

CLASES ABSTRACTAS E INTERFACES
SIMULADOR DEL CLIMA

HANNA KATHERINE ABRIL GÓNGORA
KAROL ASLEY ORJUELA MAPE

UNIVERSIDAD MANUELA BELTRÁN

PROGRAMACIÓN ORIENTADA A OBJETOS

DOCENTE

DIANA MARCELA TOQUICA RODRÍGUEZ

BOGOTÁ DC LUNES 15 DE ABRIL

1. PREGUNTAS ORIENTADORAS

1.1. ¿Cómo puede la interfaz gráfica mejorar la usabilidad de una aplicación?

- **Facilidad de uso:** Una interfaz gráfica bien diseñada donde facilita a los usuarios comprender cómo interactuar con la aplicación. Esto se logra mediante la disposición clara de elementos visuales, como botones, menús y controles, que hacen que la navegación sea intuitiva.
- **Feedback inmediato:** Una GUI proporciona feedback visual instantáneo cuando los usuarios realizan acciones. Esto incluye cambios en el estado de los elementos de la interfaz, animaciones que indican el progreso de una tarea y mensajes emergentes que confirman la finalización de una acción.
- **Consistencia:** Mantener la coherencia en el diseño y la disposición de los elementos de la interfaz mejora la usabilidad. Los usuarios pueden predecir dónde encontrar ciertas funciones y acciones, lo que reduce la confusión y la curva de aprendizaje.
- **Simplicidad:** Una interfaz gráfica simple y limpia reduce la carga cognitiva para los usuarios. Evitar el desorden visual y presentar solo la información relevante y las opciones necesarias mejora la experiencia del usuario.
- **Accesibilidad:** Una GUI accesible es fundamental para garantizar que todos los usuarios, incluidos aquellos con discapacidades, puedan interactuar con la aplicación de manera efectiva. Esto implica proporcionar opciones como tamaño de texto ajustable, soporte para lectores de pantalla y opciones de contraste para mejorar la legibilidad.
- **Personalización:** Permitir que los usuarios personalicen la interfaz según sus preferencias individuales mejora la usabilidad. Esto puede incluir la capacidad de cambiar temas, reorganizar la disposición de los elementos o ajustar la configuración de accesibilidad.
- **Flujo de trabajo eficiente:** Una GUI bien diseñada facilita la realización de tareas comunes de manera eficiente. Esto se logra mediante la optimización del flujo de trabajo y la disposición de los elementos de la interfaz de acuerdo con la frecuencia y la importancia de las acciones.

1.2. ¿Qué elementos de GUI son más efectivos para interactuar con los tipos de datos que se manejan en un simulador de clima?

- **Gráficos e imágenes:** Representar datos meteorológicos mediante gráficos de líneas, barras, mapas de calor o imágenes satelitales puede ayudar a los usuarios a visualizar patrones climáticos, tendencias y cambios en el tiempo.
- **Controles deslizantes y selectores:** Permitir a los usuarios ajustar parámetros como la fecha, la hora, la ubicación geográfica o las variables meteorológicas (temperatura, humedad, velocidad del viento, etc.) mediante controles deslizantes o menús desplegables puede facilitar la exploración y análisis de los datos.
- **Widgets de pronóstico:** Mostrar pronósticos meteorológicos futuros utilizando widgets que muestren información como la temperatura, la probabilidad de precipitación, la velocidad del viento y la humedad para diferentes momentos del día y ubicaciones geográficas.
- **Mapas interactivos:** Permitir a los usuarios interactuar con mapas interactivos donde puedan hacer zoom, desplazarse y hacer clic en ubicaciones específicas para obtener detalles meteorológicos precisos y localizados.
- **Alertas y notificaciones:** Incorporar elementos que muestren alertas meteorológicas y notificaciones de condiciones climáticas adversas o cambios significativos, como tormentas, tornados o alertas de calor.
- **Widgets de resumen:** Proporcionar widgets de resumen que muestren información meteorológica clave de manera concisa, como la temperatura actual, la previsión del tiempo para los próximos días y cualquier alerta meteorológica activa.
- **Historial y registros:** Permitir a los usuarios acceder a registros históricos de datos meteorológicos y visualizar tendencias a lo largo del tiempo utilizando gráficos o tablas interactivas.
- **Ayudas visuales y leyendas:** Incluir ayudas visuales como leyendas de colores para indicar rangos de valores (por ejemplo, colores más oscuros para temperaturas más bajas y colores más claros para temperaturas más altas) para facilitar la interpretación de los datos.

