Análisis Técnico: Módulo PacienteFrame

## Propósito del módulo

El módulo define la clase `PacienteFrame`, una interfaz gráfica desarrollada con CustomTkinter para el registro de pacientes. Incluye campos de entrada, un selector de fecha (DateEntry) y un mapa interactivo. Es parte de un sistema modular de gestión (presumiblemente médica o administrativa).

## Estructura General

La clase `PacienteFrame` hereda de `Marcos\_P`, una clase personalizada basada en `CTkFrame`. Está compuesta por dos secciones principales:

1. Panel izquierdo: Formulario para ingresar datos personales del paciente.

2. Panel derecho: Visualización de un mapa referencial (`MapWidget`).

## Inicialización (\_\_init\_\_)

La inicialización crea y organiza la interfaz usando `grid()` y `pack()` de Tkinter. Define el layout general y los widgets interactivos.

- `contenedor`: Divide la pantalla en dos columnas (formulario y mapa).

- `frame\_datos`: `CTkScrollableFrame` que contiene todos los campos del formulario.

- `frame\_mapa`: `CTkFrame` donde se carga el componente del mapa.

## Componentes del Formulario

El formulario incluye:

- Campos de texto (`Entradas\_P`) para: Nombres, Apellidos, Cédula, Email, Teléfono, Dirección.

- Calendario interactivo (`tkcalendar.DateEntry`) para seleccionar la fecha de nacimiento.

- Botones de acción: Guardar y Limpiar.

## Funcionalidades

`guardar\_paciente(self)`:

- Obtiene la fecha seleccionada en el calendario.

- Imprime la fecha formateada en consola (`YYYY-MM-DD`).

- Está listo para expandirse, por ejemplo: guardar en una base de datos.

`limpiar\_campos(self)`:

- Recorre todos los widgets del formulario y borra su contenido si son instancias de `Entradas\_P`.

- Reinicia el calendario a un valor en blanco o la fecha actual.

- Esta función garantiza que el formulario pueda reutilizarse sin cerrar la interfaz.

## Mapa Referencial

Se integra mediante la clase `MapWidget` importada desde `Ficheros.MapsView`. Se presenta en el marco derecho (`frame\_mapa`) y ocupa todo el espacio disponible mediante `expand=True`.

## Diseño y Layout

- Uso combinado de `grid()` (estructura) y `pack()` (contenido interno).

- Uso de `weight` en filas y columnas para que los elementos sean responsivos.

- `CTkScrollableFrame` permite manejar formularios más largos que el alto visible.

## Personalización y Reutilización

- Componentes personalizados (`Marcos\_P`, `Entradas\_P`, `Etiqueta\_P`, `Botones\_P`) facilitan reutilización.

- Diseño modular e integrable con controladores y sistemas de navegación por vistas.

## Posibles mejoras futuras

- Validación de campos.

- Almacenamiento de datos (BD, JSON, etc.).

- Mensajes visuales con `CTkMessagebox`.

- Integración con APIs para geolocalización o marcadores.

## Dependencias del módulo

- `customtkinter`: estilos modernos.

- `tkcalendar`: widget de fecha.

- `Ficheros.MapsView.map\_widget`: componente de mapa.

- `Layouts.layout`: widgets personalizados y estilos.