**I.E.S Fuengirola Nº1**

**DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**PROYECTO INTEGRADO**

**Sistema de Gestión Académica Privada [SGAP]**

**Departamento de Informática**

**Manual Técnico**

**Autor: Johan Sebastian Rangel González**

**ÍNDICE DE CONTENIDOS**

[**1 - Sobre éste proyecto**](#_gjdgxs)

[1.1 - Control de versiones](#_30j0zll)

[1.2 - Licencia de uso](#_1fob9te)

[**2 - Análisis del problema**](#_3znysh7)

[2.1 - Introducción al problema](#_2et92p0)

[2.2 - Antecedentes](#_tyjcwt)

[2.3 - Objetivos](#_3dy6vkm)

[2.4 - Requisitos](#_1t3h5sf)

[2.4.1 - Funcionales](#_4d34og8)

[2.4.2 - No funcionales](#_2s8eyo1)

[2.5 - Recursos](#_17dp8vu)

[2.5.1 - Software](#_3rdcrjn)

[2.5.2 - Hardware](#_26in1rg)

[**3 - Diseño de la solución software**](#_lnxbz9)

[3.1 - Modelados](#_35nkun2)

[3.1.1 - Casos de uso](#_1ksv4uv)

[3.1.2 - [ Interacción ]](#_44sinio)

[3.1.3 - [ Estado ]](#_44sinio)

[3.1.4 - [ Actividad ]](#_44sinio)

[3.2 - Prototipado gráfico](#_3j2qqm3)

[3.2.1 - [ Escritorio ]](#_1y810tw)

[3.2.2 - [ Tablets / Smartphones ]](#_1y810tw)

[3.3 - Base de datos](#_2xcytpi)

[3.3.1 - Diseño Conceptual (ER)](#_1ci93xb)

[3.3.2 - Diseño lógico (tablas normalizadas)](#_1ci93xb)

[**4 - Implementación**](#_qsh70q)

[4.1 - Codificación](#_3as4poj)

[4.1.1 - [ Usabilidad ]](#_1pxezwc)

[4.1.2 - Backend](#_49x2ik5)

[4.1.3 - Frontend](#_2p2csry)

[4.2 - [ Pruebas ]](#_147n2zr)

[**5 - Documentación**](#_23ckvvd)

[5.1 - Empaquetado / Distribución](#_ihv636)

[5.2 - Instalación](#_32hioqz)

[5.3 - Manual de Usuario / Referencia](#_1hmsyys)

[**6 - Conclusiones**](#_2grqrue)

[**7 - Bibliografía**](#_3fwokq0)

# 

# 1 - Sobre éste proyecto

## 1.1 - Control de versiones

Se empleará **Git** como sistema de control de versiones, alojando el repositorio en GitHub para garantizar la gestión de cambios y colaboraciones.

## 1.2 - Licencia de uso

El proyecto estará bajo una licencia MIT, permitiendo su uso, modificación y distribución siempre que se incluya el aviso de derechos correspondiente.

# 2 - Análisis del problema

## 2.1 - Introducción al problema

Las academias privadas enfrentan dificultades para gestionar la comunicación entre docentes, alumnos y padres. Esto genera problemas en la organización de tutorías, el seguimiento del progreso académico y la asignación de tareas.

## 2.2 - Antecedentes

Herramientas existentes como iPasen se enfocan en instituciones públicas, dejando fuera a academias privadas con necesidades específicas. Este proyecto busca cubrir esta brecha.

## 2.3 - Objetivos

* Crear una aplicación que centralice la comunicación y gestión académica.
* Ofrecer funcionalidades personalizadas para academias privadas.

## 

## 

## 

## 2.4 - Requisitos

### 

### 2.4.1 - Funcionales

1. Registro y autenticación de usuarios con roles (docente, alumno, padre).
2. Solicitud y gestión de tutorías.
3. Registro de calificaciones y asistencia.
4. Asignación y seguimiento de tareas.
5. Notificaciones automáticas.

### 2.4.2 - No funcionales

1. Interfaz intuitiva y responsiva.
2. Tiempo de respuesta inferior a 2 segundos para consultas comunes.
3. Seguridad en el almacenamiento y transmisión de datos.

## 2.5 - Recursos

### 2.5.1 - Software

* Android Studio para desarrollo móvil.
* Flask para la API REST.
* MySQL para la base de datos.

### 2.5.2 - Hardware

* Ordenador con al menos 8 GB de RAM.
* Dispositivo Android para pruebas (opcional, emulador disponible).

# 

# 

# 3 - Diseño de la solución software

## 

## 3.1 - Modelados

### 

### 3.1.1 - Casos de uso

**Actor:** Docente  
**Caso de uso:** Asignar tareas a los alumnos y gestionar su estado.

### 3.1.2 - [ Interacción ]

Representación básica del flujo de interacción entre los usuarios y el sistema.

### 3.1.3 - [ Estado ]

Definición de los estados del sistema, como "tarea pendiente", "tutoría aceptada".

### 3.1.4 - [ Actividad ]

Diagramas de las principales actividades del sistema.

