáticos de forma visual, interactiva y amigable.

4.3. Menú Ayuda

El menú "Ayuda" ofrece una guía práctica y sencilla para que los estudiantes comprendan cómo utilizar correctamente el software CartesKids v1.0. Está pensado para acompañar paso a paso a los usuarios en el proceso de ingreso y visualización de puntos en un plano cartesiano.

En esta sección se explican de forma clara las acciones básicas: ingresar valores X y Y, agregar puntos, graficarlos y eliminar los datos si se desea empezar de nuevo. Todo está redactado con un lenguaje fácil de entender, ideal para niños de primaria que están aprendiendo sobre coordenadas.

Gracias a este menú, los estudiantes pueden usar el programa con confianza y sin complicaciones, desarrollando su autonomía mientras refuerzan sus conocimientos de matemáticas de forma visual e interactiva.

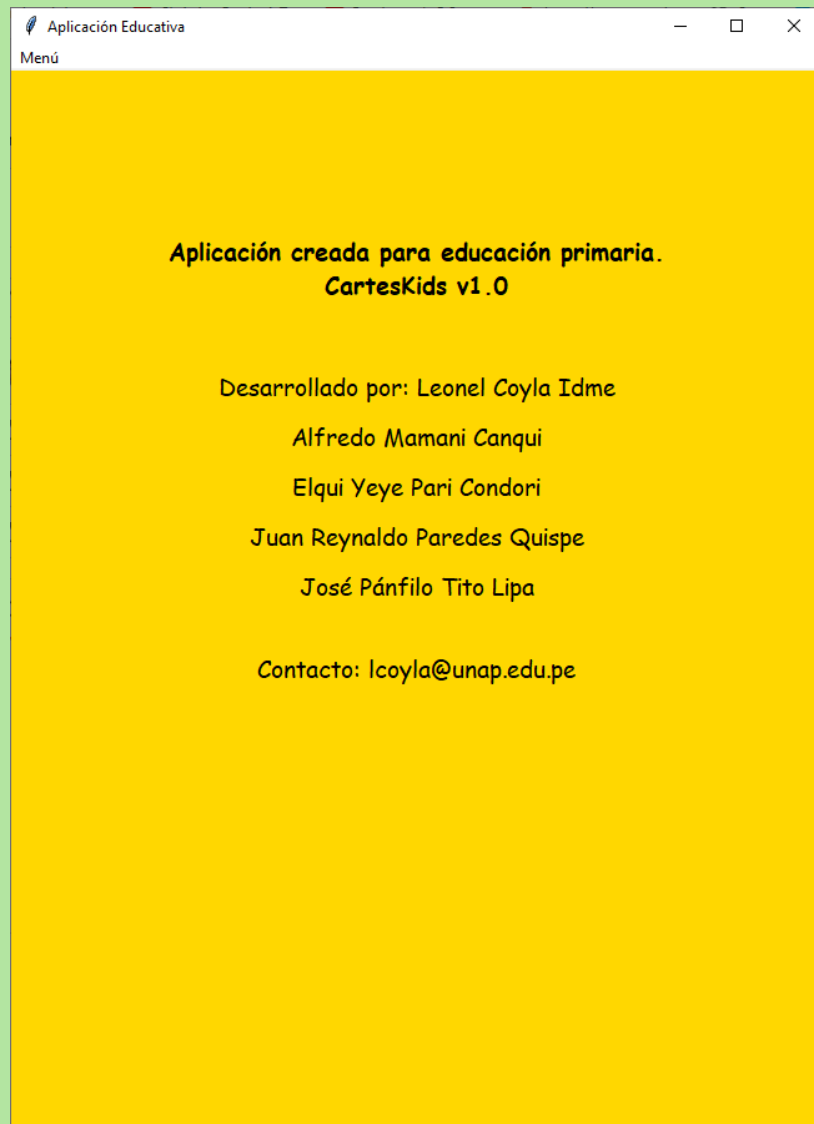


4.4. Menú Acerca de

En el menú "Acerca de", los usuarios encontrarán información esencial sobre CartesKids v1.0, la aplicación educativa diseñada para introducir a los niños al uso del plano cartesiano. Esta sección detalla el nombre completo del software (CartesKids v1.0), la versión actual, el nombre del desarrollador principal, Leonel Coyla Idme, la fecha de lanzamiento y los datos de contacto disponibles para consultas o soporte técnico.