1.3. ¿De qué manera se pueden estructurar los elementos de la GUI para facilitar una experiencia de usuario intuitiva?

- **Organización lógica:** Agrupa los elementos relacionados de manera coherente y lógica. Por ejemplo, coloca los controles de ajuste de configuración cerca de las áreas relevantes de visualización de datos.
- **Jerarquía visual:** Utiliza tamaño, color y contraste para destacar elementos importantes y crear una jerarquía visual clara. Los elementos más importantes deben ser prominentes y fáciles de identificar.
- **Diseño limpio y espaciado adecuado:** Evita el desorden visual y proporciona suficiente espacio entre los elementos para evitar la confusión. Un diseño limpio ayuda a que la interfaz sea más fácil de entender y usar.
- **Flujo de trabajo intuitivo:** Diseña el flujo de trabajo de la aplicación de manera que sea fácil de seguir para el usuario. Los pasos deben ser lógicos y seguir un orden natural de uso.
- **Consistencia en el diseño:** Mantén una consistencia visual en toda la interfaz, incluidos los colores, las tipografías, los estilos de botones y los iconos. Esto ayuda a que los usuarios se familiaricen con la aplicación y sepan qué esperar en cada pantalla.
- **Navegación clara:** Proporciona una navegación clara y fácil de entender. Utiliza menús desplegables, barras de navegación o botones de retroceso para permitir que los usuarios se muevan sin problemas por la aplicación.
- **Ayuda contextual:** Ofrece ayuda contextual cuando sea necesario, como descripciones emergentes o enlaces a tutoriales. Esto ayuda a los usuarios a comprender mejor cómo utilizar la aplicación cuando lo necesitan.
- **Pruebas de usabilidad:** Realiza pruebas de usabilidad con usuarios reales para identificar áreas de confusión o dificultad en la interfaz. Utiliza los resultados de estas pruebas para realizar ajustes y mejoras en el diseño.

1.4. ¿Cómo se gestionan los eventos en Tkinter para responder a las acciones del usuario?

- **Crear la interfaz gráfica:** Primero, debes crear la interfaz gráfica utilizando los widgets

proporcionados por Tkinter, como Button, Entry, Label, etc.

- **Definir funciones de manejo de eventos:** Luego, define las funciones que se ejecutarán cuando ocurran eventos específicos. Estas funciones serán responsables de realizar acciones específicas cuando se desencadenen los eventos.
- **Asociar funciones a eventos:** Después de definir las funciones de manejo de eventos, necesitas asociar estas funciones con los eventos correspondientes en los widgets. Esto se hace utilizando métodos como bind() para widgets específicos o command para botones.
- **Ejecutar el bucle principal:** Finalmente, inicia el bucle principal de la aplicación llamando al método mainloop() del objeto Tk. Este bucle se encargará de detectar y manejar los eventos de la interfaz gráfica.

2. EXPLICACIÓN DEL CÓDIGO EN PYTHON

```
5 class VentanaPrincipal(tk.Tk):
6     def __init__(self):
7         super().__init__()
8
9         self.attributes('-fullscreen', True)          # Configurar la
10        self.bind("<Escape>", self.salir_pantalla_completa)
11        self.crear_contenido()                        # Llamar al método para crear
12        self.title("SkyScape")
13        self.configure(bg='white')
14        imagen_icono = tk.PhotoImage(file="icono_ventana.png")
15        self.iconphoto(False, imagen_icono)
16
```

- self.attributes('-fullscreen', True): Configura la ventana para abrirse en pantalla completa.
- self.bind("<Escape>", self.salir_pantalla_completa): Vincula la tecla "Esc" a la función self.salir_pantalla_completa.
- self.crear_contenido(): Llama al método para crear el contenido de la ventana.
- self.title("SkyScape"): Establece el título de la ventana como "SkyScape".
- self.configure(bg='white'): Configura el color de fondo de la ventana como blanco.
- imagen_icono = tk.PhotoImage(file="icono_ventana.png"): Carga una imagen para usar

como ícono de la ventana.

