

# Instrucciones Extendidas - CRUNEVO

## 1. Introducción

CRUNEVO es una plataforma desarrollada con Flask y Python pensada para compartir apuntes, interactuar como en una red social académica, y fomentar la colaboración entre estudiantes. Este documento reúne todas las instrucciones necesarias para entender cómo se estructura el proyecto, cómo desplegarlo, personalizarlo y escalarlo en el futuro.

## 2. Estructura del Proyecto

La estructura de carpetas incluye una carpeta principal 'CRUNEVO' que contiene un subdirectorío también llamado 'crunevo', donde están los módulos como `__init__.py`, `models`, `routes`, `templates` y `static`. Esto permite separar la lógica del núcleo del entorno general del proyecto.

- `run.py`: Archivo principal de ejecución que importa `create_app` desde `crunevo.__init__`
- `.env`: Archivo oculto que contiene claves como `SECRET_KEY` y `SQLALCHEMY_DATABASE_URI`
- `requirements.txt`: Lista de dependencias a instalar
- `config.py`: Clase `Config` con las configuraciones base para Flask

## 3. Base de Datos y Modelos

Se utiliza SQLAlchemy como ORM para conectar con la base de datos. Los modelos se encuentran en `crunevo/models` y definen las tablas como usuarios, notas, productos, etc.

Recuerda ejecutar:

```
>>> from crunevo import create_app
>>> from crunevo.models import db
>>> app = create_app()
>>> with app.app_context(): db.create_all()
```

## 4. Blueprints y Rutas

Cada conjunto de funcionalidades (`auth`, `admin`, `store`, `notes`) se maneja como un blueprint en la carpeta `routes`. Por ejemplo, `auth_routes.py` gestiona el login y registro, mientras `store_routes.py` gestiona la tienda. Todos se registran en `__init__.py` dentro de `create_app()`.

## Instrucciones Extendidas - CRUNEVO

### 5. Despliegue en Render y GitHub

Para desplegar en Render:

1. Crear un repositorio GitHub con la carpeta principal CRUNEVO
2. Agregar un archivo `run.py` con la línea: `from crunevo import create_app`  
`app = create_app()`
3. En Render, crear un nuevo servicio web desde ese repositorio
4. Establecer la variable `PORT` en 10000 o dejarlo automático
5. No incluir el archivo `.env` en GitHub, pero añadirlo manualmente en Render > Environment variables

### 6. Recomendaciones Futuras

- Considerar migrar a Django si el proyecto escala significativamente
- Usar PostgreSQL en producción
- Implementar Nginx + Gunicorn para servidor profesional
- Añadir tests unitarios con pytest
- Utilizar un CDN para las imágenes
- Agregar autenticación por tokens si se espera API pública