



Formando líderes para la construcción  
de un nuevo país en paz

## **PLAN DE PRUEBAS PARA SISTEMA DE MONITOREO DE ACCESO BASADO EN RECONOCIMIENTO FACIAL PARA EL CONTROL Y LA SUPERVISION PRECISA EN UN SALON DE CLASE**

Autor 1: Juan Pablo Marquez Sanchez

1004922828

Autor 2: Samuel Andres Celis Lizcano

1091964042

Autor 3: Yorman Rodolfo Rodriguez Jaimes

1005027437

Universidad de Pamplona

Facultad de Ingeniería y Arquitectura

Ingeniería de Sistemas

2024

Villa del Rosario – Norte de Santander



## Índice

1.	Introducción .....	4
1.1	Propósito .....	4
1.2	Alcance .....	4
1.3	Objetivos de Pruebas .....	4
2.	Estrategia de pruebas.....	5
2.1	Enfoque de pruebas .....	5
2.2	Recursos de Pruebas.....	5
2.3	Planificación de Pruebas .....	6
2.4	Criterios de Aceptación .....	7
3.	Casos de Prueba.....	7
3.1	Identificación de Casos de Prueba.....	7
3.2	Descripción de Casos de Pruebas .....	8
4.	Ejecución de las pruebas.....	10
4.1	Procedimientos de Ejecución .....	10
4.2	Registro de Defectos.....	11
5.	Riesgos y Contingencias .....	12
5.1	Riesgos .....	12
5.2	Estrategias de Mitigación .....	12
6.	Aprobación .....	12
6.1	Aprobación del cliente .....	12



## 7. Historial de Revisiones ..... 13



## 1. Introducción

### 1.1 Propósito

El propósito de este documento es describir la estrategia de pruebas para el SISTEMA DE MONITOREO DE ACCESO BASADO EN RECONOCIMIENTO FACIAL PARA EL CONTROL Y LA SUPERVISION PRECISA EN UN SALON DE CLASE.

Este documento busca otorgar un orden y justificación a las pruebas de se van a utilizar para validar los avances de programación que se realicen durante y después de la fase de codificación.

### 1.2 Alcance

El presente plan de pruebas busca cubrir las 3 partes del código (Front-end, Back-end y el modelo predictivo), por medio de pruebas establecidas para cada parte del desarrollo se espera lograr tener exactitud en la navegación e implementación de todas las partes, por último, realizar una revisión del funcionamiento de la aplicación con todos sus componentes.

### 1.3 Objetivos de Pruebas

- Validar las funciones utilizadas para el tratamiento de los datos de los usuarios.
- Verificar la navegación entre las diferentes pantallas de la interfaz.
- Identificar porcentajes de error en la identificación de las personas en el modelo creado.
- Corregir errores previos a la integración general de las partes.



## 2. Estrategia de pruebas

### 2.1 Enfoque de pruebas

Como ya se comentó las pruebas serán divididas dependiendo del componente que se desee probar, sin embargo, todos los componentes estarán siendo evaluados con Pruebas Unitarias. Pero cada componente tendrá ciertas pruebas especiales, las cuales son:

**Front-End:** Pruebas de Interfaz de usuario, Pruebas de Integración.

**Back-End:** Pruebas de Integración, Pruebas Funcionales.

**Modelo Predictivo:** Prueba de la precisión del modelo, Prueba de Carga.

### 2.2 Recursos de Pruebas

- **Personal de Pruebas:** Las pruebas serán realizadas por cada desarrollador en su área, en este caso por el encargado del Front-End (Samuel Celis), el encargado del Back-End (Yorman Rodriguez) y el encargado del modelo predictivo (Juan Marquez).
- **Ambiente de Pruebas:** Todas las pruebas serán realizadas en el editor de código Visual Studio Code, utilizando las librerías de Python especializadas en Pruebas Unitarias. En caso de las pruebas de Interfaz serán realizadas ejecutando la interfaz para verificar y comprobar el comportamiento y la ubicación de los elementos.



- **Herramientas de Pruebas:** Se utilizará la librería unittest de Python, los computadores de los desarrolladores y el editor de código Visual Studio Code.

