<!--Estudio Shonos-->

# Proyecto final

{

<Por="calculadora"/>

}

# Contenidos

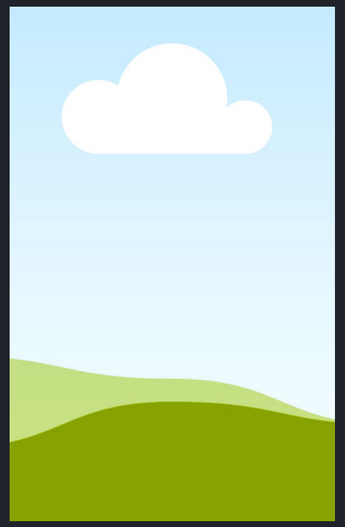
# Introducción {

El código implementa una calculadora científica usando la biblioteca Tkinter en Python. Proporciona una interfaz gráfica con botones para operaciones matemáticas básicas (suma, resta, multiplicación, división), funciones científicas (trigonometría, raíz cuadrada, potencia), y un historial para registrar cálculos. Además, maneja errores y está diseñada con un estilo moderno y minimalista.

- Estructuración.
- Revisión de errores.
- Corrección.
- Asignación del equipo.
- Formateo.
- Limpieza.
- Exportación.
- Depuración.

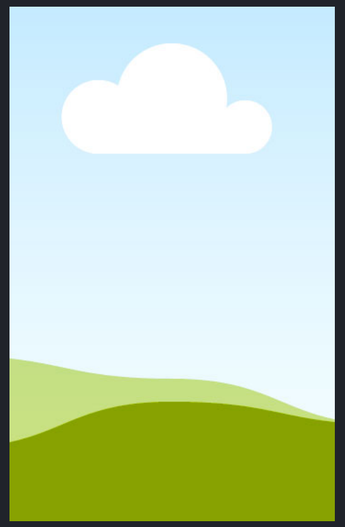
}

# Equipo {



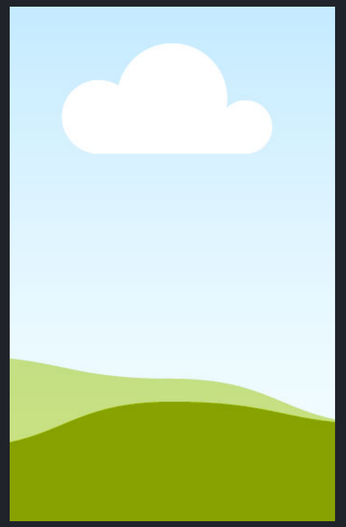
ULISES HERNANDEZ

DIRECCIÓN DE PROYECTO



DIEGO

DIRECCIÓN TÉCNICA



URI

RESPONSABLE EQUIPO

}

# Funciones {

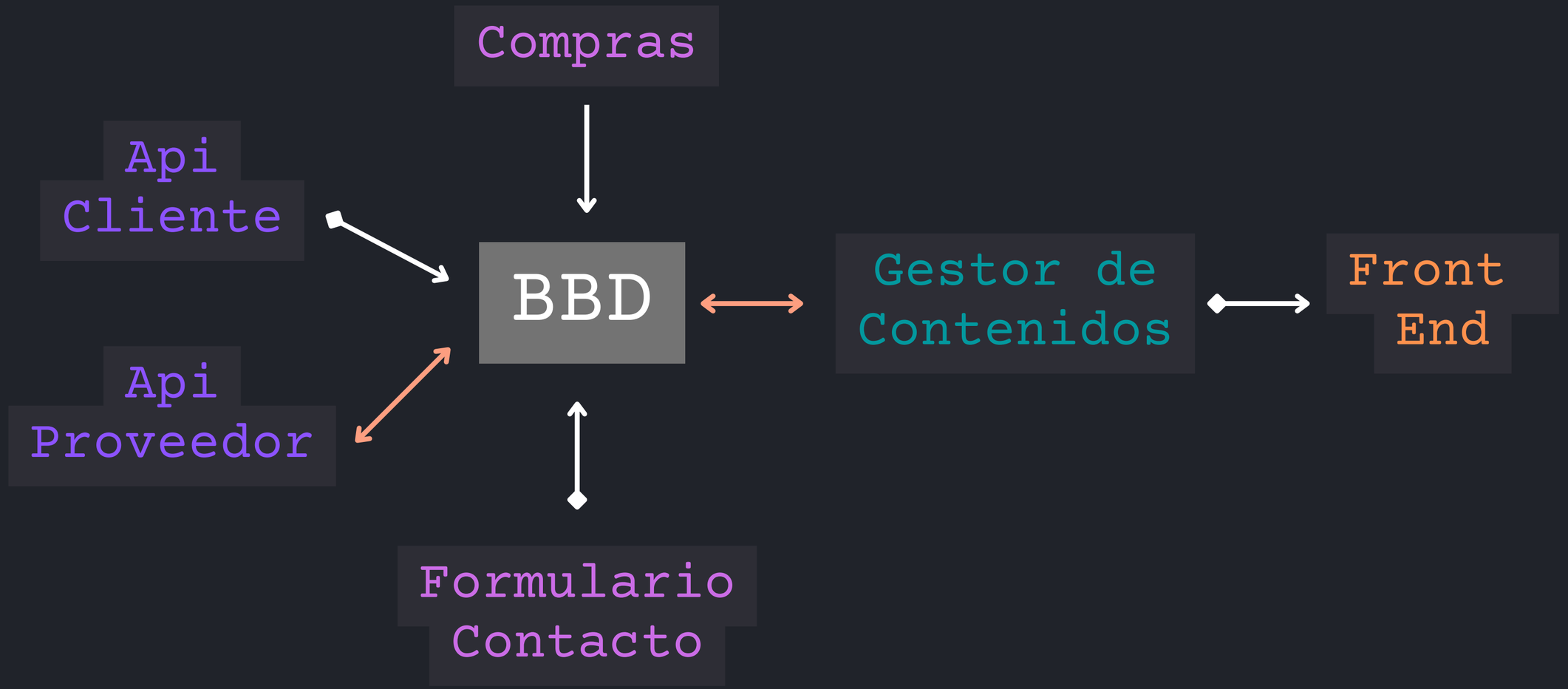
click_boton: Gestiona las acciones según el botón presionado

_operacion_trigonometrica: Realiza cálculos trigonométricos (sin, cos, tan)

crear_botones: Crea los botones de la calculadora

}

Estructura {


Compras
Api
Cliente
BBD
Gestor de
Contenidos
Front
End
Api
Proveedor
Formulario
