



**Fe y Alegría**

Movimiento de Educación Popular Integral y Promoción Social  
E C U A D O R



**UNIDAD EDUCATIVA FRANCISCO GRACÍA JIMÉNEZ**

# EXPLORE AND DISCOVERY: SOLUCIONES BASADAS EN DATOS

**Tema: Hábitos del sueño y  
alimenticios de los adolescentes y  
como influyen en el rendimiento  
académico.**



**Programación y Base De Datos**

**Msc. Yoselin Vera Vera**

**AÑO LECTIVO  
2024 - 2025**



**Presentado por:**

**Siomara Angelina  
Valle Sánchez**

# ETAPA 1:

## PLANIFICACIÓN

### PÁGINA 1: METAS Y OBJETIVOS



## 1. METAS:

- Implementar un programa informático que promueva hábitos saludables en adolescentes para mejorar su rendimiento académico.
- Integrar Python y SQL para crear un sistema que analice datos, genere recomendaciones y proporcione guías personalizadas basadas en los hábitos de sueño y alimentación.

## 2. OBJETIVO GENERAL:

- Desarrollar un programa que recopile datos sobre hábitos del sueño, alimenticios, actividad física e higiene para generar recomendaciones personalizadas a los estudiantes y mejorar su rendimiento académico.

## 3. OBJETIVOS ESPECÍFICOS:

- Analizar los hábitos actuales de sueño, alimentación e higiene de los adolescentes a través de una entrada de datos en el programa “VitaliZzz”.
- Implementar un sistema de evaluación personalizado que califique los hábitos y proporcione recomendaciones específicas.
- Desarrollar un menú interactivo que permita a los usuarios explorar opciones de mejora y registre sus datos en una base de datos SQL.

## ETAPA 2:

# ANÁLISIS

## PÁGINA 2: ANÁLISIS DE NECESIDAD



### 1. PROBLEMAS O DESAFÍOS EXISTENTES

- Inconsistencia en los hábitos de sueño en adolescentes, lo que afecta su concentración y memoria.
- Falta de conocimiento sobre cómo una dieta balanceada mejora el desempeño cognitivo.
- Escasez de herramientas accesibles que integren recomendaciones personalizadas basadas en datos reales.

### 2. NECESIDAD DE LA APLICACIÓN

- Fomentar hábitos saludables mediante un programa accesible y fácil de usar.
- Proveer análisis de datos individuales que conecten hábitos diarios con resultados académicos.
- Establecer un sistema dinámico que adapte las recomendaciones según los datos ingresados.

### 3. RECOPIACIÓN (DATOS, VARIABLES, INFORMACIÓN)

#### Variables

#### Datos

#### Información

Horas de sueño

Horas dormidas por noche, frecuencia de interrupciones

Patrón de sueño promedio por usuario, correlación con el rendimiento académico.

Calidad del sueño

Sueño profundo, calidad percibida (escala de 1 a 10)

Evaluación de la calidad del sueño según las respuestas y su impacto en la energía diaria.

Azúcares consumidas

Cantidad diaria (gramos)

Nivel de consumo de azúcares y su correlación con el desempeño escolar y energético.

Calificaciones

Promedio mensual o trimestral

Relación entre rendimiento académico y hábitos analizados.

## ETAPA 2: ANÁLISIS

### PÁGINA 3: ANÁLISIS DE ESTUDIOS DE VIABILIDAD



### 3. RECOPILOCIÓN (DATOS, VARIABLES, INFORMACIÓN)

- **Datos:** números, respuestas en encuestas, mediciones concretas.
- **Información:** análisis o interpretación de los datos.

Datos recolectados mediante preguntas de opción múltiple en el programa y la información será procesada a través del programa en su análisis de datos del usuario.

### 4. FACTORES ECONÓMICOS

El proyecto no genera costos adicionales porque todo el desarrollo se realiza con herramientas gratuitas (Python, SQLite). Si los costos son \$0, se indica que la estudiante es quien realiza todo el trabajo con recursos propios y gratuitos.

### 5. FACTORES TÉCNICOS

- Uso de Python como lenguaje principal debido a su versatilidad.
- Integración de SQLite para la gestión eficiente de bases de datos.
- Bibliotecas adicionales: Pandas para análisis de datos, Matplotlib para gráficos.
- Análisis y visualización de datos (gráficas de progreso, patrones de sueño, etc.).
- Utilización de Visual Studio Code como IDE principal

# ETAPA 2: ANÁLISIS

## PÁGINA 3: ANÁLISIS DE ESTUDIOS DE VIABILIDAD



### 6. FACTORES OPERATIVOS

- Los usuarios (adolescentes) utilizarán el programa a través de una interfaz sencilla para registrar sus datos personales (horas de sueño, alimentos consumidos, calificaciones, etc.).
- El sistema calificará sus hábitos y generará recomendaciones personalizadas según los resultados ingresados.
- Los usuarios recibirán informes personalizados, gráficos o tablas que reflejen su progreso en tiempo real.
- Las recomendaciones incluirán sugerencias prácticas y específicas para mejorar sus hábitos (e.g., horarios de sueño, ajustes en la dieta)

### 7. RECURSOS DISPONIBLES

- Bibliotecas de código abierto para análisis de datos y visualización.
- Documentación extensa de Python y SQL para soporte técnico.

### 8. COSTOS Y BENEFICIOS ESPERADOS

- **Costos:** Tiempo de desarrollo y pruebas, sin costos monetarios significativos.
- **Beneficios:** Mejoras en la salud y el rendimiento académico de los usuarios.

### 9. RIESGOS Y POSIBLES OBSTÁCULOS

- Baja participación en la recolección de datos.
- Posibilidad de errores iniciales en la generación de recomendaciones.

# ETAPA 2: ANÁLISIS

## PÁGINA 4: ANÁLISIS DE REQUISITOS



### 1. NECESIDADES DEL USUARIO

- Tener acceso a guías personalizadas sobre hábitos saludables.
- Recibir recomendaciones específicas basadas en sus datos.
- Interactuar con una interfaz amigable y comprensible.

### 2. COMPONENTES CLAVE DE LA APLICACIÓN

- Un menú interactivo con Opciones para registrar datos y recibir guías.
- Recibir recomendaciones a partir de la calificación que obtengan los usuarios.
- Base de datos SQL para el Almacenamiento eficiente de información.
- Análisis de la información del usuario en la entrada de datos para la generación de recomendaciones basadas en los datos ingresados.

### 3. HERRAMIENTAS PARA EL DIAGRAMA DE FLUJO

Utilizar Lucidchart para Crear diagramas de flujo que expliquen el funcionamiento del sistema.

# ETAPA 2: ANÁLISIS

## PÁGINA 5: ANÁLISIS DE GESTIÓN DE PROYECTOS INFORMÁTICOS



### 1. PLANIFICACIÓN DE TAREAS Y RECURSOS

1. Diseño del sistema y estructura de la base de datos.
2. Desarrollo del menú interactivo y conexión con la base de datos.
3. Implementación de algoritmos para analizar datos.
4. Pruebas del sistema y ajustes finales.