## 2.3 Planificación de Pruebas

Front-End		
Pruebas	Fecha de Inicio	Fecha de Finalización
Pruebas Unitarias ( <b>Unit Test</b> )	19 de octubre	25 de octubre
Pruebas de Interfaz de Usuario ( <b>UI Test</b> )	26 de octubre	1 de noviembre
Pruebas de Integración	2 de noviembre	8 de noviembre

Back-End		
Pruebas	Fecha de Inicio	Fecha de Finalización
Pruebas Unitarias ( <b>Unit Test</b> )	9 de noviembre	15 de noviembre
Pruebas de Integración	16 de noviembre	22 de noviembre
Pruebas Funcionales/End-to-End ( <b>E2E</b> )	23 de noviembre	29 de noviembre

Modelo Predictivo		
Pruebas	Fecha de Inicio	Fecha de Finalización
Prueba de la precisión del modelo	30 de octubre	5 de noviembre
Prueba de la carga del modelo	6 de noviembre	8 de noviembre



Prueba del preprocesamiento de imágenes	9 de noviembre	11 de noviembre
---	----------------	-----------------

## 2.4 Criterios de Aceptación

Para considerar de forma exitosa cada prueba realizada se deben obtener los resultados esperados, por cada uno de los desarrolladores.

**Front-End:** Carga de los elementos en los lugares esperados, movimiento entre las pantallas de la aplicación, funcionamiento de los elementos individualmente

**Back-End:** Correcta gestión de datos (almacenamiento, recuperación y actualización), respuestas consistentes a las solicitudes del front-end, manejo adecuado de errores, autenticación y autorización seguras, y un rendimiento eficiente bajo carga.

**Modelo Predictivo:** precisión adecuada en las predicciones sobre los datos de prueba (superior al 85%), correcta carga y ejecución del modelo sin errores, procesamiento eficiente y adecuado de las imágenes de entrada (incluyendo redimensionamiento y normalización), y tiempos de predicción dentro de límites aceptables para su uso en tiempo real o cercano al tiempo real.

## 3. Casos de Prueba

### 3.1 Identificación de Casos de Prueba

- Pantallas de la Interfaz



- Función para guardar los datos de los usuarios (Usuario, Contraseña, Biometría, etc...).
- Modelo de predicción de los estudiantes
- Integración de todas las pantallas de interfaz
- Integración de la interfaz con el modelo de predicción
- Integración de la interfaz con el Back-End
- Integración de la Interfaz, Back-End y modelo de predicción.

### 3.2 Descripción de Casos de Pruebas

- **Pantallas de la interfaz:** Durante este caso de prueba se ejecutará individualmente cada una de las pantallas de la interfaz para probar la ubicación de los elementos que la componen. Con estas pruebas se esperan resultados como: ubicaciones, colores, tipo de letra y tamaños deseados que fueron establecidos en el modelo.
- **Función para guardar los datos de los usuarios:** En este caso de prueba se debe verificar que los datos proporcionados se almacenen correctamente en la base de datos. Los casos de prueba incluirán validar que los datos obligatorios (como usuario, documento de identidad y contraseña) se guarden sin errores cuando se ingresan correctamente, que se manejen apropiadamente errores como campos vacíos o formatos inválidos (por ejemplo, contraseña incorrecta)





- **Modelo de predicción de los estudiantes:** Para este caso de pruebas se debe verificar la precisión del modelo al reconocer rostros en diversas condiciones (iluminación, ángulos, expresiones) y manejar correctamente tanto imágenes válidas como fallos en la detección de rostros no registrados. Se probará el manejo eficiente de imágenes de alta y baja calidad, el tiempo de respuesta del modelo para predicciones en tiempo real, y la capacidad del modelo para diferenciar rostros similares sin generar falsos positivos.
- **Integración de todas las pantallas de la interfaz:** En este caso se busca ejecutar el programa con todas las pantallas listas aprobadas con anterioridad, en esta prueba se busca confirmar el correcto cargue de los elementos de acuerdo a la navegación del usuario por la aplicación. Se espera como resultado una navegación limpia y sin complicaciones.
- **Integración de la interfaz con el modelo de predicción:** Lo que la prueba busca es observar como se comporta la carga de cámara del modelo en las pantallas de la interfaz definidas para esta. Como resultado se espera que esta aparezca de manera correcta en su ubicación establecida en el modelo de la aplicación.
- **Integración de la interfaz con el Back-End:** En este caso se ejecutará la interfaz con las validaciones de usuario y contraseña, dependiendo de esto la información cargará en sus respectivas ubicaciones, llenando las tablas de los cursos. Como resultado se espera una navegación limpia,