- `self.iconphoto(False, imagen_ícono)`: Establece la imagen cargada como ícono de la ventana.

```
17     def salir_pantalla_completa(self, event=None):
18         self.attributes('-fullscreen', False)           # Cambiar el e
19
```

- `def salir_pantalla_completa(self, event=None)::` Define un método para cambiar el estado de pantalla completa al presionar "Esc".

```
21     def crear_contenido(self):
22         # Label del título del programa
23         label_bienvenida = tk.Label(self, text="Bienvenido a SkyScap
24         label_bienvenida.pack(side=tk.TOP, fill=tk.X)
25
26         texto="""SkyScape es una herramienta avanzada de simulación
27         los diferentes fenómenos meteorológicos de forma interactiva
28         label_bienvenida = tk.Label(self, text=texto, font=("Arial",
29         label_bienvenida.pack(side=tk.TOP, fill=tk.X)
30
31         # Frame para ingreso de información
32         self.frame_ingreso = FrameIngreso(self)
33         self.frame_ingreso.pack(side=tk.TOP, expand=True, pady=10)
34
35         # Frame para mostrar resultado
36         self.frame_resultado = FrameResultado(self)
37         self.frame_resultado.pack(side=tk.TOP, expand=True, pady=10)
38
```

- Este método se encarga de crear y organizar los elementos dentro de la ventana, como etiquetas, campos de entrada y botones.

```
class FrameIngreso(tk.Frame):
    def __init__(self, root):
        super().__init__(root, bg="white", width=730, height=450, bo
```

- `class FrameIngreso(tk.Frame)::` Define una clase llamada `FrameIngreso` que hereda de `tk.Frame`, lo que significa que esta clase representa un marco en la interfaz gráfica.
- `def __init__(self, root)::` Define el método de inicialización de la clase `FrameIngreso`. Toma `root` como argumento, que se refiere al elemento padre al que pertenece este marco.

- `super().__init__(root, bg="white", width=730, height=450, borderwidth=1, relief=tk.FLAT, highlightbackground="grey", highlightthickness=1)`: Llama al método de inicialización de la clase padre (`tk.Frame`) para configurar el marco con ciertas propiedades, como el color de fondo, el ancho, el alto, el ancho del borde y el estilo del relieve.

```

52     label_pais = tk.Label(self, text="País:", font=("Arial", 10, "bold"),bg='white')
53     label_pais.pack(expand=True, padx=10, pady=5)
54     entrada_pais = tk.Entry(self, borderwidth=1, relief=tk.FLAT, highlightbackground="grey")
55     entrada_pais.pack(expand=True, padx=10, pady=5)
56
57     label_ciudad = tk.Label(self, text="Ciudad:", font=("Arial", 10, "bold"),bg='white')
58     label_ciudad.pack(expand=True, padx=10, pady=5)