Este espacio tiene como propósito ofrecer transparencia sobre la autoría y origen de la herramienta, reforzando así la confianza de los usuarios en su uso. Además, permite establecer una vía directa de comunicación con el desarrollador ante cualquier duda o sugerencia.

Gracias a esta sección, los estudiantes, docentes y padres pueden conocer de manera clara la procedencia del software y confirmar que están utilizando una herramienta confiable y diseñada con fines educativos específicos.



5. PREGUNTAS FRECUENTES (FAQS) SOBRE EL SOFTWARE

5.1. ¿Qué es CartesKids y cuál es su propósito?

CartesKids v1.0 es una aplicación educativa interactiva diseñada especialmente para estudiantes de educación primaria. Su propósito principal es facilitar la comprensión del plano cartesiano mediante la visualización y graficación de pares ordenados (X, Y) de forma sencilla y divertida.

A través de una interfaz amigable y colorida, los estudiantes pueden ingresar coordenadas, agregar puntos, verlos representados en un gráfico y eliminar datos según sea necesario. Esta herramienta permite reforzar conceptos matemáticos fundamentales como la ubicación en el plano, el uso de coordenadas y la interpretación visual de puntos en un sistema de ejes.

Además, CartesKids promueve el aprendizaje autónomo y el desarrollo del pensamiento lógico-matemático en un entorno lúdico, contribuyendo a una mejor comprensión de los contenidos curriculares en el área de Matemática.

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

Para ejecutar CartesKids v1.0 de manera adecuada, se requiere un sistema con Windows 7 o superior, un procesador de al menos 1.0 GHz, 2 GB de memoria RAM, y al menos 200 MB de espacio libre en disco. Además, es necesario tener instalada la versión más reciente de Python (3.10 o superior) junto con las bibliotecas tkinter, matplotlib y Pillow (PIL). Se recomienda una resolución mínima de pantalla de 1024x768 píxeles para una visualización óptima de la interfaz gráfica.

5.3. ¿Cómo instalo o configuro CartesKids?

Para instalar o configurar CartesKids v1.0, primero debe asegurarse de tener Python instalado en su equipo. Luego, copie el código fuente en un archivo llamado CartesKids.py y guárdelo en una carpeta de su preferencia. Desde la terminal o línea de comandos, navegue hasta esa carpeta y ejecute el programa utilizando `python CartesKids.py`. Si cuenta con un archivo ejecutable (CartesKids.exe), simplemente haga doble clic sobre él para iniciar la aplicación sin necesidad de configuración adicional. También puede utilizar un instalador (mysetup.exe) si está disponible, el cual guiará el proceso de instalación de forma automática.

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

Si CartesKids v1.0 no funciona correctamente, se recomienda primero verificar que su sistema cumpla con los requisitos mínimos, como tener Python y las bibliotecas necesarias (Tkinter, Matplotlib y PIL) correctamente instaladas. También es importante asegurarse de que el archivo de imagen requerido (por ejemplo, Puntos.png) esté ubicado en el mismo directorio del código. Si el problema persiste, reinicie la aplicación o su equipo. En caso de errores específicos, revise los mensajes en la consola para identificar posibles causas. Finalmente, si necesita ayuda adicional, puede contactar al desarrollador mediante el correo proporcionado en la sección “Acerca de” de la aplicación.

6. ADVERTENCIAS Y SUGERENCIAS DEL SOFTWARE DE APLICACIÓN.

6.1. Advertencias

Asegúrate de ingresar valores numéricos válidos dentro del rango permitido (de 0 a 10), ya que datos fuera de ese límite pueden generar errores en la visualización del gráfico. Además, es fundamental que todas las bibliotecas requeridas por CartesKids v1.0 (como Tkinter, Matplotlib y PIL) estén correctamente instaladas y actualizadas para garantizar el funcionamiento adecuado del software. Si experimentas inconvenientes, revisa cuidadosamente los mensajes de error y consulta la documentación incluida o recurre a foros especializados, donde otros usuarios pueden ayudarte a resolver problemas similares.