### 2. SEGUIMIENTO Y CONTROL DE AVANCES

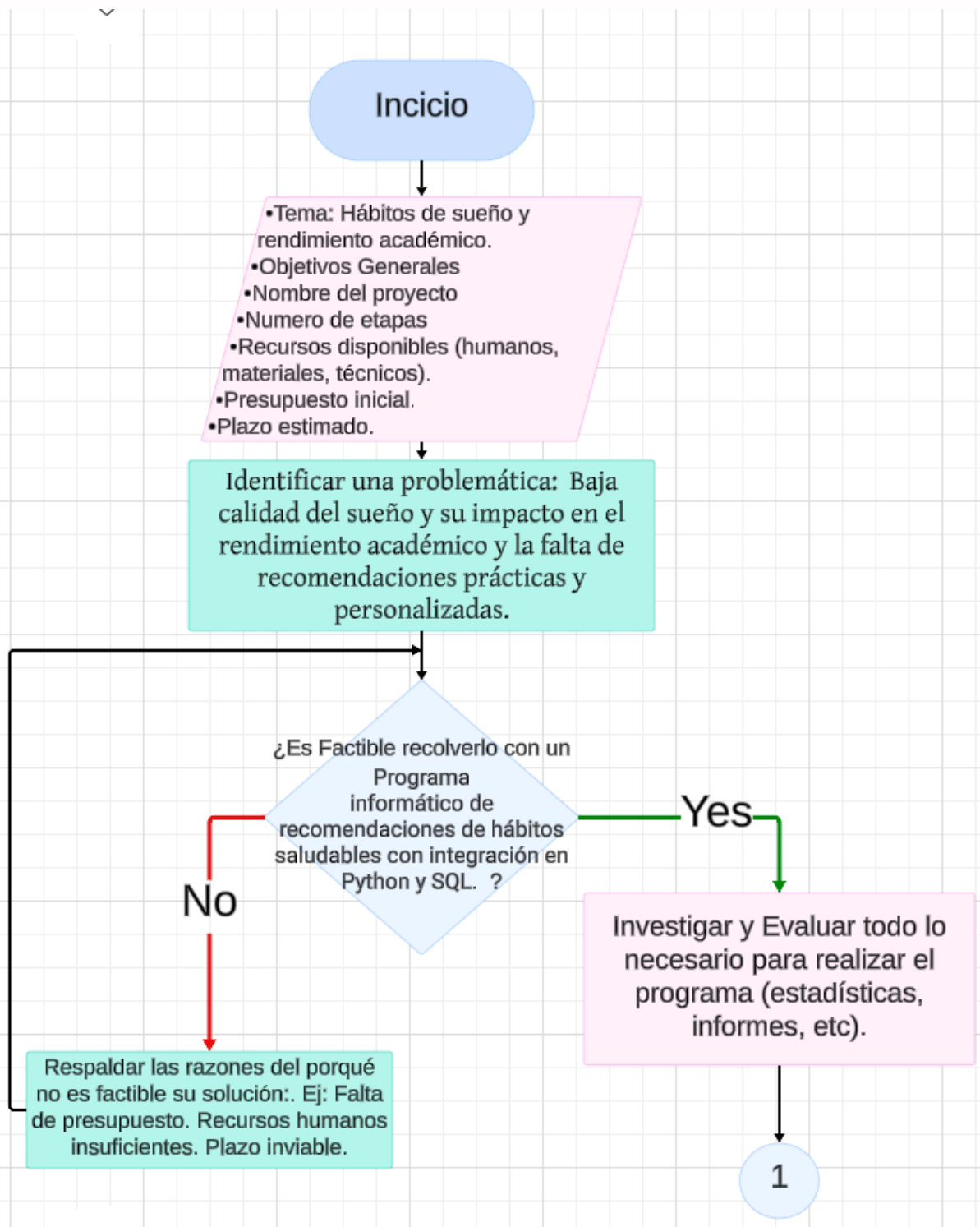
- Revisar el progreso semanalmente para cumplir con los plazos.
- Realizar pruebas unitarias para validar cada módulo antes de integrarlo.
- Retroalimentación/ Feedback de todos los procedimientos que se ha completado.

### 3. EQUIPO DE DESARROLLO

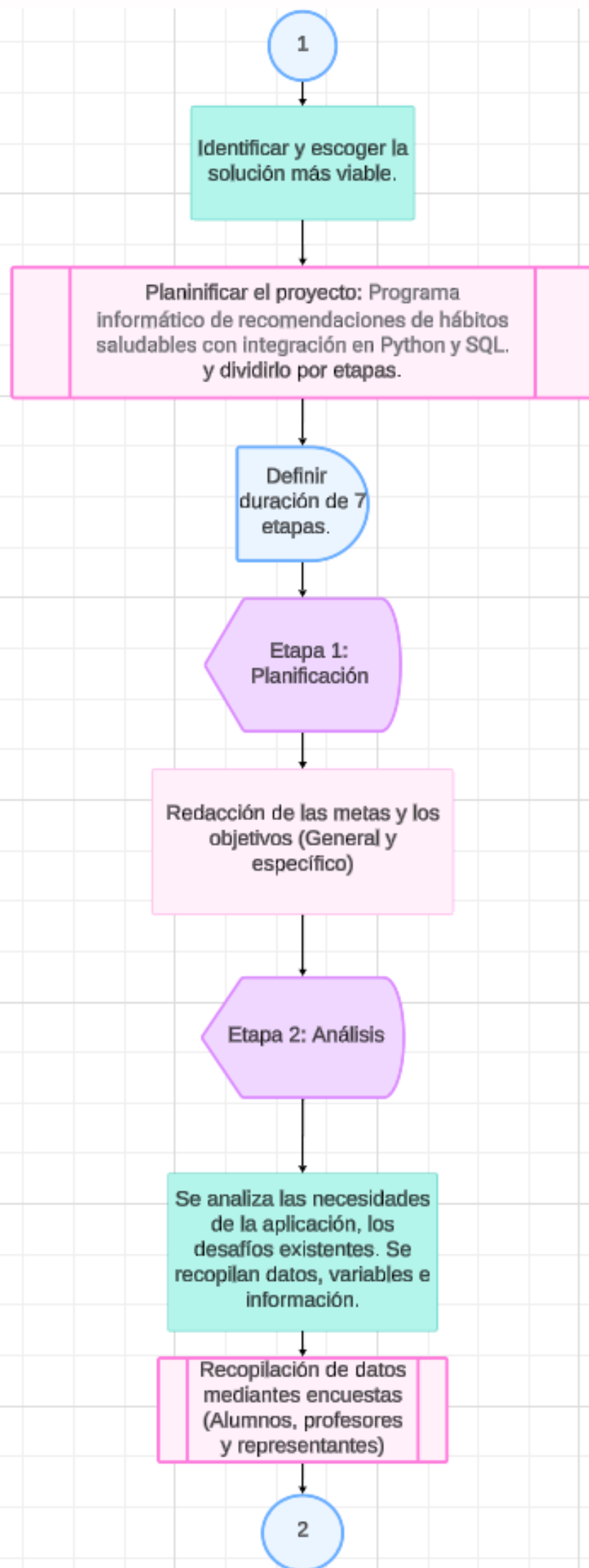
- Personas externas que me pueden enseñar sobre como desempeñar mis habilidades.
- Mi trabajo se sentra en:
  1. Investigación y análisis de datos iniciales (encuestas, entrevistas).
  2. Diseño y desarrollo del programa en Python.
  3. Implementación de la base de datos en SQLite.
  4. Pruebas del programa y ajustes según los resultados.
  5. Documentación del proyecto (objetivos, procesos, y resultados).
- La Función de mi docente técnico es la de Supervisar y guiar el proceso de desarrollo, proporcionando retroalimentación técnica y asegurando que el proyecto cumpla con los objetivos propuestos y la calidad esperada.