59     entrada_ciudad = tk.Entry(self, borderwidth=1, relief=tk.FLAT, highlightbackground="grey")
60     entrada_ciudad.pack(expand=True, padx=10, pady=5)
61
62     texto="""Por favor ingrese la siguiente información para poder realizar la simulación""
63     label_bienvenida = tk.Label(self, text=texto, font=("Arial", 10), pady=10,bg="white")
64     label_bienvenida.pack(side=tk.TOP, fill=tk.X)
65
66     label_cantidad_dias = tk.Label(self, text="Ingrese la cantidad de dias:", font=("Arial", 10, "bold"),bg='white')
67     label_cantidad_dias.pack(side=tk.LEFT, fill=tk.X, expand=True, padx=5, pady=10)
68     entrada_cantidad_dias = tk.Entry(self, borderwidth=1, relief=tk.FLAT, highlightbackground="grey")
69     entrada_cantidad_dias.pack(side=tk.LEFT, fill=tk.X, expand=True, padx=5, pady=10)
70     separador_dias = tk.Label(self, text="|", bg='white', font=("Arial", 20, "bold"),bg="white")
71     separador_dias.pack(side=tk.LEFT, fill=tk.X, expand=True,pady=50)
72
73     label_temperatura = tk.Label(self, text="Temperatura:", font=("Arial", 10, "bold"),bg='white')
74     label_temperatura.pack(side=tk.LEFT, fill=tk.X, expand=True, padx=5, pady=10)
75     entrada_temperatura = tk.Entry(self, borderwidth=1, relief=tk.FLAT, highlightbackground="grey")
76     entrada_temperatura.pack(side=tk.LEFT, fill=tk.X, expand=True, padx=5, pady=10)
77     unidades_temperatura = tk.Label(self, text="°C", bg='white', font=("Arial", 10, "bold"),bg="white")
78     unidades_temperatura.pack(side=tk.LEFT, fill=tk.X, expand=True,pady=10)
79
80     label_humedad = tk.Label(self, text="Humedad:", font=("Arial", 10, "bold"),bg='white')
81     label_humedad.pack(side=tk.LEFT, fill=tk.X, expand=True, padx=5, pady=10)
82     entrada_humedad = tk.Entry(self, borderwidth=1, relief=tk.FLAT, highlightbackground="grey")
83     entrada_humedad.pack(side=tk.LEFT, fill=tk.X, expand=True, padx=5, pady=10)
84     unidades_humedad = tk.Label(self, text="%", bg='white', font=("Arial", 10, "bold"),bg="white")
85     unidades_humedad.pack(side=tk.LEFT, fill=tk.X, expand=True,pady=10)
86
87     label_presion_at = tk.Label(self, text="Presión Atmosferica:",font=("Arial", 10, "bold"),bg='white')
88     label_presion_at.pack(side=tk.LEFT, fill=tk.X, expand=True, padx=5, pady=10)
89     entrada_presion_at = tk.Entry(self, borderwidth=1, relief=tk.FLAT, highlightbackground="grey")
90     entrada_presion_at.pack(side=tk.LEFT, fill=tk.X, expand=True, padx=5, pady=10)
91     unidades_presion_at = tk.Label(self, text="hPa", bg='white',font=("Arial", 10, "bold"),bg="white")
92     unidades_presion_at.pack(side=tk.LEFT, fill=tk.X, expand=True,pady=10)
93

