

# Documentación Preliminar – Proyecto Integrador Portfolio Web

**Alumno:** Mateo Ulla

**Curso:** 6° Año – Instituto Técnico Renault

**Materia:** Laboratorio de Programación

**Año:** 2025

## 1. Introducción

Proyecto **Portfolio Web** donde pueda mostrar mis datos, estudios, experiencias, habilidades y proyectos.

Además, el sitio incluye un **login de administrador** para poder editar mi información de forma segura, y una sección para descargar mi **CV en PDF**.

El objetivo fue hacer una web profesional, con un diseño moderno, adaptable a cualquier dispositivo, y que me sirva como **carta de presentación en el mundo laboral**, especialmente para el área de programación.

## 2. Objetivo del Proyecto

El objetivo general fue **diseñar y desarrollar una aplicación web full stack** (front end + back end + base de datos) que funcione como portfolio personal.

### Objetivos específicos:

- Implementar un **login de administrador** para editar la información del sitio.
- Guardar los datos en una **base de datos MySQL**, conectada con **Flask** mediante **PyMySQL**.
- Aplicar **Programación Orientada a Objetos (POO)** utilizando sus cuatro pilares (abstracción, encapsulamiento, herencia y polimorfismo).
- Diseñar una interfaz **adaptable (responsive)** con **Bootstrap**, que se vea bien en celular, tablet y PC.
- Incluir **animaciones sutiles y transiciones** para mejorar la experiencia visual.
- Desplegar la aplicación en **Render** y la base de datos en **Clever Cloud**.

- Cumplir con las medidas básicas de **seguridad** usando **Flask-WTF** y **hash de contraseña**.

### 3. Planificación y Organización

#### Etapas de trabajo:

1. **Investigación y diseño inicial:**  
Revisé portfolios existentes para inspirarme en el estilo visual. Elegí una **paleta oscura** y minimalista, con tonos grises y beige para mantener un look profesional.
2. **Diseño del modelo de datos:**  
Dibujé un **modelo relacional** con tablas para **personal**, **experience**, **skills** y **admin**.  
Definí las relaciones y campos necesarios para almacenar toda la info del portfolio.
3. **Desarrollo del backend:**  
Usé **Flask (Python)** como framework principal.  
Implementé los modelos con **POO**, marcando los cuatro pilares con comentarios en el código.  
Usé **PyMySQL** para conectar con la base de datos y **Werkzeug** para manejar los hash de las contraseñas.
4. **Desarrollo del frontend:**  
Creé los templates con **Jinja2** y usé **Bootstrap** para el diseño responsivo.  
Agregué una animación suave en la foto de perfil.
5. **Integración y testing:**  
Probé la app en distintos tamaños de pantalla (celu, tablet y notebook) para asegurar que todo se vea bien.  
Revisé que el login y la edición funcionen sin errores.
6. **Despliegue:**  
Configuré la base en **Clever Cloud** y la app en **Render**, usando variables de entorno para conectar ambas partes.
7. **Documentación:**  
Finalmente, escribí este documento para dejar constancia de todo el proceso de desarrollo y planificación.

### 4. Descripción Técnica del Proyecto

- **Lenguaje principal:** Python 3

- **Framework backend:** Flask
- **Base de datos:** MySQL (deploy en Clever Cloud)
- **Conector:** PyMySQL
- **Frontend:** HTML + CSS (Bootstrap 5) + Jinja2
- **Seguridad:** Flask-WTF + Hash de contraseñas con Werkzeug
- **Deploy:** Render (para la app) y Clever Cloud (para la base de datos)
- **Control de versiones:** Git y GitHub

### Estructura general:

```
portfolio_flask/
├── app.py
├── models.py
├── forms.py
├── db.py
├── config.py
├── templates/
├── static/
└── schema.sql
```

### Pilares de POO usados:

- **Encapsulamiento:** atributos privados (`_table`) en los modelos.
- **Abstracción:** métodos que ocultan detalles SQL (`all()`, `find()`, etc.).
- **Herencia:** todos los modelos heredan de `BaseModel`.
- **Polimorfismo:** método `all()` redefinido en `SkillsModel` para ordenar distinto.

## 5. Aprendizajes y Dificultades

Al principio me costó bastante entender cómo configurar las rutas del login y que los formularios de Flask-WTF funcionen bien con los templates.

Otro desafío fue hacer que la web sea totalmente **responsive** sin romper el diseño. Me ayudó mucho Bootstrap, pero tuve que ajustar cosas en CSS para que se vea bien en celular.

En cuanto al deploy, fue la parte más técnica: configurar Render y Clever Cloud con las variables de entorno fue nuevo para mí, pero al final salió todo bien.

## 6. Conclusión

Este proyecto me sirvió para unir todo lo que aprendí en la materia: programación, bases de datos, diseño web y lógica.

Además, fue una buena práctica de cómo trabajar en un proyecto real que podría mostrarle a una empresa o usar como portfolio profesional.

Aprendí a organizar mejor mi código, a trabajar con frameworks, y sobre todo a tener paciencia cuando las cosas no funcionan a la primera.

Siento que el resultado final representa bastante mi estilo y lo que quiero seguir haciendo en el futuro: **desarrollo y tecnología aplicada a los negocios**.

## 7. Próximos pasos

- Añadir un formulario de contacto para que me puedan escribir directamente.
- Mejorar la optimización para SEO y tiempos de carga.

**Córdoba, Argentina – Octubre 2025**