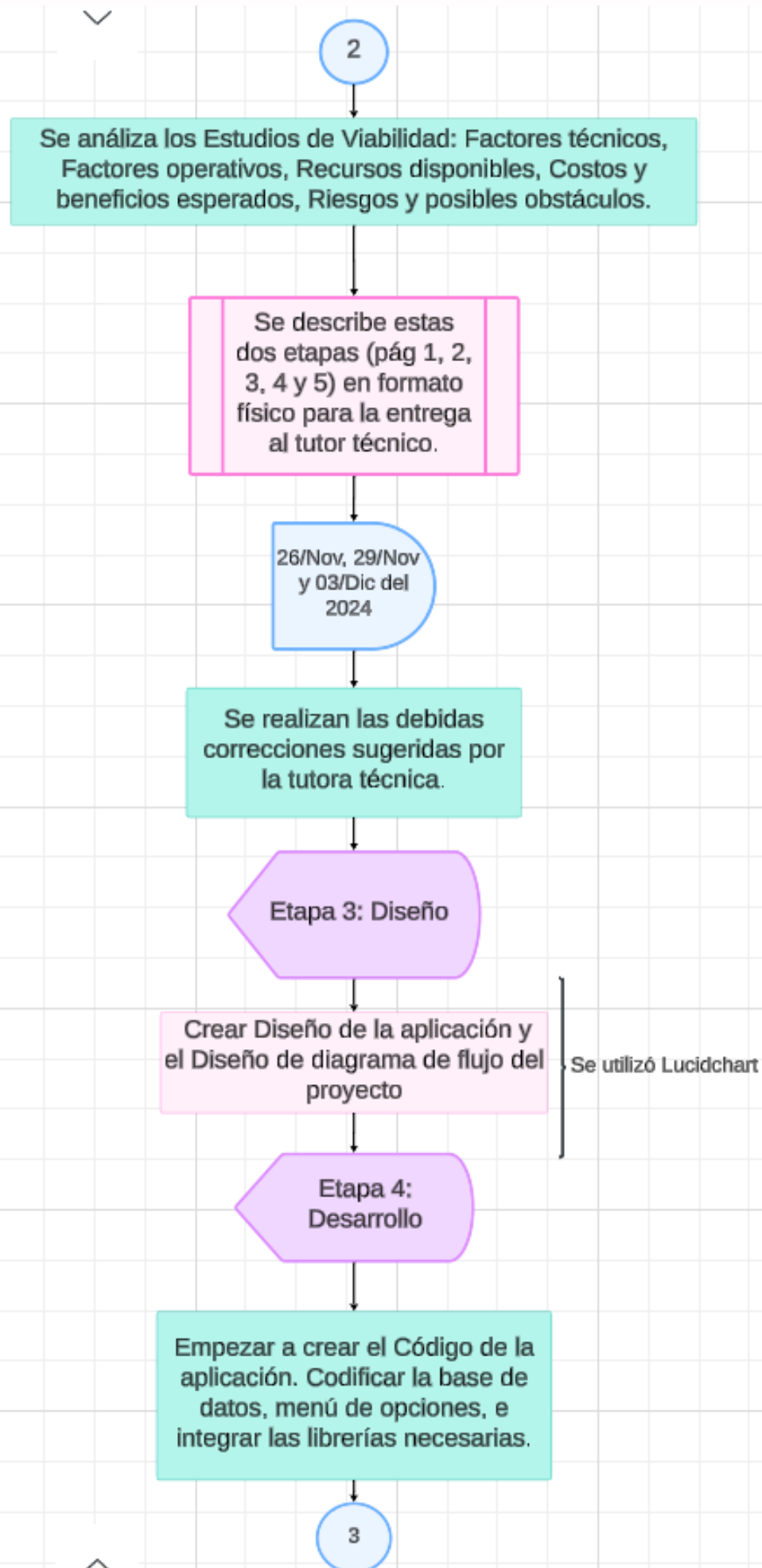
# ETAPA 3:

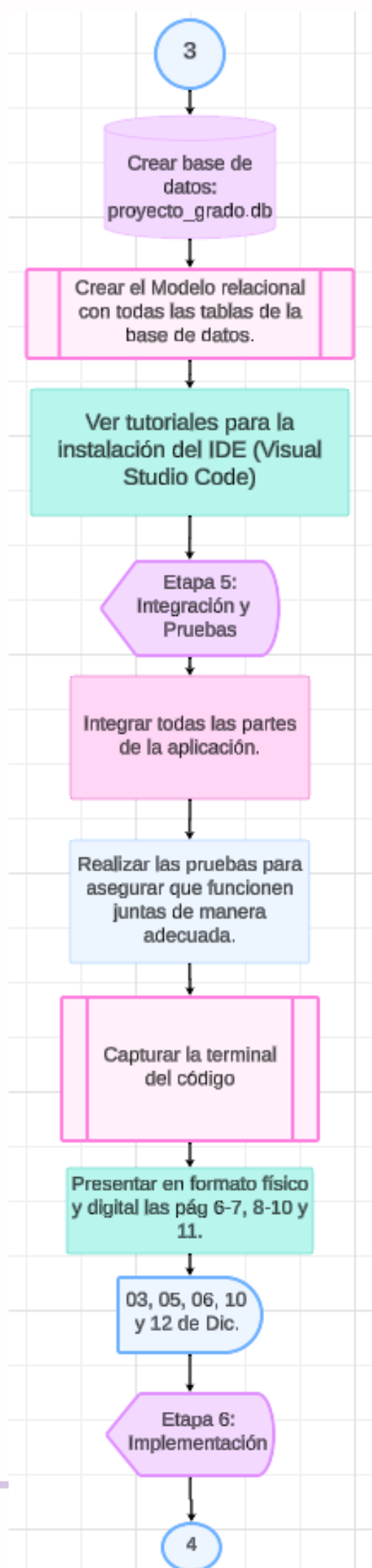
## DIAGRAMA DE FLUJO

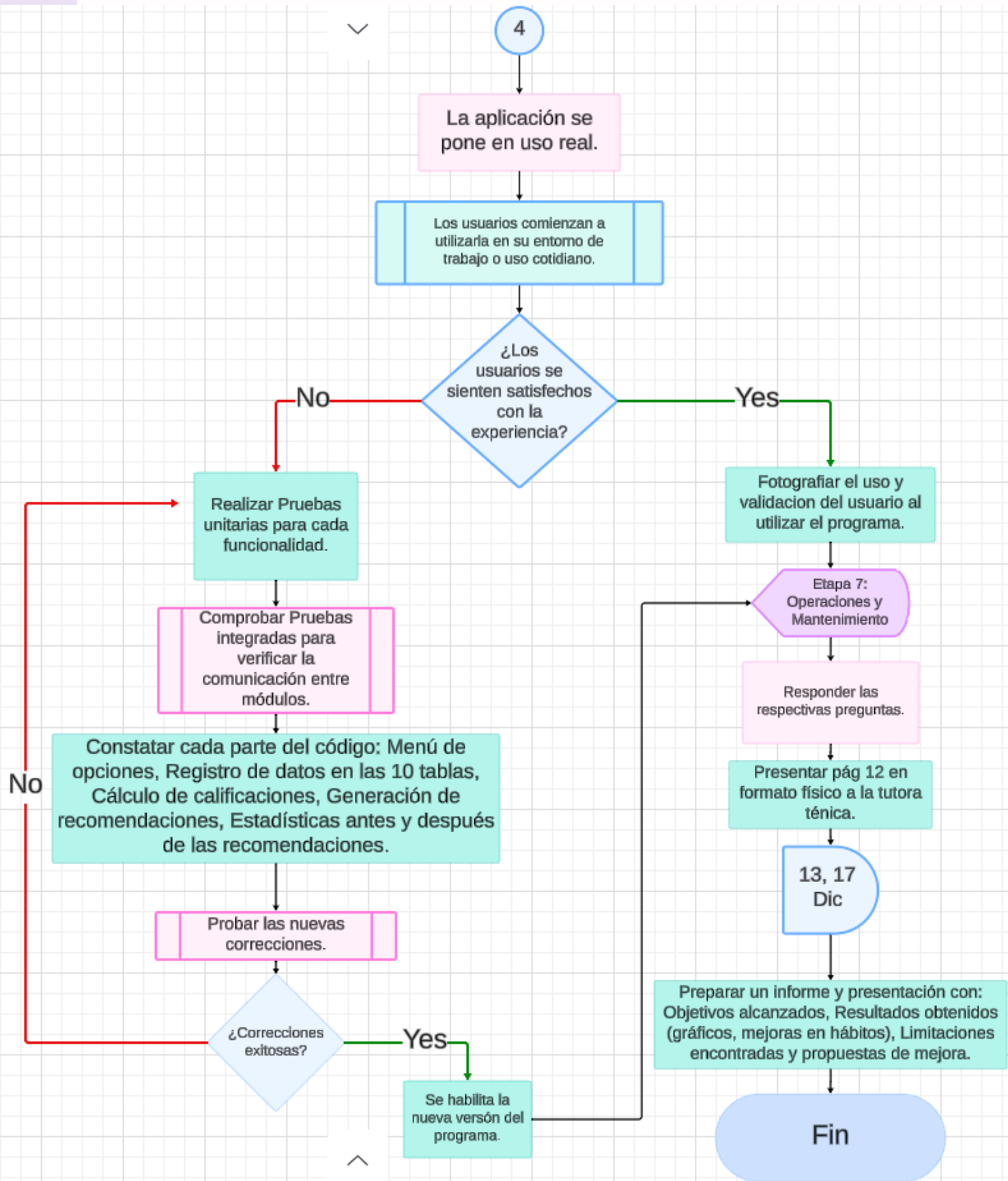












# ETAPA 4

## CÓDIGO DE LA APLICACIÓN (MENÚ DE OPCIONES)



```
import os
```

```
def mostrar_menu():
    print("\n=====")
    print("¡Bienvenido a VitaliZzz!")
    print("Tu guía integral para mejorar tus hábitos de salud y rendimiento académico")
    print("=====")
    print("Selecciona una opción:")
    print("1. Necesitas una dieta balanceada según tus preferencias, edad y peso? Encuentra la guía perfecta aquí.")
    print("2. Conoce tus hábitos de sueño y obtén consejos para descansar mejor.")
    print("3. Descubre cómo una rutina de ejercicios sencilla puede mejorar tu concentración.")
    print("4. Aprende sobre la higiene del sueño y su impacto en tu rendimiento académico.")
    print("5. ¿Sabes cuánta agua debes consumir diariamente? Ajusta tu hidratación para rendir al máximo.")
    print("6. Consejos para reducir el uso de dispositivos electrónicos antes de dormir.")
    print("7. ¿Cómo manejar el estrés académico? Técnicas de relajación y organización.")
    print("0. Salir")
    print("=====")

def ejecutar_opcion(opcion):
    if opcion == "1":
        print("[Guía de Dieta Balanceada]")
        print("Consulta personalizada de dieta para mejorar tu nutrición.")
        # Aquí puedes insertar lógica adicional para mostrar guías específicas
    elif opcion == "2":
        print("[Hábitos de Sueño]")
        print("Analiza tus patrones de sueño y recibe recomendaciones para descansar mejor.")
        # Lógica adicional para guiar al usuario en hábitos de sueño
    elif opcion == "3":
        print("[Ejercicio y Concentración]")
        print("Rutinas de ejercicio simples para mejorar tu enfoque y salud general.")
        # Lógica adicional sobre ejercicios recomendados
    elif opcion == "4":
        print("[Higiene del Sueño]")
        print("Consejos para mantener una rutina de sueño saludable y mejorar tu rendimiento.")
        # Consejos específicos sobre higiene del sueño
    elif opcion == "5":
        print("[Hidratación Adecuada]")
        print("Calcula cuánta agua debes consumir diariamente para mantenerte hidratado.")
        # Incluir cálculo o guía sobre hidratación adecuada
    elif opcion == "6":
        print("[Uso de Dispositivos Electrónicos]")
        print("Consejos para minimizar el uso de dispositivos antes de dormir.")
        # Guía para reducir la exposición a pantallas
    elif opcion == "7":
        print("[Control del Estrés Académico]")
        print("Técnicas de relajación y organización para manejar el estrés.")
        # Recomendaciones y técnicas de manejo del estrés
    elif opcion == "0":
        print("¡Gracias por usar VitaliZzz! ¡Hasta la próxima!")
        exit()
    else:
        print("Opción no válida. Por favor, selecciona una opción del menú.")

def main():
    while True:
        mostrar_menu()
        opcion = input("Ingresa el número de la opción que desees: ")
        ejecutar_opcion(opcion)
        input("\nPresiona Enter para continuar...") # Pausa antes de mostrar el menú nuevamente
        os.system('cls' if os.name == 'nt' else 'clear') # Limpia la consola para mejor legibilidad

if __name__ == "__main__":
    main()
```