```

- Se crean etiquetas (tk.Label) y campos de entrada (tk.Entry) para solicitar información al usuario, como el país, la ciudad, la temperatura, etc. Se especifican las propiedades de estos elementos, como el texto, la fuente, el estilo, el color de fondo y el grosor del borde.
- Se crea una etiqueta de bienvenida que muestra un mensaje al usuario para indicar qué tipo de información debe ingresar.

```

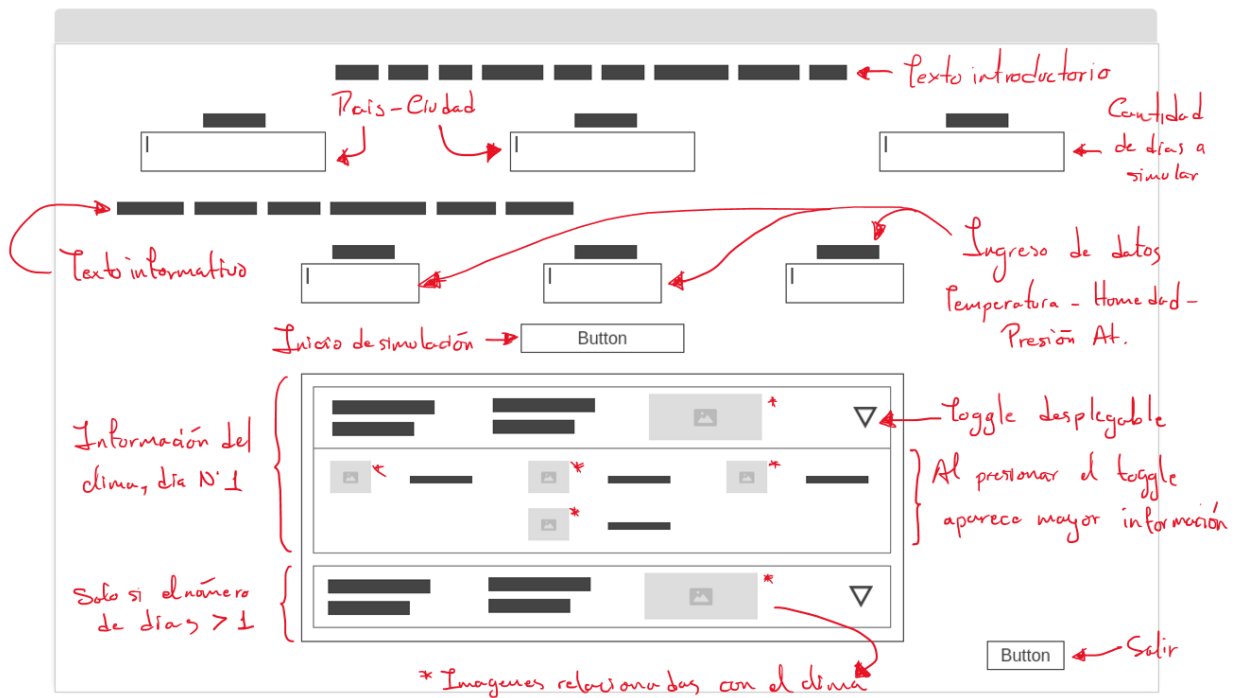
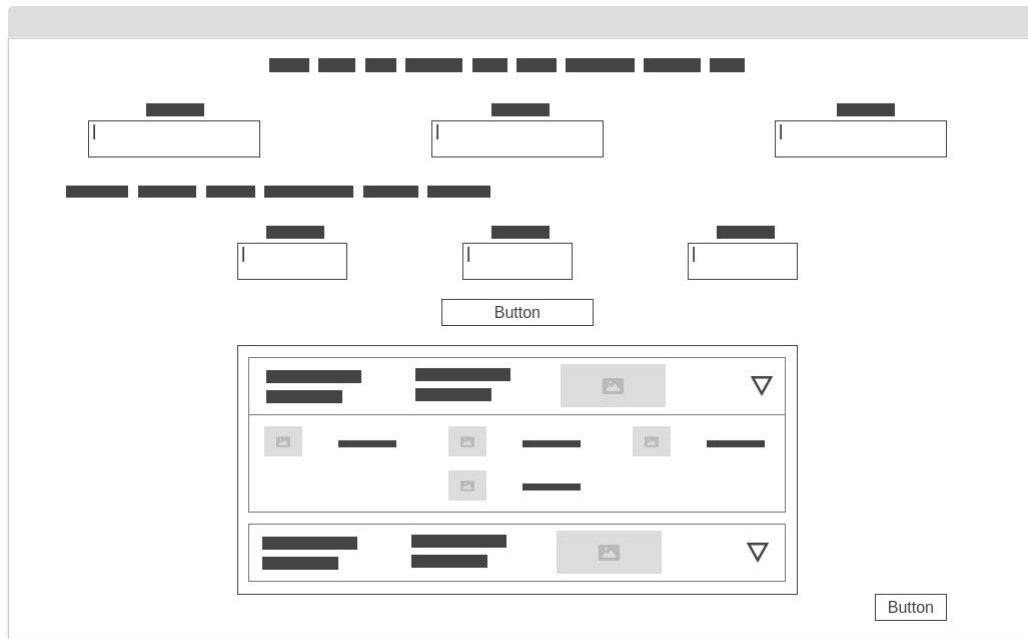
94         boton_calcular = tk.Button(self, text="Calcular", command=self
95         boton_calcular.place(relx=0.538, rely=0.95, anchor='se')
96
97     def calcular(self):
98         info = self.entrada_info.get()
99         # Llamamos al método mostrar_resultado de la ventana princip
100         self.master.mostrar_resultado(info)
101

