

Informe Técnico del Sistema de Licencias Escolares

1. Introducción

Este informe documenta el desarrollo, implementación y despliegue de un sistema web para la gestión de licencias escolares, dirigido a instituciones educativas que requieren llevar un control de solicitudes de permisos por parte de sus profesores. El sistema está desarrollado en Python usando el framework Flask, con SQLite como sistema de base de datos.

2. Objetivo

Diseñar e implementar un sistema simple, accesible y funcional que permita a los profesores:

- Registrar solicitudes de licencia
- Visualizar el estado de sus solicitudes

Y a los administradores:

- Gestionar profesores
- Aprobar o rechazar licencias

3. Tecnologías y Librerías Utilizadas

3.1 Lenguajes y Herramientas

- **Python 3.13:** Lenguaje principal del backend
- **Flask:** Framework web liviano para Python
- **Jinja2:** Motor de plantillas usado por Flask
- **SQLite:** Base de datos embebida utilizada para persistencia
- **HTML5 y CSS3:** Para la interfaz del usuario (frontend)

3.2 Librerías de Python

Las siguientes librerías deben estar instaladas en el entorno del proyecto:

Instalación global o local (recomendado: entorno virtual)

```
pip install flask  
pip install werkzeug
```

Detalles:

- **Flask:** framework principal del backend

- **Werkzeug:** para el manejo de contraseñas con funciones como `generate_password_hash` y `check_password_hash`

4. Estructura del Proyecto

```
colegio-licencias/  
├── app.py  
├── database.db  
├── templates/  
│   ├── base.html  
│   ├── login.html  
│   ├── dashboard_admin.html  
│   ├── dashboard_profesor.html  
│   ├── register_profesor.html  
│   └── solicitudes.html  
└── static/  
    ├── css/  
    └── styles.css
```

5. Instalación y Ejecución Local

5.1 Requisitos Previos

- Tener Python 3.10 o superior instalado

5.2 Crear entorno virtual (opcional pero recomendable)

```
python -m venv venv  
source venv/bin/activate # Linux/macOS  
o  
venv\Scripts\activate   # Windows
```

5.3 Instalar dependencias

```
pip install flask werkzeug
```

5.4 Ejecutar la aplicación

```
python app.py
```

Luego acceder a <http://localhost:5000> en el navegador.

6. Base de Datos

La base de datos utilizada es SQLite, y se inicializa automáticamente con las siguientes tablas:

Tabla: profesores

- id
- nombre

- carnet (valor único)
- contraseña (encriptada)
- turno
- especialidad

Tabla: licencias

- id
- profesor_id (FK)
- fecha_inicio
- fecha_fin
- motivo
- estado (por defecto 'En espera')

7. Roles y Funcionalidades

7.1 Profesor

- Iniciar sesión con carnet y contraseña
- Solicitar licencia indicando:
 - Fecha inicio y fin (validando que sean futuras)
 - Motivo
- Visualizar todas sus solicitudes y sus estados

7.2 Administrador (credenciales: `admin/admin`)

- Gestionar usuarios (ver y eliminar profesores)
- Registrar nuevos profesores
- Aprobar o rechazar solicitudes de licencia

8. Estilo y Diseño

Se utilizó un diseño minimalista con CSS personalizado. Se recomienda mejorar la interfaz usando frameworks como Bootstrap en futuras versiones.

9. Consideraciones de Seguridad

- Las contraseñas están almacenadas en formato hash usando Werkzeug
- Las sesiones están gestionadas con una clave secreta definida en `app.secret_key`
- Las rutas críticas verifican el rol del usuario antes de permitir acceso

10. Despliegue en Render.com

Pasos resumidos:

1. Subir el proyecto a un repositorio en GitHub
2. Crear una cuenta en <https://render.com>

3. Crear un nuevo servicio web (Web Service)
4. Seleccionar el repositorio
5. Configurar:
 - **Start Command:** `python app.py`
 - **Runtime:** Python 3.13
 - **Build Command:** dejar vacío
6. Render creará el entorno y alojará el sitio en un subdominio

Notas:

- Render no permite escritura directa sobre archivos como `database.db`, por lo que se recomienda usar PostgreSQL o SQLite temporal sin persistencia para pruebas.
- Para soporte de base de datos persistente, adaptar la app a PostgreSQL

11. Recomendaciones Futuras

- Permitir edición de datos del profesor
- Agregar validación más robusta en formularios
- Mejorar interfaz visual con Bootstrap o Tailwind
- Agregar notificaciones por correo o WhatsApp (usando Twilio)

12. Conclusión

Este sistema proporciona una solución funcional y ligera para gestionar licencias escolares, diferenciando claramente los roles y garantizando una experiencia sencilla y clara. Gracias a su uso de librerías mínimas, es fácil de mantener, desplegar y extender.