# CÓDIGO PYTHON

## (GRÁFICOS, ESTADÍSTICAS)



```
import matplotlib.pyplot as plt
import pandas as pd

# Datos simulados (puedes reemplazar con datos reales de tu encuesta)
datos = {
    'Horas_Sueno': [6, 8, 7, 5, 9],
    'Calidad_Sueno': [7, 8, 6, 5, 9],
    'Horas_Tecnologia': [4, 2, 5, 3, 6]
}

# Crear DataFrame
df = pd.DataFrame(datos)

# Función para generar gráficos
def generar_graficos():
    # Gráfico 1: Promedio de horas de sueño
    plt.figure(figsize=(6, 4))
    plt.bar(['Horas de Sueño'], [df['Horas_Sueno'].mean()], color='blue')
    plt.title('Promedio de Horas de Sueño')
    plt.ylabel('Horas')
    plt.savefig("promedio_horas_sueno.png")
    plt.show()

    # Gráfico 2: Correlación entre sueño y uso de tecnología
    plt.figure(figsize=(6, 4))
    plt.scatter(df['Horas_Sueno'], df['Horas_Tecnologia'], color='green')
    plt.title('Correlación: Sueño vs Tecnología')
    plt.xlabel('Horas de Sueño')
    plt.ylabel('Horas de Tecnología')
    plt.savefig("correlacion_sueno_tecnologia.png")
    plt.show()

    # Gráfico 3: Calidad promedio de sueño
    plt.figure(figsize=(6, 4))
    plt.hist(df['Calidad_Sueno'], bins=5, color='orange', edgecolor='black')
    plt.title('Distribución de la Calidad del Sueño')
    plt.xlabel('Calidad (1-10)')
    plt.ylabel('Frecuencia')
    plt.savefig("calidad_sueno.png")
    plt.show()

# Menú interactivo para el programa
def menu():
    while True:
        print("\n=====")
        print("  Análisis de Encuestas")
        print("=====")
        print("1. Mostrar Promedio de Horas de Sueño")
        print("2. Mostrar Correlación Sueño vs Tecnología")
        print("3. Mostrar Distribución de la Calidad del Sueño")
        print("0. Salir")

        opcion = input("Selecciona una opción: ")

        if opcion == '1':
            print("Generando gráfico del promedio de horas de sueño...")
            generar_graficos()
        elif opcion == '2':
            print("Generando gráfico de correlación entre sueño y tecnología...")
            generar_graficos()
        elif opcion == '3':
            print("Generando gráfico de distribución de calidad del sueño...")
            generar_graficos()
        elif opcion == '0':
            print("Saliendo del programa. ¡Gracias por usarlo!")
            break
        else:
            print("Opción no válida. Intenta nuevamente.")

# Ejecutar el menú
if __name__ == "__main__":
    menu()
```

# CÓDIGO PYTHON

## (BASE DE DATOS, SQL)



```
import sqlite3
```

```
# Crear la conexión a la base de datos
conexion = sqlite3.connect("proyecto_grado.db")
cursor = conexion.cursor()
```

```
# Crear tablas
```

```
# Tabla Usuarios
```

```
cursor.execute("""
```

```
CREATE TABLE IF NOT EXISTS Usuarios (
    id_usuario INTEGER PRIMARY KEY AUTOINCREMENT,
    nombre TEXT NOT NULL,
    edad INTEGER NOT NULL,
    genero TEXT NOT NULL,
    email TEXT,
```

```
)
""")
```

```
# Tabla HabitosSueno
```

```
cursor.execute("""
```

```
CREATE TABLE IF NOT EXISTS HabitosSueno (
    id_habito_sueno INTEGER PRIMARY KEY AUTOINCREMENT,
    id_usuario INTEGER NOT NULL,
    horas_sueno INTEGER,
    calidad_sueno INTEGER,
    hora_acostarse TIME,
    hora_despertarse TIME,
    siestas_diarias BOOLEAN,
    duracion_siestas INTEGER,
```

```
)
""")
```

```
# Tabla Alimentacion
```

```
cursor.execute("""
```

```
CREATE TABLE IF NOT EXISTS Alimentacion (
    id_alimentacion INTEGER PRIMARY KEY AUTOINCREMENT,
    id_usuario INTEGER NOT NULL,
    frutas_consumidas INTEGER,
    verduras_consumidas INTEGER,
    comida_ultraprocesada BOOLEAN,
    agua_consumida_litros DECIMAL,
    frecuencia_comida_rapida INTEGER,
    alergias_alimentarias TEXT,
    horario_comidas TEXT,
```

```
)
""")
```

```
# Tabla Materias
```

```
cursor.execute("""
```

```
CREATE TABLE IF NOT EXISTS Materias (
    id_materia INTEGER PRIMARY KEY AUTOINCREMENT,
    nombre_materia TEXT NOT NULL,
    nivel_dificultad INTEGER,
    profesor TEXT,
    creditos INTEGER
```

```
)
""")
```

```
# Tabla RendimientoAcademico
```

```
cursor.execute("""
```

```
CREATE TABLE IF NOT EXISTS RendimientoAcademico (
    id_rendimiento INTEGER PRIMARY KEY AUTOINCREMENT,
    id_usuario INTEGER NOT NULL,
    id_materia INTEGER NOT NULL,
    asistencias_clase INTEGER,
    total_clases INTEGER,
```

```
)
""")
```

# CÓDIGO PYTHON

## (BASE DE DATOS, SQL)



```
# Tabla DispositivosElectronicos
cursor.execute("""
CREATE TABLE IF NOT EXISTS DispositivosElectronicos (
    id_dispositivo INTEGER PRIMARY KEY AUTOINCREMENT,
    id_usuario INTEGER NOT NULL,
    horas_uso_diario INTEGER,
    uso_antes_de_dormir BOOLEAN,
    tipo_dispositivo TEXT,
    aplicaciones_mas_usadas TEXT,
    hora_uso_maximo TIME,
    modo_noche_activado BOOLEAN,
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
)
""")

# Tabla ActividadFisica
cursor.execute("""
CREATE TABLE IF NOT EXISTS ActividadFisica (
    id_actividad INTEGER PRIMARY KEY AUTOINCREMENT,
    id_usuario INTEGER NOT NULL,
    dias_activos_semana INTEGER,
    minutos_por_sesion INTEGER,
    tipo_actividad TEXT,
    frecuencia_entrenamiento INTEGER,
    deportes_practicados TEXT,
    objetivo_fisico TEXT,
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
)
""")

# Tabla Recomendaciones
cursor.execute("""
CREATE TABLE IF NOT EXISTS Recomendaciones (
    id_recomendacion INTEGER PRIMARY KEY AUTOINCREMENT,
    id_usuario INTEGER NOT NULL,
    tipo_recomendacion TEXT,
    contenido TEXT,
    fecha_recomendacion DATE,
    prioridad INTEGER,
    estado TEXT,
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
)
""")

# Tabla HistoricoRespuestas
cursor.execute("""
CREATE TABLE IF NOT EXISTS HistoricoRespuestas (
    id_respuesta INTEGER PRIMARY KEY AUTOINCREMENT,
    id_usuario INTEGER NOT NULL,
    tipo_encuesta TEXT,
    duracion_encuesta INTEGER,
    calificacion_general DECIMAL,
    fecha_encuesta DATE,
    respuesta_texto TEXT,
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
)
""")

# Confirmar cambios y cerrar conexión
conexion.commit()
conexion.close()

print("Tablas creadas exitosamente en la base de datos 'proyecto_grado.db'")
```