*"Formando líderes para la construcción de un nuevo país en paz"*



con información relacionada al usuario que se registre, validaciones de la información que se consulta y guarda.

- **Integración de la Interfaz, Back-End y modelo de predicción:** En este caso final por denominarlo así, se busca realizar una prueba completa de la aplicación. Como resultado se espera una navegación limpia, con validaciones e información específica de cada usuario, permitiendo utilizar el sistema de reconocimiento facial para hacer los registros de los estudiantes.

## 4. Ejecución de las pruebas

### 4.1 Procedimientos de Ejecución

- **Pantallas de la Interfaz:** Prueba a cargo de Samuel Celis, los resultados se registrarán en un formato de registro de pruebas en cual se escribirán todos los resultados obtenidos.
- **Función para guardar los datos de los usuarios:** Prueba a cargo de Yorman Rodriguez, los resultados serán registrados en un formato de pruebas en la cual se escribirán todos los resultados obtenidos.
- **Modelo de predicción de los estudiantes:** Prueba a cargo de Juan Marquez, los resultados serán registrados en un formato de pruebas en la cual se escribirán todos los resultados obtenidos.



- **Integración de todas las pantallas de interfaz:** Prueba a cargo de Samuel Celis, los resultados se registrarán en un formato de registro de pruebas en la cual se escribirán todos los resultados obtenidos.
- **Integración de la interfaz con el modelo de predicción:** Prueba a cargo de Samuel Celis y Juan Marquez, los resultados serán registrados en un formato de pruebas en la cual se escribirán todos los resultados obtenidos.
- **Integración de la interfaz con el Back-End:** Prueba a cargo de Samuel Celis y Yorman Rodriguez, los resultados serán registrados en un formato de pruebas en la cual se escribirán todos los resultados obtenidos.
- **Integración de la Interfaz, Back-End y modelo de predicción:** Prueba a cargo del equipo completo, los resultados serán registrados en un formato de pruebas en la cual se escribirán todos los resultados obtenidos.

## 4.2 Registro de Defectos

A medida que se realice una prueba se van a ir registrando los defectos y errores obtenidos con las pruebas, estos serán anotados en una hoja de errores y bugs para su pronta corrección. El principal criterio será, ¿Realiza la operación exacta esperada? Si esta pregunta no tiene una respuesta positiva concreta, se considerará como una prueba fallida y se ajustará el código hasta obtener el resultado esperado.



## 5. Riesgos y Contingencias

### 5.1 Riesgos

- Incumplimiento en el calendario del proyecto
- No claridad en el objetivo de la prueba
- Dudas con el modelo definido inicialmente de la interfaz.

### 5.2 Estrategias de Mitigación

- Manejar los tiempos de las actividades estableciendo horarios de trabajo diarios para llevar una concordancia con el calendario establecido.
- Leer e informarse acerca de los objetivos de las pruebas, tener claro que se quiere obtener como resultado.
- Establecer un modelo previamente analizado y aprobado para evitar cambios significativos en el desarrollo que puedan afectar la relevancia de algunas pruebas.

## 6. Aprobación

### 6.1 Aprobación del cliente

Para obtener la aprobación de los usuarios se realizarán pruebas de Usuarios, tanto a docentes como personas particulares, se aplicarán a personas expertas y personas no familiarizadas con la tecnología.



## 7. Historial de Revisiones

Versión	Descripción	Autor	Fecha
1.0	Creación del documento	Samuel Celis	11 – 10 - 2024