## 3.2 - Prototipado gráfico

### 3.2.1 - [ Escritorio ]

No aplica en este proyecto.

### 3.2.2 - [ Tablets / Smartphones ]

Diseño de las pantallas principales:

1. Login.
2. Dashboard.
3. Calendario.
4. Vista de tareas.

## 

## 

## 3.3 - Base de datos

### 3.3.1 - Diseño Conceptual (ER)

El diseño conceptual de la base de datos incluye las siguientes entidades principales:

* **Usuarios**: Representa a docentes, alumnos y padres. Incluye atributos como ID, nombre, correo electrónico, contraseña y rol.
* **Tutorías**: Contiene información sobre las reuniones entre docentes y padres. Atributos: ID, fecha, hora, motivo, estado.
* **Tareas**: Define las actividades asignadas a los alumnos. Atributos: ID, descripción, fecha de entrega, estado.
* **Calificaciones**: Registro del desempeño académico. Atributos: ID, alumno, asignatura, nota.
* **Asistencias**: Detalla la presencia de los alumnos. Atributos: ID, alumno, fecha, estado (presente/ausente).

Relaciones clave:

* Un usuario puede tener múltiples tutorías.
* Un alumno puede tener múltiples calificaciones y asistencias.
* Una tarea está asociada a uno o más alumnos.

### 3.3.2 - Diseño lógico (tablas normalizadas)

A partir del diseño conceptual, se definen las tablas normalizadas:

* **Usuarios** (ID\_usuario, nombre, correo, contraseña, rol).
* **Tutorías** (ID\_tutoría, ID\_docente, ID\_padre, fecha, hora, motivo, estado).
* **Tareas** (ID\_tarea, descripción, fecha\_entrega, estado).
* **Asignaciones** (ID\_asignación, ID\_tarea, ID\_alumno).
* **Calificaciones** (ID\_calificación, ID\_alumno, asignatura, nota).
* **Asistencias** (ID\_asistencia, ID\_alumno, fecha, estado).

Claves primarias y foráneas:

* Las claves primarias son los campos "ID\_\*" de cada tabla.
* Claves foráneas, como "ID\_docente" y "ID\_padre" en la tabla Tutorías, están relacionadas con la tabla Usuarios.
* "ID\_tarea" en la tabla Asignaciones está relacionada con la tabla Tareas.

# 4 - Implementación

## 4.1 - Codificación

### 4.1.1 - [ Usabilidad ]

Implementación de interfaces intuitivas para usuarios.

### 4.1.2 - Backend

Desarrollo de API REST con Flask para gestionar datos y lógica del servidor

### 4.1.3 - Frontend

Aplicación móvil desarrollada en Android Studio.

## 4.2 - [ Pruebas ]

Ejecución de pruebas unitarias y funcionales para garantizar la calidad.

# 

# 

# 

# 

# 

# 5 - Documentación

## 5.1 - Empaquetado / Distribución

El empaquetado se realiza mediante la generación de un archivo APK desde Android Studio. Este APK podrá ser distribuido directamente a los usuarios o a través de plataformas de distribución como Google Play Store.

Pasos:

1. Verificar que la configuración de la aplicación esté finalizada.
2. Acceder a Android Studio y seleccionar la opción "Build" > "Generate Signed APK".
3. Seguir las indicaciones para generar el archivo firmado..

## 5.2 - Instalación

**Requisitos previos:**

* Dispositivo Android con versión mínima 8.0.
* Habilitar "Fuentes desconocidas" en los ajustes del dispositivo.

**Instrucciones:**

1. Descargar el archivo APK en el dispositivo.
2. Abrir el archivo descargado.
3. Seguir las indicaciones en pantalla para completar la instalación.

## 5.3 - Manual de Usuario / Referencia

**Inicio de sesión:**

1. Abrir la aplicación.
2. Introducir las credenciales proporcionadas (usuario y contraseña).

**Navegación por la aplicación:**

* **Dashboard:** Acceso rápido a tareas, tutorías y notificaciones.
* **Calendario:** Visualización de tutorías programadas.
* **Tareas:** Gestión de las tareas asignadas.

**Gestión de tutorías:**

1. Acceder al calendario.
2. Seleccionar una fecha disponible.
3. Solicitar la tutoría indicando el motivo.

# 6 - Conclusiones

El desarrollo de este proyecto ha permitido cubrir una necesidad importante dentro del ámbito académico privado, proporcionando una herramienta eficaz para la comunicación y gestión. La aplicación combina usabilidad, funcionalidad y tecnología moderna para atender las demandas de usuarios en roles diversos.

Entre las fortalezas del proyecto se encuentra:

1. La adaptabilidad a distintos tamaños de academias.
2. La centralización de funcionalidades clave en un solo sistema.

Sin embargo, existen áreas de mejora para versiones futuras, como:

1. Implementar soporte multilingüe.
2. Integración con plataformas externas como Google Calendar.
3. Desarrollo de una versión web complementaria.

# 

# 7 - Bibliografía

1. Documentación oficial de Android Developers.
2. Flask Documentation: https://flask.palletsprojects.com.
3. PythonAnywhere Documentation: https://help.pythonanywhere.com.