# CÓDIGO PYTHON

## (ENTRADA DE DATOS DE LOS USUARIOS)



```
import sqlite3

# Conexión a la base de datos
conexion = sqlite3.connect("proyecto_grado.db")
cursor = conexion.cursor()

# Crear tabla Usuarios (si no existe)
cursor.execute("""
CREATE TABLE IF NOT EXISTS Usuarios (
    id_usuario INTEGER PRIMARY KEY AUTOINCREMENT,
    nombre TEXT NOT NULL,
    edad INTEGER NOT NULL,
    genero TEXT NOT NULL,
    correo TEXT NOT NULL
)
""")
conexion.commit()

# Función para registrar usuario
def registrar_usuario():
    print("¡Bienvenido al sistema VitaliZzz!")
    print("Por favor, completa la siguiente información:")

    # Solicitar datos al usuario
    nombre = input("Nombre: ")
    while True:
        try:
            edad = int(input("Edad: "))
            break
        except ValueError:
            print("Por favor, ingresa una edad válida en números.")

    genero = input("Género (Masculino/Femenino/Otro): ")
    correo = input("Correo electrónico: ")

    # Guardar datos en la base de datos
    cursor.execute("INSERT INTO Usuarios (nombre, edad, genero, correo) VALUES (?, ?, ?, ?)",
        (nombre, edad, genero, correo))
    conexion.commit()
    print("¡Usuario registrado con éxito!")

# Llamar a la función
registrar_usuario()

# Cerrar conexión
conexion.close()
```

# ETAPA 5

Integración de todas las partes de la aplicación.

- Pruebas para asegurar que funcionen juntas de manera adecuada.
- En esta parte debe adjuntar la validación del código, es decir, que funcione.



```
pbd.py 1  entrada de datos.py X
C: > Users > Point > OneDrive > Documents > entrada de datos.py > registrar_usuario
11     nombre TEXT NOT NULL,
12     edad INTEGER NOT NULL,
13     genero TEXT NOT NULL,
14     correo TEXT NOT NULL
15 )
16 '''
17 conexion.commit()
18
19 # Función para registrar usuario
20 def registrar_usuario():
21     print("¡Bienvenido al sistema VitaliZzz!")
22     print("Por favor, completa la siguiente información:")
23
24     # Solicitar datos al usuario
25     nombre = input("Nombre: ")
26     while True:
27         try:
28             edad = int(input("Edad: "))
29             break
30         except ValueError:
31             print("Por favor, ingresa una edad válida en números.")
32
33     genero = input("Género (Masculino/Femenino/Otro): ")
34     correo = input("Correo electrónico: ")
35
36     # Guardar datos en la base de datos
37     cursor.execute("INSERT INTO Usuarios (nombre, edad, genero, correo)
38     | | | | (nombre, edad, genero, correo))
39     conexion.commit()
40     print("¡Usuario registrado con éxito!")
41
42 # Llamar a la función
43 registrar_usuario()
44
```

(Entrada de datos de los usuarios)

PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL RESULTADOS DE LA PRUEBA

```
PS C:\Users\Point\OneDrive\Documents> & 'c:\Users\Point\AppData\Local\Microsoft\
.\debugpy\launcher' '55798' '--' 'C:\Users\Point\OneDrive\Documents\entrada de da
¡Bienvenido al sistema VitaliZzz!
Por favor, completa la siguiente información:
Nombre: Siomara Angelina Valle Sánchez
Edad: 17
Género (Masculino/Femenino/Otro): Femenino
Correo electrónico: angelina21sioma@gmail.com
¡Usuario registrado con éxito!
PS C:\Users\Point\OneDrive\Documents> █
```

Terminal

# PRUEBAS

## (ESTADÍSTICAS Y GRÁFICOS)



Code editor showing Python code for data analysis and visualization:

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 # Datos simulados (puedes reemplazar con datos reales de tu encuesta)
5 datos = {
6     'Horas_Sueno': [6, 8, 7, 5, 9],
7     'Calidad_Sueno': [7, 8, 6, 5, 9],
8     'Horas_Tecnologia': [4, 2, 5, 3, 6]
9 }
10
11 # Crear DataFrame
12 df = pd.DataFrame(datos)
13
14 # Función para generar gráficos
15 def generar_graficos():
16     # Gráfico 1: Promedio de horas de sueño
17     plt.figure(figsize=(6, 4))
18     plt.bar(['Horas de Sueño'], [df['Horas_Sueno'].mean()], color='blue')
19     plt.title('Promedio de Horas de Sueño')
20     plt.ylabel('Horas')
21     plt.savefig("promedio_horas_sueno.png")
22     plt.show()
23
24     # Gráfico 2: Correlación entre sueño
25     plt.figure(figsize=(6, 4))
26     plt.scatter(df['Horas_Sueno'], df['Horas_Tecnologia'])
27     plt.title('Correlación: Sueño vs Tecnología')
28     plt.xlabel('Horas de Sueño')
29     plt.ylabel('Horas de Tecnología')
30     plt.savefig("correlacion_sueno_tecnologia.png")
31     plt.show()
32
33     # Gráfico 3: Calidad promedio de sueño
34     plt.figure(figsize=(6, 4))
```

Output window showing the first graph:

Figure 1

Promedio de Horas de Sueño

Horas

Horas de Sueño

(x, y) = (Horas de Sueño, 0.72)

Terminal output:

```
PS C:\Users\Point\OneDrive\Documents>
PS C:\Users\Point\OneDrive\Documents> c:: cd 'c:\Us
-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy

=====
Análisis de Encuestas
=====
1. Mostrar Promedio de Horas de Sueño
2. Mostrar Correlación Sueño vs Tecnología
3. Mostrar Distribución de la Calidad del Sueño
0. Salir
Selecciona una opción: 1
Generando gráfico del promedio de horas de sueño...
```

Terminal

Gráfica

Siomara Valle

# PRUEBAS

## (MENÚ DE OPCIONES)



```
pbd.py 1  menú de opciones.py X  entrada de datos.py  Estadísticas.py

C: > Users > Point > OneDrive > Documents > menú de opciones.py > ...

