

Manual del Programador

Estructura del proyecto

...

flask-app/

└─ app/

```
| └─ __init__.py      # Inicialización Flask y SQLAlchemy
| └─ config.py        # Configuración de la app (correo, BD, etc.)
| └─ padel_app.py     # Rutas y lógica principal
| └─ models.py        # Modelos de la base de datos
| └─ correo.py        # Función para envío de emails
| └─ utils.py         # Validaciones y utilidades
| └─ init_db.py       # Script para inicializar la BD y datos por defecto
| └─ run.py           # Arranque de la app
| └─ static/          # Archivos estáticos (CSS, JS, imágenes)
|   └─ templates/     # Plantillas HTML (Jinja2)
└─ requirements.txt   # Dependencias del proyecto
└─ README.md          # Descripción general y guía rápida
```

...

Instalación y ejecución

1. Clona el repositorio

```
`git clone ...`
```

2. Instala dependencias

```
`pip install -r requirements.txt`
```

3. Inicializa la base de datos

```
`python app/init_db.py`
```

4. Arranca la aplicación

```
`python app/run.py`
```

5. Accede en tu navegador a <http://localhost:5000>

Configuración

- Edita `config.py` para los parámetros de correo, base de datos, etc.
- Variables sensibles pueden ir en un `.env` (opcional).

Modelos principales

- Usuario: datos personales, login, admin.
- Club: información de cada club.
- Reserva: reservas de pistas.
- Torneo: gestión de torneos e inscripciones.
- Noticia: noticias del club.

Lógica principal

- Rutas en `padel_app.py`:
 - `/register`, `/login`, `/logout`, `/reservar_pista`, `/torneos`, `/noticias`,
`/admin_panel`, etc.
- Validaciones: en `utils.py`.
- Envío de emails: en `correo.py` usando Flask-Mail.
- Transacciones: uso de transacciones explícitas y rollback en operaciones críticas.

Personalización

- Estilos: en `static/css/styles.css`.
- JS de reservas: en `static/calendar.js`.
- Plantillas: en `templates/`.

Despliegue

- Puedes desplegar en cualquier servidor compatible con Flask (Heroku, PythonAnywhere, etc.).