# Planificación de pruebas

## Alcance:

El plan de pruebas abarca la validación de la aplicación web que permite a los padres de familia consultar las horas de entrada y salida de los alumnos mediante matrícula y filtro de fechas. El sistema también permite descargar los resultados en un archivo.

## Objetivos:

* Verificar la correcta conexión de la aplicación con la API y base de datos del Conalep.
* Validar la autenticidad y exactitud de los datos mostrados.
* Confirmar la funcionalidad de filtrado por matrícula y rango de fechas.
* Asegurar la correcta exportación/descarga de los datos.
* Identificar defectos en la interfaz y experiencia de usuario.

## **Tipos de pruebas:**

* Pruebas funcionales
* Pruebas de integración
* Pruebas de regresión
* Pruebas automatizadas con Selenium

## **Recursos necesarios:**

**Hardware:** PC con Windows   
**Software:** Navegadores (Chrome, Firefox), Selenium, Python 3, API de proveedor de servicio de reconocimiento facial (Hikvision / Syscom)  
**Datos de prueba:** Matrículas de alumnos ficticios o datos anonimizados

## Roles y responsabilidades:

* **Tester**: Diseñar, implementar y ejecutar pruebas con Selenium
* **Desarrollador backend**: Soporte en API y base de datos
* **Administrador del sistema**: Proveer acceso y datos de prueba

## Criterios de entrada:

* Aplicación web desplegada en entorno de pruebas
* API funcional y accesible
* Datos de prueba cargados

## Criterios de salida:

* Todos los casos de prueba ejecutados
* Defectos críticos corregidos o documentados
* Validación del correcto funcionamiento del sistema

1.8 Entregables:

* Plan de pruebas
* Casos de prueba documentados
* Reportes de ejecución
* Informe de cierre de pruebas

# Análisis y diseño de pruebas

## identificación de casos de prueba

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Módulo | Descripción del Caso de Prueba | Resultado Esperado | Prioridad |
| CP-001 | Consulta | Búsqueda exitosa con matrícula válida y rango de fechas de inicio y final. | Se muestran los registros de entrada/salida y la opción de descarga se habilita. | Alta |
| CP-002 | Consulta | Búsqueda con matrícula válida, pero sin registros en el rango de fechas. | Se muestra un mensaje de "Por favor llene todos los campos". | Alta |
| CP-003 | Consulta | Búsqueda con matrícula inválida (formato incorrecto o inexistente). | Se muestra mensaje de error/validación para el usuario y se solicita verificar la información. | Alta |
| CP-004 | Consulta | Búsqueda con rango de fechas incorrecto (ej. fecha final < fecha inicial). | Se muestra mensaje de error/validación para el usuario y se solicita verificar la información. | Media |
| CP-005 | Descarga | Descargar datos después de una consulta exitosa. | Se genera y descarga un archivo (PDF) con los datos mostrados. en el mismo formato que en la aplicación | Alta |

## Estrategia de Pruebas

Se adoptará un enfoque de pruebas de caja negra.

* **Priorización de la funcionalidad central:** Se priorizará la automatización de la funcionalidad de consulta (filtrado y visualización) y descarga.
* **Uso de Datos de Prueba:** Se comenzará con datos de prueba positivos (escenarios exitosos) para validar la funcionalidad básica, seguido de datos de prueba negativos (errores de matrícula/fechas) para validar el manejo de excepciones.

## Definición de Datos de Prueba

Se necesitarán los siguientes tipos de datos en el entorno de pruebas:

* **Matrículas Válidas:** Al menos 3 matrículas con diferentes volúmenes de datos históricos (mucha, poca y ninguna información en un rango).
* **Matrículas Inválidas:** Matrículas con formato incorrecto (letras, símbolos), y una matrícula que no exista en el sistema.
* **Rangos de Fechas:** Rango de fechas corto (1 día), rango largo (1 mes), rango en el futuro, y rango invertido

# Implementación y Ejecución de Pruebas

## Creación de Scripts de Prueba

* **Herramienta:** Se usará **Selenium con Python** para la automatización.
* **Técnica:** Se implementarán scripts siguiendo el patrón **Page Object Model (POM)** para una mejor mantenibilidad.
* **Enfoque:** Los scripts se enfocarán en simular la interacción de un padre de familia (introducir matrícula, seleccionar fechas, hacer clic en consultar, validar resultados y hacer clic en descargar).

## Ejecución de Pruebas

* **Automatizadas:** Ejecución diaria de la *suite* de pruebas de regresión una vez que la funcionalidad esté estable.
* **Manuales:** Pruebas de usabilidad y pruebas exploratorias se realizarán manualmente en cada iteración importante.

## Documentación de Resultados

Cada ejecución de prueba (manual o automatizada) deberá registrar:

* **ID del Caso de Prueba**
* **Fecha y hora de ejecución**
* **Resultado** (Pasa/Falla)
* **Evidencia** (Capturas de pantalla para pruebas manuales; logs/reportes para automatizadas).
* **Referencia al Defecto** (si la prueba falla).