```

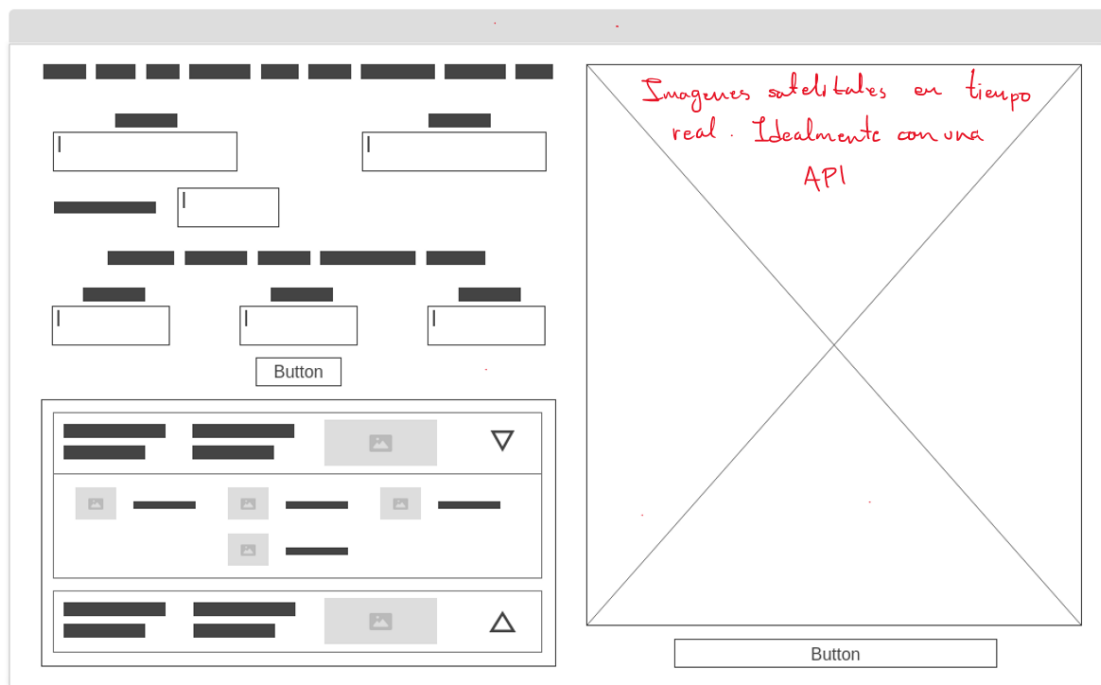
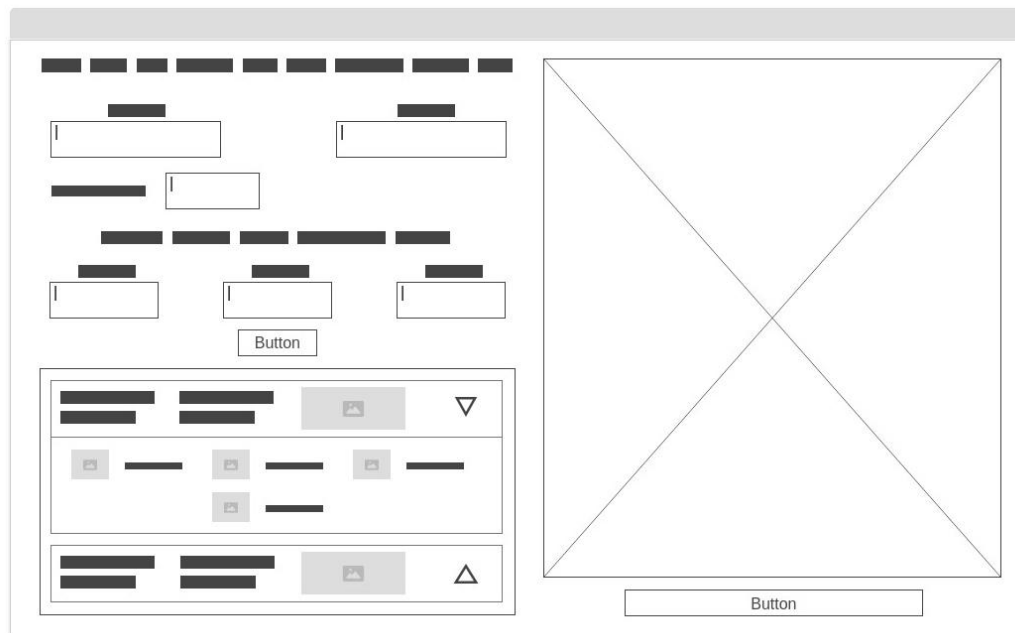
- Se crea un botón (tk.Button) con el texto "Calcular" que llamará al método self.calcular cuando se haga clic en él. Se especifican propiedades como el texto, el comando asociado, el ancho y el estilo.
- Define un método calcular que se ejecuta cuando se presiona el botón "Calcular". Obtiene la información ingresada por el usuario y llama al método mostrar_resultado del elemento padre (self.master), que en este caso es la ventana principal, para mostrar el resultado.

3. DIAGRAMA WIREFRAME (MOCKUP)

3.1. WIREFRAME 1



3.2. WIREFRAME 2





4. EXPLICACION Y USO


En este caso se va a realizar un simulador del clima el cual sea capaz de predecir el clima según ciertos parámetros.

