**Paso 1: Investigación y Planificación**

1. **Familiarízate con PySimpleGUI**:
   * Revisa la [documentación oficial](https://www.pysimplegui.org/en/latest/#elements) para entender los elementos disponibles y cómo se utilizan.
2. **Mockups de Interfaz**:
   * Dibuja bocetos de cada ventana que necesitas: la ventana principal, la ventana para agregar categorías, y las ventanas para agregar ingresos y gastos.
   * Identifica los elementos GUI necesarios para cada ventana (botones, tablas, cuadros de texto, etc.).

**Paso 2: Diseño de Datos**

1. **Estructura de Datos**:
   * Decide cómo almacenarás los movimientos financieros en memoria. Por ejemplo, podrías usar listas de diccionarios o un DataFrame de pandas.
   * Define cómo se estructurará cada movimiento (e.g., {"tipo": "gasto", "titulo": "Cena", "monto": 20, "categoria": "Comida"}).
2. **Persistencia de Datos**:
   * Decide el formato para exportar/importar datos (e.g., CSV, JSON).
   * Investiga cómo manejar archivos en Python para leer y escribir datos.

**Paso 3: Descomposición de Funcionalidades**

1. **División en Módulos y Funciones**:
   * Crea un módulo para la interfaz gráfica (e.g., gui.py).
   * Crea un módulo para la lógica de negocio (e.g., logic.py).
   * Crea un módulo para la gestión de datos (e.g., data\_manager.py).
2. **Funcionalidades Clave**:
   * **Interfaz Gráfica**: Implementa las ventanas y elementos GUI.
   * **Gestión de Categorías**: Funciones para agregar y validar categorías.
   * **Gestión de Movimientos**: Funciones para agregar ingresos y gastos, y validar entradas.
   * **Persistencia**: Funciones para exportar e importar datos automáticamente.

**Paso 4: Desarrollo Iterativo**

1. **Desarrollo por Iteraciones**:
   * Implementa y prueba cada funcionalidad de manera incremental.
   * Comienza con la interfaz básica y asegúrate de que los elementos GUI se muestren correctamente.
2. **Pruebas Unitarias**:
   * Desarrolla pruebas para cada función unitaria, especialmente para la lógica de negocio y la gestión de datos.
3. **Manejo de Errores**:
   * Implementa mensajes de error claros, como el que se muestra cuando no hay categorías disponibles.

**Paso 5: Integración y Revisión**

1. **Integración de Módulos**:
   * Asegúrate de que todos los módulos trabajen juntos sin problemas.
   * Verifica que los datos se exporten e importen correctamente.
2. **Revisión y Mejora**:
   * Revisa el código para mejorar la legibilidad y eficiencia.
   * Considera la retroalimentación de usuarios para realizar mejoras.

**Paso 6: Documentación**

1. **Documentación del Código**:
   * Asegúrate de que cada función y módulo esté bien documentado.
   * Escribe un archivo README.md que explique cómo instalar y usar el programa