3  def mostrar_menu():
4
5      print("¡Bienvenido a VitalizZZ!")
6      print("Tu guía integral para mejorar tus hábitos de salud y rendimiento académico")
7      print("=====")
8      print("Selecciona una opción:")
9      print("1. Necesitas una dieta balanceada según tus preferencias, edad y peso? Encuentra")
10     print("2. Conoce tus hábitos de sueño y obtén consejos para descansar mejor.")
11     print("3. Descubre cómo una rutina de ejercicios sencilla puede mejorar tu concentración")
12     print("4. Aprende sobre la higiene del sueño y su impacto en tu rendimiento académico.")
13     print("5. ¿Sabes cuánta agua debes consumir diariamente? Ajusta tu hidratación para ren")
14     print("6. Consejos para reducir el uso de dispositivos electrónicos antes de dormir.")
15     print("7. ¿Cómo manejar el estrés académico? Técnicas de relajación y organización.")
16     print("0. Salir")
17     print("=====")
18
19     def ejecutar_opcion(opcion):
20         if opcion == "1":
21             print("\n[Guía de Dieta Balanceada]")
22             print("Consulta personalizada de dieta para mejorar tu nutrición.")

PROBLEMAS 1  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  RESULTADOS DE LA PRUEBA  PUERTOS

=====
Selecciona una opción:
1. Necesitas una dieta balanceada según tus preferencias, edad y peso? Encuentra la guía perfecta aquí.
2. Conoce tus hábitos de sueño y obtén consejos para descansar mejor.
3. Descubre cómo una rutina de ejercicios sencilla puede mejorar tu concentración.
4. Aprende sobre la higiene del sueño y su impacto en tu rendimiento académico.
5. ¿Sabes cuánta agua debes consumir diariamente? Ajusta tu hidratación para rendir al máximo.
6. Consejos para reducir el uso de dispositivos electrónicos antes de dormir.
7. ¿Cómo manejar el estrés académico? Técnicas de relajación y organización.
0. Salir
=====
Ingresa el número de la opción que deseas: 1

[Guía de Dieta Balanceada]
Consulta personalizada de dieta para mejorar tu nutrición.

