

Diseño Detallado de Arquitectura de Software

Juan Carlos Gómez Hernández¹

¹ Universidad Linda Vista, Ex-Finca Santa Cruz No. 1, 29750, México correo
1@ulv.edu.mx

² Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com

³ ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

1. Diagrama de clases.

1.1. Diagrama de clase interfaz.



Figura 1. Diagrama de clase 1.

1.2. Diagrama de clase simulador.

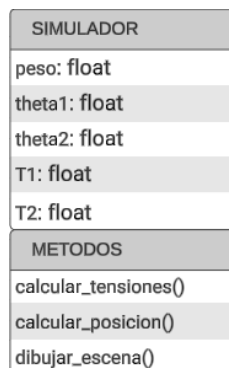


Figura 2. Diagrama de clase 2.

1.3. Diagrama de clase grafico.

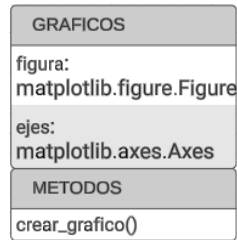


Figura 3. Diagrama de clase 3.

1.4. Detalle de Pseudocódigo

Calcular Tensiones

```

1 Calcular Tensiones
2 Funci n calcular_tensiones(peso, theta1, theta2):
3     Convertir theta1 y theta2 a radianes
4     Denominador = sin(theta1_rad + theta2_rad)
5     Si denominador es 0:
6         Retornar 0, 0
7     T1 = peso * sin(theta2_rad) / denominador
8     T2 = peso * sin(theta1_rad) / denominador
9     Retornar T1, T2

```

Dibujar Escena

```

1 Funci n dibujar_escena(pantalla, peso, theta1, theta2):
2     Limpiar pantalla con color blanco
3     Calcular tensiones usando calcular_tensiones(peso, theta1
4         , theta2)
5     Calcular posici n del cuerpo usando calcular_posicion()
6     Dibujar im genes (cuerpos, poleas, cuerdas) en la
7         pantalla
8     Mostrar tensiones y ngulos en la pantalla

```

Conversor

```

1 Funci n conversor(Kg):
2     retornar Kg * 9.81

```

Crear Gráfico

```

1 Funci n crear_grafico(peso, theta1, theta2):
2     Crear figura y ejes en matplotlib
3     Graficar cuerpo y vectores de fuerza (Peso, T1, T2)
4     Ajustar l mites y leyenda del gr fico
5     Retornar imagen para pygame

```

2. Estructura de datos

La estructura de datos utilizada en el Simulador de Equilibrio se organiza en varias funciones y variables que permiten manejar tanto la lógica del simulador como su representación gráfica. A continuación, se describen los principales elementos:

2.1. Simulador

Atributos	
peso: float	Representa el peso del objeto en estudio.
theta1: float	Primer ángulo en grados.
theta2: float	Segundo ángulo en grados.
T1: float	Tensión calculada en la cuerda 1.
T2: float	Tensión calculada en la cuerda 2
Métodos	
calcular_tensiones()	Calcula las tensiones basadas en los ángulos y el peso.
calcular_posicion()	Determina la posición del cuerpo basado en las tensiones.

2.2. Grafico

Atributos	
figura: matplotlib.figure.Figure	Almacena la figura del gráfico generado.
ejes: matplotlib.axes.Axes	Contiene los ejes para dibujar el gráfico.
Métodos	
crear_grafico()	Genera un gráfico que muestra las fuerzas actuantes sobre el cuerpo.

2.3. Interfaz

Atributos	
pantalla: pygame.Surface	Representa la superficie donde se dibujará el simulador.
Métodos	
iniciar()	Inicializa la interfaz gráfica y comienza el bucle principal del simulador.
mostrar_mensaje()	Muestra mensajes informativos al usuario.

2.4. Otras implementaciones

Variables Globales	
WIDTH, HEIGHT	Definen las dimensiones de la ventana del simulador.
WHITE, BLACK, etc.	Colores utilizados para el fondo y otros elementos gráficos.
Variables para manejar el estado de arrastre del mouse, que permiten a los usuarios interactuar con los objetos en la pantalla.	
Parámetros Físicos	
Se definen variables como g (gravedad) y constantes relacionadas con el sistema físico que se simula, facilitando cálculos precisos durante la ejecución.	

3. Interfaces de los Componentes.

Las interfaces entre los componentes del sistema son cruciales para garantizar que cada parte funcione correctamente y se comunique eficazmente. A continuación, se describen las interfaces clave:

Interfaz Gráfica (Interfaz):

- Esta interfaz actúa como punto de contacto entre el usuario y el simulador. Maneja eventos de entrada (como clics y arrastres) y actualiza la visualización basada en las interacciones del usuario.
- La interfaz gráfica está diseñada para ser intuitiva, permitiendo a los usuarios iniciar simulaciones, ajustar parámetros y ver resultados sin complicaciones.

Módulo de Cálculo (Simulador):

- Este módulo realiza todos los cálculos necesarios para determinar las tensiones y posiciones basadas en los parámetros ingresados por el usuario.
- Se comunica con la interfaz gráfica para proporcionar resultados que se mostrarán al usuario, como tensiones calculadas y posiciones de los cuerpos.

Módulo Gráfico (Grafico):

- Este módulo es responsable de generar gráficos utilizando Matplotlib. Convierte estos gráficos a un formato que puede ser utilizado por Pygame, permitiendo su visualización dentro del simulador.
- La comunicación entre este módulo y el módulo de cálculo es esencial, ya que necesita recibir datos sobre las tensiones y posiciones para crear representaciones gráficas precisas.

Interacción entre Componentes:

- Cuando un usuario ajusta un parámetro (por ejemplo, peso o ángulo), la interfaz gráfica llama al módulo de cálculo para actualizar las tensiones. Luego, esos resultados se envían al módulo gráfico para actualizar la visualización.
- Este flujo de información debe ser eficiente para asegurar que el simulador responda rápidamente a las acciones del usuario, manteniendo una experiencia fluida.