6.2. Sugerencias

Para aprovechar al máximo las funcionalidades de CartesKids v1.0, se recomienda utilizar el software en un entorno libre de distracciones, especialmente si se trabaja con niños en el aula. Supervisa el uso de la herramienta para asegurar que los estudiantes ingresen correctamente los valores numéricos y comprendan el significado de los pares ordenados.

Es útil preparar previamente una lista de coordenadas que los alumnos puedan graficar, lo cual facilitará el aprendizaje y permitirá un trabajo más organizado. Asimismo, se sugiere fomentar la participación activa, animando a los estudiantes a interpretar los puntos graficados en el plano cartesiano.

Por último, realiza una revisión periódica del software para asegurarte de que funciona correctamente y, de ser posible, mantén una copia de seguridad del archivo ejecutable y de instalación en caso de necesitar reinstalarlo.

7. Código

```
1 import tkinter as tk
2 from tkinter import Menu
3 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
4 import matplotlib.pyplot as plt
5 from matplotlib.ticker import MultipleLocator
6
7 from PIL import Image, ImageTk
8
9 class GraficoApp:
10     def __init__(self, contenedor):
11         self.contenedor = contenedor
12         self.puntos = []
13
14         # Título
15         self.sizeLabel = tk.Label(contenedor, text="PLANO CARTESIANO",
16                                   bg='#FFD700', font=("Comic Sans MS", 16, "bold"))
17         self.sizeLabel.pack(pady=10)
18
19         # Ingreso de datos de las variables X,Y
20         entradaFrame = tk.Frame(contenedor, bg='#FFD700')
21         entradaFrame.pack(pady=5)
22         tk.Label(entradaFrame, text="Ingreso X:", bg='#FFD700',
23                 font=("Comic Sans MS", 12)).pack(side=tk.LEFT, padx=5)
24         self.entry_x = tk.Entry(entradaFrame, font=("Comic Sans MS", 12))
25         self.entry_x.pack(side=tk.LEFT, padx=5)
26
27         entradaFrame_y = tk.Frame(contenedor, bg='#FFD700')
28         entradaFrame_y.pack(pady=5)
29         tk.Label(entradaFrame_y, text="Ingreso Y:", bg='#FFD700',
30                 font=("Comic Sans MS", 12)).pack(side=tk.LEFT, padx=5)
31         self.entry_y = tk.Entry(entradaFrame_y, font=("Comic Sans MS", 12))
32         self.entry_y.pack(side=tk.LEFT, padx=5)
33
34         # Botones en una sola fila
35         botonesFrame = tk.Frame(contenedor, bg='#FFD700')
36         botonesFrame.pack(pady=10)
37
38         # Crear los botones con el mismo tamaño y más grandes
39         self.agregar_button = tk.Button(botonesFrame, text="Agregar Punto", command=self.agregarPuntos,
40                                         bg='#FF69B4', font=("Comic Sans MS", 12), width=20)
41         self.agregar_button.pack(side=tk.LEFT, padx=5, expand=True)
42
43         self.dibujar_button = tk.Button(botonesFrame, text="Graficar", command=self.dibujaPuntos,
44                                         bg='#87CEEB', font=("Comic Sans MS", 12), width=20)
45         self.dibujar_button.pack(side=tk.LEFT, padx=5, expand=True)
46
47         self.eliminar_button = tk.Button(botonesFrame, text="Eliminar Datos", command=self.eliminarDatos,
48                                         bg='#FF6347', font=("Comic Sans MS", 12), width=20)
49         self.eliminar_button.pack(side=tk.LEFT, padx=5, expand=True)
50
51         # Crea el gráfico
52         self.figure, self.ejes = plt.subplots(figsize=(5, 5))
53         self.ejes.set_facecolor('#FFFACD')
54         self.canvas = FigureCanvasTkAgg(self.figure, master=contenedor)
55         self.canvas.get_tk_widget().pack(pady=10)
56
57         # Configurar ejes
58         self.ejes.set_xlim(0, 10)
59         self.ejes.set_ylim(0, 10)
60         self.ejes.xaxis.set_major_locator(MultipleLocator(1))
61         self.ejes.yaxis.set_major_locator(MultipleLocator(1))
62         self.ejes.grid(True, linestyle='--', linewidth=0.5)
63
64         # Etiquetas para mostrar mensajes de error y éxito
65         self.errorLabel = tk.Label(contenedor, text="", bg='#FFD700', fg='red', font=("Comic Sans MS", 12))
66         self.errorLabel.pack(pady=5)
67
68         self.successLabel = tk.Label(contenedor, text="", bg='#FFD700', fg='green', font=("Comic Sans MS", 12))
