

# ASISTENTE PERSONAL PARA EL CONTROL DE PESO

**Cliente:** Miguel

**Necesidad:**

Miguel es un joven interesado en mejorar sus hábitos alimenticios, pero le resulta difícil llevar un control manual de lo que consume a lo largo del día. Actualmente, anota sus comidas en una libreta, pero esto genera desorden y no le permite analizar fácilmente si está siguiendo una dieta balanceada.

**Objetivo del Proyecto:**

Desarrollar una aplicación de escritorio en Python con PyQt5 que le permita:

1. Registrarse e iniciar sesión de manera segura para mantener un historial personalizado.
2. Registrar sus comidas (desayuno, comida o cena) de forma automática según la hora del día.
3. Almacenar los datos en una base de datos local para futura consulta y análisis.

**Beneficios para Miguel:**

✓ **Organización:** Ya no dependerá de notas físicas.

✓ **Automatización:** El sistema clasificará sus comidas según la hora.

✓ **Seguimiento:** Podrá revisar su alimentación en cualquier momento.

## Secciones

### *1. Configuración Inicial*

- **Estructura del proyecto:**

```
/proyecto_comidas
├── main.py           (script principal)
├── database.py       (manejo de la base de datos)
├── login_window.py   (interfaz de login/signup)
└── food_form.py      (formulario de registro de comidas)
```

## 2. Interfaz de Login/Signup (PyQt5)

- **Archivo:** login\_window.py.
- **Elementos:**
  - Dos pestañas (Login / Signup).
  - Campos: Usuario, Contraseña (y confirmación en Signup).
  - Botones: Iniciar Sesión, Registrarse.
- **Lógica:**
  - **Signup:**
    - Validar que el usuario no exista en users.
    - Guardar datos en users y credentials.
    - Mostrar mensaje: *"Registro exitoso. Ahora inicia sesión."*.
  - **Login:**
    - Verificar credenciales en credentials.
    - Si son correctas, abrir food\_form.py.
    - Si no, sugerir: *"Usuario no registrado. Por favor, regístrate."*.

## 3. Formulario de Comidas (PyQt5)

- **Archivo:** food\_form.py.
- **Elementos:**
  - 5 campos de texto para productos consumidos.
  - Selector de hora (usar QTimeEdit).
  - Botón Guardar.
- **Lógica:**
  - Al enviar el formulario:
    - Obtener la hora ingresada.
    - Determinar el tipo de comida:
      - **Desayuno:** Si hora < 12:00.
      - **Comida:** Si  $12:00 \leq \text{hora} < 16:00$ .
      - **Cena:** Si hora  $\geq 18:00$ .
    - Guardar cada producto en products con su tipo.
    - Mostrar confirmación: *"Comida registrada correctamente."*.

## 5. Flujo del Programa

1. Ejecutar main.py (contiene la ventana principal).
2. Mostrar login\_window.py.
3. Si el login es exitoso, cerrar la ventana de login y abrir food\_form.py.

4. El usuario registra comidas hasta cerrar la aplicación.

## 6. Recomendaciones

- **Pruebas:**
  - Verificar que no se puedan registrar usuarios duplicados.
  - Probar el guardado de comidas con horas límite (12:00, 16:00, 18:00).
- **Extras (opcional):**
  - Añadir un menú de visualización de comidas guardadas.
  - Implementar hashing para contraseñas (ej: `hashlib`).

## Anexos: Diseño UI/UX

### 1. Paleta de Colores (Tema Moderno y Minimalista)

- **Fondo principal:** #F5F5F5 (gris claro) o #FFFFFF (blanco puro).
- **Botones/Énfasis:** #4CAF50 (verde fresco, transmite salud) o #2196F3 (azul confiable).
- **Texto principal:** #333333 (gris oscuro) para legibilidad.
- **Alertas/Éxito:** #FF9800 (naranja para advertencias), #4CAF50 (verde para confirmaciones).
- **Header/Barra superior:** #2E7D32 (verde oscuro) o #1565C0 (azul oscuro).

### 2. Tamaños de Ventana

- **Login/Signup:**
  - 400x300 px (compacta, centrada en la acción principal).
- **Formulario de comidas:**
  - 600x500 px (espacio suficiente para 5 campos + selector de hora).

### 3. Tipografía

- **Títulos:** Segoe UI Bold (o Arial Bold) tamaño 16-18px.
- **Texto normal:** Segoe UI (o Roboto) tamaño 12-14px.
- **Placeholders:** Mismo tipo de letra en #888888 (gris claro).

## 4. Diseño de Formularios

- **Campos de texto (QLineEdit):**
  - Altura: 30px, bordes redondeados (border-radius: 5px).
  - Borde sutil: 1px solid #CCCCCC.
- **Botones (QPushButton):**
  - Padding: 10px 20px, bordes redondeados (8px).
  - Efecto hover: Cambiar a un tono más oscuro (ej: #3E8E41 para verde).
- **Selector de hora (QTimeEdit):**
  - Ajustar a formato HH:MM AM/PM (más intuitivo).






## 5. Distribución de Elementos

- **Login/Signup:**
  - Logo (opcional) en la parte superior.
  - Campos centrados verticalmente, botones alineados a la derecha.
  - Enlace tipo "*¿No tienes cuenta? Regístrate aquí*" (alternar entre pestañas).
- **Formulario de comidas:**
  - Grid layout para los 5 campos de productos.
  - Selector de hora y botón "*Guardar*" en la parte inferior.
  - Mensaje de confirmación flotante (ej: "*¡Comida guardada!*").

## 6. Feedback Visual

- **Validación de campos:**
  - Borde rojo (#FF0000) si hay error, verde (#4CAF50) si es válido.
- **Animaciones simples:**
  - Transición suave al cambiar entre ventanas (QPropertyAnimation).
  - Efecto de carga al guardar datos (ej: QProgressBar oculto).

## 7. Iconografía (Opcional pero Recomendado)

- Usar íconos de [Material Design Icons](#) o [FontAwesome](#):
  - Login:  o ícono de usuario.
  - Comida:  (desayuno),  (almuerzo),  (cena).
  - Guardar:  o ícono de disco.

## 8. Responsividad

- Asegurar que los elementos se redimensionen correctamente:
  - Usar QHBoxLayout, QVBoxLayout, y QGridLayout.
  - Márgenes uniformes (10-15px entre elementos).