

1. Descripcion general

MacrosGym es una aplicacion escrita con Python 3.11 y Flet orientada a organizar macros, micros, rutinas y progreso de entrenamiento dentro de una sola experiencia mobile-friendly. Cada feature vive en un modulo desacoplado y la navegacion principal se resuelve con un `NavigationBar` y una hoja flotante para agregar registros.

2. Sintaxis y arquitectura por modulos

- `main.py`: Define la pagina Flet, aplica el tema oscuro, configura la barra de navegacion y conecta cada vista declarada en `features/`. Tambien carga variables desde `.env.local` y gestiona el `BottomSheet` para las acciones rapidas.
- `features/home.py`: Pantalla de bienvenida con resumen de calorías, accesos directos y tarjetas informativas que utilizan estilos responsive declarados en `responsive.py`.
- `features/macros.py` y `features/micros.py`: Formularios controlados que registran consumos de macro y micronutrientes, guardando la informacion en `macroentreno.json` mediante helpers especializados.
- `features/workouts.py`: Gestor de rutinas con favoritos, chips por dia de la semana y persistencia en `data/workouts.json` usando `dataclasses` y funciones CRUD dedicadas.
- `features/progress.py`: Tarjetas y graficos que resumen calorías, balance y rachas semanales aprovechando controles de Flet como `LineChart` y `ProgressRing`.

3. Persistencia y configuraciones

Los datos principales viven en `macroentreno.json`, mientras que los entrenamientos persisten en `data/workouts.json`. La carpeta `services/` centraliza reglas de negocio y el archivo `.env.local` permite definir claves externas sin exponerlas en el repositorio.

4. Ejecucion en modo desarrollo

- Crear un entorno virtual, instalar dependencias con `pip install -r requirements.txt` y ejecutar `python main.py` para levantar la aplicacion en modo Flet desktop.
- Los assets declarados se cargan automaticamente gracias al parametro `assets_dir` especificado al invocar `ft.app`.

5. Construccion del ejecutable

- `PyInstaller` utiliza `main.spec` para reproducir `dist/main.exe`. Se puede regenerar ejecutando: `pyinstaller --clean --noconfirm main.spec`.
- El ejecutable empaqueta Flet, los modulos de `features` y la carpeta `assets`, permitiendo probar el proyecto sin instalar Python en el equipo evaluador.

6. Recomendaciones de prueba

- Iniciar en Home, abrir la hoja + y cargar datos de macros, micros y ejercicios; cada insercion debe verse reflejada en los listados y en la vista Progreso.
- Para reiniciar el entorno de pruebas basta eliminar `macroentreno.json` y `data/workouts.json`; el sistema recrea automaticamente plantillas vacias al iniciarse.