Presiona Enter para continuar...[ ]
```

Terminal



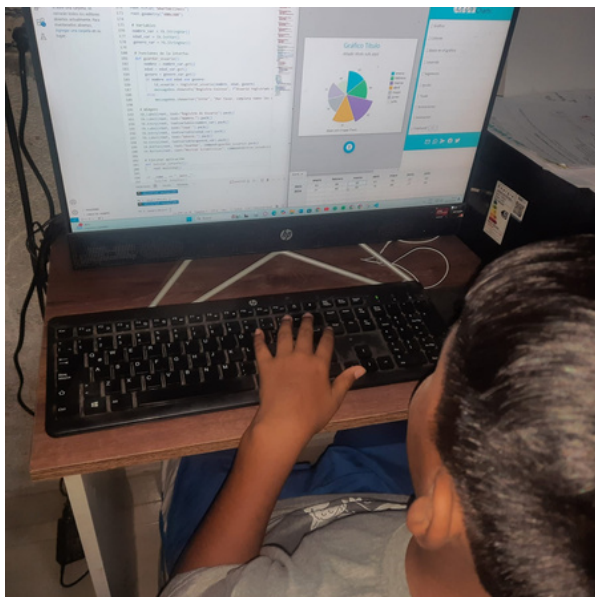
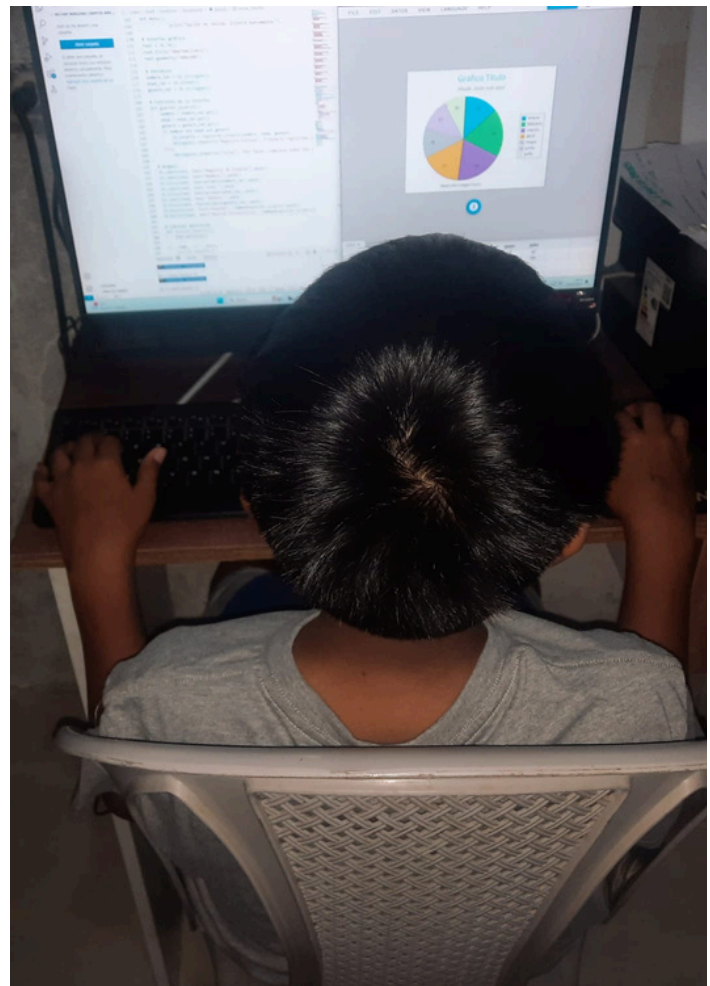
# ETAPA 6:

- La aplicación se pone en uso real.
- Los usuarios comienzan a utilizarla en su entorno de trabajo o uso cotidiano.

En esta parte debe agregar alguna evidencia de que su app, aplicación, base de datos o sistema es utilizado por algún usuario.



## Evidencias:



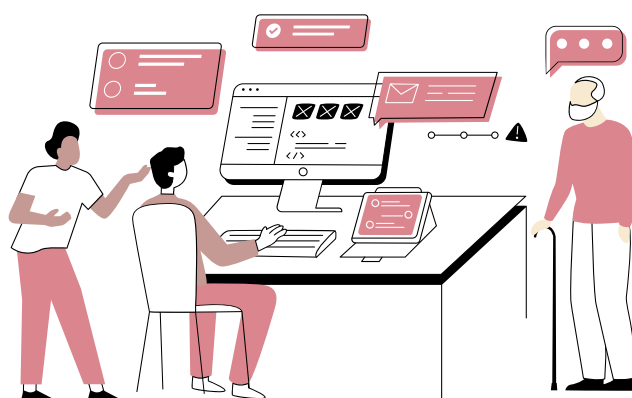
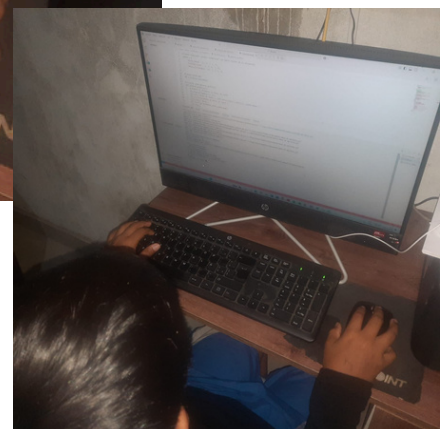
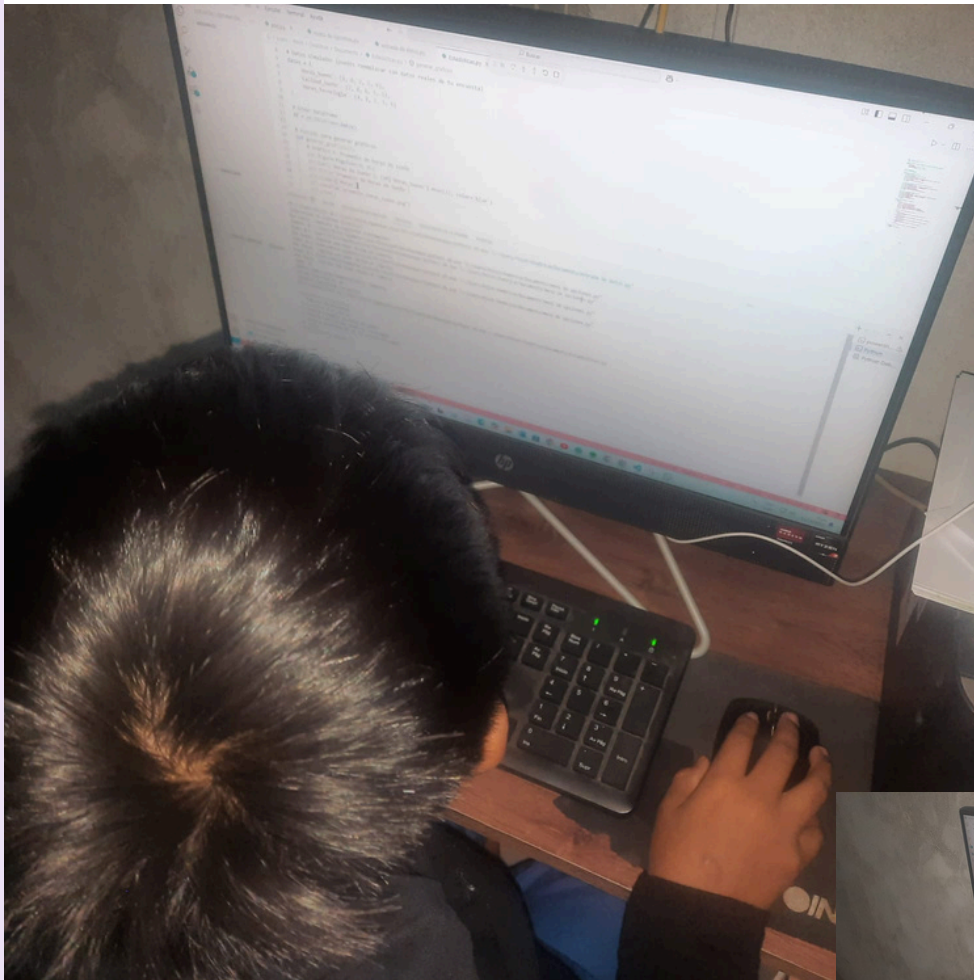
# ETAPA 6:

- La aplicación se pone en uso real.
- Los usuarios comienzan a utilizarla en su entorno de trabajo o uso cotidiano.

En esta parte debe agregar alguna evidencia de que su app, aplicación, base de datos o sistema es utilizado por algún usuario.



## Evidencias:



# ETAPA 7:

**Responder a las siguientes preguntas:**



## 1. ¿Cómo se supervisará y mantendrá la aplicación en funcionamiento?

El programa incluye validaciones básicas para asegurar que los datos ingresados sean correctos (por ejemplo, que la edad sea un número válido). Realizar pruebas semanales en la institución( en caso de estar ausente, delegar) para verificar que todas las funcionalidades, como el registro de datos, las recomendaciones, y las estadísticas, funcionen sin problemas.

## 2. ¿Cómo se gestionarán las actualizaciones y correcciones de errores?

Implementaré mejoras o nuevas funcionalidades en períodos definidos (por ejemplo, cada trimestre). Esto puede incluir agregar nuevas preguntas al registro o ajustar el cálculo de las recomendaciones.

Mantener un archivo o tabla en la base de datos para registrar errores reportados por los usuarios.

- Correcciones Rápidas:
  - Si detectas un error crítico, lo puedes corregir directamente desde Visual Studio y actualizar el programa.
  - Para garantizar que otros puedan realizar actualizaciones, incluiré una sección en un manual donde explique cómo modificar el código base para cambios simples (como agregar tablas o ajustar gráficos).

## 3. ¿Cuál es el plan de respaldo en caso de fallos o interrupciones en el servicio?

- Manual de Usuario:
  - Incluiré una sección específica en el manual con soluciones prácticas, como:
    - Error de base de datos: Cómo crear nuevamente el archivo .db desde el código.
    - Error en el programa: Cómo reinstalar Python y las bibliotecas necesarias.
    - Gráficos no visibles: Solución para instalar o reinstalar Matplotlib.
  - Ejemplo de mensaje en el manual:
  - "Si aparece el error 'No se encuentra la base de datos', vaya a Visual Studio, ejecute el programa y automáticamente se generará un nuevo archivo de base de datos."
- Respaldo de la Base de Datos:
  - Guardar regularmente el archivo de base de datos proyecto\_grado.db en una carpeta segura o en la nube (por ejemplo, Google Drive). Esto asegura que, si el archivo se pierde o daña, pueda restaurarse fácilmente.

## 4. ¿Cómo se recopilarán y utilizarán los comentarios de los usuarios para mejorar la aplicación?

Crearé un formulario sencillo en papel o digital donde los usuarios puedan reportar problemas o sugerencias (por ejemplo: "¿Qué te gustaría que haga el programa?").

- Preguntas clave:
  - i. ¿Qué te gusta del programa?
  - ii. ¿Qué crees que podría mejorar?
  - iii. ¿Tuviste algún problema usándolo?
- Recolectaré las respuestas cada mes y priorizaré las mejoras más solicitadas.
- Usaré los comentarios para planificar nuevas versiones del programa y mejorar su funcionalidad según las necesidades reales de los usuarios.