69         self.successLabel.pack(pady=5)
70
71     def agregarPuntos(self):
72         x_texto = self.entry_x.get()
73         y_texto = self.entry_y.get()
74
75         try:
76             x = float(x_texto)
77             y = float(y_texto)
78
79             # Comprobar si los valores se encuentran dentro del rango permitido (de 0 a 10).
```

```

81         if not (0 <= x <= 10 and 0 <= y <= 10):
82             raise ValueError("Los valores deben estar entre 0 y 10.")
83
84         self.puntos.append((x, y))
85         self.errorLabel.config(text="") # Borrar el mensaje de error
86         self.successLabel.config(text=f"Punto ({x}, {y}) agregado.") # Muestra mensaje de éxito
87         self.entry_x.delete(0, tk.END)
88         self.entry_y.delete(0, tk.END)
89
90     except ValueError as e:
91         self.errorLabel.config(text=f"Error: Ingrese solo valores numéricos válidos.") # Muestra mensaje de error
92         self.successLabel.config(text="") # Borrar el mensaje de éxito.
93
94     def dibujaPuntos(self):
95         if not self.puntos:
96             return
97         self.ejes.clear()
98         self.ejes.set_facecolor('#FFFACD')
99         self.ejes.set_xlim(0, 10)
100        self.ejes.set_ylim(0, 10)
101        self.ejes.xaxis.set_major_locator(MultipleLocator(1))
102        self.ejes.yaxis.set_major_locator(MultipleLocator(1))
103        x_vals, y_vals = zip(*self.puntos)
104        self.ejes.scatter(x_vals, y_vals, color='red', label='Puntos ingresados')
105        self.ejes.legend(loc="upper center", bbox_to_anchor=(0.5, -0.15), ncol=1)
106        self.ejes.axhline(0, color='black', linewidth=0.5)
107        self.ejes.axvline(0, color='black', linewidth=0.5)
108        self.ejes.grid(True, linestyle='--', linewidth=0.5)
109        self.ejes.set_xlabel("Eje X")
110        self.ejes.set_ylabel("Eje Y")
111        self.ejes.set_title("Gráfico de Puntos")
112        self.canvas.draw()
113
114     def eliminarDatos(self):
115         """Eliminar los puntos y actualizar el gráfico"""
116         self.puntos.clear() # Eliminar los puntos de la lista
117         self.ejes.clear() # Limpiar el gráfico
118         self.ejes.set_facecolor('#FFFACD')
119         self.ejes.set_xlim(0, 10)
120         self.ejes.set_ylim(0, 10)
121         self.ejes.xaxis.set_major_locator(MultipleLocator(1))
122         self.ejes.yaxis.set_major_locator(MultipleLocator(1))
123         self.ejes.grid(True, linestyle='--', linewidth=0.5)
124         self.ejes.set_xlabel("Eje X")
125         self.ejes.set_ylabel("Eje Y")
126         self.ejes.set_title("Gráfico de Puntos")
127         self.canvas.draw() # Redibujar el gráfico
128         self.errorLabel.config(text="") # Borrar el mensaje de error
129         self.successLabel.config(text="Datos eliminados.") # Mensaje de éxito
130
131 class Aplicacion:
132     def __init__(self, ventana):
133         self.ventana = ventana
134         self.ventana.title("Aplicación Educativa")
135         self.ventana.geometry("650x850")
136         self.ventana.configure(bg="#FFD700")
137
138         # Menú
139         self.menu = Menu(ventana)
140         ventana.config(menu=self.menu)
141
142         menuPrincipal = Menu(self.menu, tearoff=0)
143         self.menu.add_cascade(label="Menú", menu=menuPrincipal)
144         menuPrincipal.add_command(label="Presentación", command=self.mostrarPresentacion)
145         menuPrincipal.add_command(label="Plano Cartesiano", command=self.mostrarPlanoCartesiano)
146         menuPrincipal.add_command(label="Ayuda", command=self.mostrarAyuda)
147         menuPrincipal.add_command(label="Acerca de", command=self.mostrarAcerca)
148
149         # Frame para mostrar los contenidos dinámicos
150         self.frameContenido = tk.Frame(ventana, bg="#FFD700")
151         self.frameContenido.pack(fill=tk.BOTH, expand=True)
152
153     def limpiarFrame(self):
154         for widget in self.frameContenido.winfo_children():
155             widget.destroy()
156
157     def mostrarPresentacion(self):
158         self.limpiarFrame()
159         tk.Label(self.frameContenido, text=" ", bg="#FFD700", font=("Comic Sans MS", 16, "bold")).pack(pady=20)
160