En este caso el usuario tendrá que ingresar el país y la ciudad donde quiere hacer la predicción


del clima, se debe tener en cuenta que se planea usar un API para que se pueda verificar la veracidad tanto del país como de la ciudad, esto se hace con el motivo del cálculo de la predicción del clima ya que dependiendo si la ciudad se clasifica como fría o caliente, dichos parámetros pueden cambiar.

Además de eso el usuario tendrá que ingresar la temperatura, la humedad, y la presión atmosférica, con estos parámetros se colocaran ciertas condiciones dependiendo del tipo del clima, estos se pueden ver en la siguiente tabla.


Tipo de clima	Temperatura	Humedad	Presión atmosférica
Soleado 	Climas templados: 15°C - 25°C	Climas templados o cálidos: 30% - 60%	Condición climática estable: 1013 hPa
	Climas cálidos: 30°C	Climas desérticos: <30%	
Estos son rangos generales y pueden variar según la ubicación geográfica, la estación del año y otros factores locales. Un día soleado típicamente implica cielos despejados o con pocas nubes, con una sensación agradable y luminosa debido a la presencia del sol.			
Parcialmente nublado 	Climas templados: 10°C - 25°C	Climas templados o cálidos: 40% - 70%	Condición climática estable: 1000 hPa a 1013 hPa
	Climas cálidos: 20°C hasta 35°C o más	Zonas húmedas o costeras: >70%	
Un día parcialmente nublado implica que habrá una mezcla de sol y nubes en el cielo. Esto puede resultar en cambios sutiles en la temperatura, ya que la cobertura de nubes puede bloquear parte de la radiación solar, enfriando ligeramente el ambiente en comparación con un día completamente despejado. La presión atmosférica y la humedad suelen estar en niveles moderados durante este tipo de clima.			

Lluvioso 	Climas templados: 5°C - 20°C	Climas templados o cálidos: 70%.	Depende de la intensidad de la lluvia, normalmente: 990 hPa a 1010 hPa
	Climas cálidos: 15°C - 30°C	Zonas húmedas o costeras: 100%.	

Es importante tener en cuenta que estos son rangos generales y pueden variar según la ubicación geográfica y las condiciones específicas del sistema meteorológico que esté generando la lluvia. La temperatura y la presión atmosférica suelen tener una relación inversa durante la lluvia, ya que la actividad atmosférica asociada a la precipitación puede causar una disminución temporal en la presión.


Tormentas 	Climas templados: 10°C - 25°C	Condición climática estable: 80% - 100%	Condición climática estable: <1000 hPa
	Climas cálidos: 20°C -35°C		

Es importante recordar que las tormentas pueden ser muy variables en términos de su intensidad y duración. Estos rangos son generales y pueden variar según la ubicación geográfica y la naturaleza específica de la tormenta (como tormentas eléctricas, tormentas de viento, etc.). Durante una tormenta eléctrica, también se pueden observar cambios en la carga eléctrica en la atmósfera, lo que puede afectar a los campos electromagnéticos locales y a la distribución de la presión atmosférica.


Nieve 	Climas fríos: <0°C	Condición climática estable: 70% - 80%	Condición climática estable: 990 hPa a 1013 hPa
	Climas con inviernos fríos: -5°C hasta -15°C		

Es importante tener en cuenta que estos son rangos generales y pueden variar según la región geográfica y las condiciones locales. Por ejemplo, en climas más fríos y polares, las temperaturas durante una nevada pueden ser mucho más bajas que en áreas más templadas con inviernos suaves. La presión

atmosférica y la humedad durante una nevada suelen reflejar las condiciones típicas del clima frío y húmedo asociado con la precipitación en forma de nieve.



Viento fuerte 	Climas templados: 10°C - 25°C	En general, la humedad relativa puede ser moderada a alta, especialmente si el viento proviene de áreas marítimas o de mayor humedad.	Condición climática estable: 990 hPa a 1013 hPa
		En áreas más secas, la humedad relativa durante el viento fuerte puede ser más baja.	

Es importante tener en cuenta que el viento fuerte puede estar asociado con diferentes condiciones meteorológicas, como frentes fríos, sistemas de baja presión o vientos locales intensificados por la topografía. La sensación térmica durante un viento fuerte puede ser más fría debido al enfriamiento del viento en la piel, incluso si la temperatura real es más alta.