Contacto

}

# Puntos clave {

01

Diseño en Tkinter con botones organizados por filas y columnas.

02

Suma, resta, multiplicación, división

03

Trigonometría, raíz cuadrada, potencia.

04

Registra y muestra los cálculos realizados

05

Notifica expresiones inválidas o números no permitidos

}

# Aspectos de mejora {

## Puntos de mejora.

- Validación de entrada: Restringir caracteres no válidos al escribir.
- Formato del historial: Mejorar la presentación para facilitar la lectura.
- Conversión a radianes: Opción para cambiar entre grados y radianes.
- Diseño adaptable: Soporte para pantallas de diferentes tamaños.
- Más funciones: Agregar logaritmos, factorial, y constantes como π y e.

## Como organizarse.

- Definir objetivos: Lista de funcionalidades clave (básicas y avanzadas).
- Diseño inicial: Bosquejar la interfaz y estructura del código.
- Dividir en módulos: Separar lógica, interfaz y funciones matemáticas.
- Priorizar: Implementar primero funciones básicas, luego agregar avanzadas.
- Pruebas: Verificar errores en cálculos y usabilidad.
- Iterar: Mejorar diseño, añadir funcionalidades y optimizar código.

}

<!--Estudio Shonos-->

# Gracias {

<Por="Javier Mercado"/>

}