```

```

161 tk.Label(self.frameContenido, text="Bienvenido a la Aplicación Educativa", bg="#FFD700",
162         font=("Comic Sans MS", 16, "bold")).pack(pady=20)
163 tk.Label(self.frameContenido, text="PLANO CARTESIANO", bg="#FFD700",
164         font=("Comic Sans MS", 16, "bold")).pack(pady=20)
165 # Cargar la imagen
166 try:
167     # Leer la imagen mediante PIL.
168     image = Image.open("Puntos.png")
169     image = image.resize((400, 400))
170     imageTk = ImageTk.PhotoImage(image)
171     imageLabel = tk.Label(self.frameContenido, image=imageTk, bg="#FFD700")
172     imageLabel.image = imageTk # Mantiene una referencia a la imagen
173     imageLabel.pack(pady=20) # Mostrar imagen
174
175 except Exception as e:
176     print(f"Error al cargar la imagen: {e}")
177
178 tk.Label(self.frameContenido, text="CartesKids", bg="#FFD700",
179         font=("Comic Sans MS", 16, "bold")).pack(pady=20)
180
181
182 def mostrarPlanoCartesiano(self):
183     self.limpiarFrame()
184     GraficoApp(self.frameContenido)
185
186 def mostrarAyuda(self):
187     self.limpiarFrame()
188
189     tk.Label(self.frameContenido, text="\n\n\nINSTRUCCIONES DE USO", bg="#FFD700",
190             font=("Comic Sans MS", 14)).pack(pady=20)
191
192     tk.Label(
193         self.frameContenido,
194         text="1. Ingrese los valores X y Y en los campos correspondientes.\n"
195             "  Estos valores deben estar dentro del rango de 0 a 10.\n\n"
196             "2. Haz clic en el botón "Agregar Punto".\n"
197             "  Esto registrará el par ordenado (X, Y).\n\n"
198             "3. Una vez ingresados todos los puntos que desees,\n"
199             "  pulsa el botón "Graficar" para visualizar los puntos en el plano cartesiano.\n\n"
200             "4. Si necesitas comenzar de nuevo,\n"
201             "  presiona el botón "Eliminar Datos" para limpiar todos los puntos y reiniciar el gráfico.",
202         bg="#FFD700",
203         font=("Comic Sans MS", 10),
204         justify="center",
205         anchor="w"
206     ).pack(pady=20, fill='x', padx=20)
207
208
209 def mostrarAcerca(self):
210     self.limpiarFrame()
211     tk.Label(self.frameContenido, text="\n\n\n\nAplicación creada para educación primaria.\n"
212             f"CartesKids v1.0", bg="#FFD700", font=("Comic Sans MS", 14, "bold")).pack(pady=20)
213     # Listado de contenidos para las etiquetas.
214     textos = [
215         "\nDesarrollado por: Leonel Coyla Idme",
216         "Alfredo Mamani Canqui",
217         "Elqui Yeye Pari Condori",
218         "Juan Reynaldo Paredes Quispe",
219         "José Pánfilo Tito Lipa",
220         "\nLanzamiento: 12 abril 2025"
221         "\nContacto: lcoyla@unap.edu.pe"
222     ]
223
224     # Construir las etiquetas utilizando un bucle.
225     for texto in textos:
226         tk.Label(self.frameContenido, text=texto, bg="#FFD700", font=("Comic Sans MS", 14)).pack(pady=4)
227
228 if __name__ == "__main__":
229     ventana = tk.Tk()
230     app = Aplicacion(ventana)
231     ventana.mainloop()
232

```