Niebla 	Climas templados: 5°C - 15°C	Condición climática estable: 100%	Condición climática estable: 990 hPa a 1013 hPa
	Climas fríos: <5°C		

Es importante tener en cuenta que la niebla puede tener diferentes tipos y características según la región y las condiciones locales. La visibilidad reducida y la sensación de humedad en el aire son características comunes de la niebla, y estas condiciones pueden persistir hasta que la niebla se disipe debido a cambios en la temperatura, el viento u otras condiciones atmosféricas.

Heladas	Climas templados: -5°C - 5°C	Condición climática estable: 20%	Condición climática estable: 990 hPa a 1013 hPa
----------------	--	--	---

	Climas fríos: ≤-15°C		
<p>Es importante tener en cuenta que las heladas pueden ocurrir tanto en climas secos como en climas húmedos, y la intensidad de la helada puede afectar la temperatura y la humedad relativa en diferentes formas. Durante una helada severa, las temperaturas pueden caer drásticamente y la humedad relativa puede ser muy baja, lo que contribuye a condiciones frías y secas.</p>			
	Temperaturas extremas de calor		
	Climas templados: 30°C - 35°C	Climas templados: 10% - 30%	Condición climática estable: 990 hPa a 1013 hPa
	Climas cálidos: 40°C - 50°C	Climas cálidos o áridos: <10%	
	Temperaturas extremas de frío		
	Climas templados o fríos: -20°C hasta -40°C	Climas templados o fríos: <30%	Condición climática estable: 990 hPa a 1013 hPa
<p>Es importante tener en cuenta que las temperaturas extremas pueden tener impactos significativos en la salud humana, la agricultura, la infraestructura y el medio ambiente. Durante estos eventos, es crucial tomar precauciones adecuadas y seguir las recomendaciones de las autoridades meteorológicas para protegerse del calor o del frío extremo.</p>			

A partir de la información anterior se definirán los condicionales necesarios para según los datos ingresados por el usuario (país, ciudad(opcional), temperatura y presión atmosférica) se pueda dar una predicción del clima medianamente acertada, no en tiempo real, pero al menos similar a lo que podría ser.

5. DEMOSTRACION DE INTERFAZ GRAFICA



The screenshot shows a web application window titled "SkyScape". The main heading is "Bienvenido a SkyScape: Simulación del Clima". Below this, a descriptive text states: "SkyScape es una herramienta avanzada de simulación del clima que te permite explorar y comprender los diferentes fenómenos meteorológicos de forma interactiva." The interface contains a form with the following elements:

- Input fields for "Pais:" and "Ciudad:".
- A prompt: "Por favor ingrese la siguiente información para poder realizar la simulación".
- Input fields for "Ingrese la cantidad de días:", "Temperatura:" (with a unit of °C), "Humedad:" (with a unit of %), and "Presión Atmosferica:" (with a unit of hPa).
- A "Calcular" button.
- A large empty rectangular box for the simulation results.
- A "Salir" button in the bottom right corner.

Hay que tener en cuenta que este es un prototipo de la interfaz gráfica, por motivos de complejidad en el código, y tiempo reducido se muestra de esta manera, pero se planea realizar una mejor implementación de la interfaz grafica a futuro.

6. DECISIÓN DE DISEÑO O CONSIDERACION IMPORTANTE

Para el diseño de interfaces gráficas, no basamos en simuladores, o páginas del clima ya existentes, recopilamos la información necesaria y lo más importante que se tenía que mostrar en la interfaz. Además, tuvimos en cuenta como se iba a mostrar la información entre otras, respecto a colores, tipos de letra, imágenes entre otros aún está bajo decisión para el proyecto final.

7. PILARES DE POO IMPLEMENTADOS

Nuestra estrategia incluye la implementación completa de los pilares de la programación orientada a objetos al finalizar el proyecto. En esta etapa inicial, nos enfocamos en la integración de conceptos de herencia y polimorfismo. Esta selección inicial nos permite establecer una base sólida para la expansión gradual de la funcionalidad del simulador del clima y la incorporación de los restantes pilares a medida que avanza el desarrollo.

8. CONCLUSIONES

El desarrollo de un simulador del clima con GUI y Tkinter no solo nos impulsa el crecimiento técnico, sino que también fomenta el desarrollo de habilidades creativas y de resolución de problemas. Es una oportunidad emocionante para aplicar el conocimiento adquirido en el aula en un proyecto práctico y significativo.