

DOCUMENTACIÓN EQUIPO 5 – SIC

Planeación de la solución y su diseño:

El proyecto propone una solución integral para gestionar eficientemente la asistencia de profesores y la afluencia de estudiantes en diversos entornos, inicialmente en salones de clases y con la perspectiva de expansión hacia otros contextos, como entornos empresariales. La solución se basa en la implementación de tecnologías avanzadas, incluyendo sensores de temperatura, cámaras, lector NFC y reconocimiento facial.

Para la implementación, se utilizarán herramientas sólidas como Python junto con FastAPI, permitiendo una flexibilidad inicial y la posibilidad de migrar a lenguajes más robustos, como C#, para optimizar el rendimiento a medida que la solución escala.

La arquitectura de la solución contempla el uso de servicios en la nube, específicamente Azure, para alojar la API central. La base de datos, en este caso MongoDB, también se desplegará en la nube, garantizando un acceso eficiente y seguro a los datos.

Además, la experiencia de usuario se potenciará mediante la aplicación Flutter, diseñada para funcionar tanto en dispositivos móviles como en versiones web. Esta aplicación permitirá realizar registros de entrada y salida, calcular el uso promedio por salón, gestionar los datos del profesor y acceder al calendario de materias de manera intuitiva y eficaz.

Puntos clave a pensar de la solución:

Datos que Podemos Obtener:

- Registro de asistencia de profesores y estudiantes.
- Afluencia o cantidad de asistentes por salón.
- Datos de temperatura ambiente.
- Información biométrica a través del reconocimiento facial.
- Patrones de asistencia a lo largo del tiempo.
- Datos relacionados con el uso de la plataforma Flutter, como registros de entrada y salida, asistencia promedio y datos del profesor.

Información y Significado de los Datos:

- Identificación de patrones de asistencia y ausencia.
- Análisis de la eficiencia del proceso educativo en función de la asistencia.
- Evaluación de la ocupación de salones y su optimización.
- Monitoreo de la temperatura ambiente y su variación.
- Creación de perfiles de asistentes a eventos específicos.
- Evaluación del rendimiento y la carga de trabajo de los profesores.

Necesidad de la Nube:

Sí, la nube es esencial para alojar la API (Azure) y la base de datos (MongoDB), permitiendo el acceso eficiente, escalabilidad y seguridad de los datos.

Monetización de Datos:**Potenciales Usuarios:**

- Instituciones educativas.
- Empresas para el control de asistencia de empleados.

Formas de Monetización:

- Venta de licencias de uso de la plataforma.
- Suscripciones mensuales para el acceso y mantenimiento.
- Ofrecer servicios premium con características avanzadas.

Usuarios de Datos:

- Instituciones educativas para mejorar la gestión académica.
- Empresas para el control de asistencia de empleados.

Forma de Cobro Ideal:

Una combinación de modelos de negocio, como un pago inicial por la implementación y una tarifa recurrente mensual o anual por servicios y actualizaciones continuas.

Comprador Potencial:**Instituciones Educativas:**

Escuelas primarias, secundarias y universidades.

Empresas:

Para el control de asistencia y gestión de empleados.

Prototipado para la Solución de Gestión de Asistencia:**Registro de Asistencia:****Simulación de Reconocimiento Facial:**

Emplear un módulo de cámara (puede ser una cámara web) para simular el reconocimiento facial.

Utilizar un software de procesamiento de imágenes como deepface para detectar rostros y simular la verificación biométrica.

Uso del Lector NFC:

Emular la función de un lector NFC mediante dispositivos de desarrollo o tarjetas de prueba NFC y con esto activar el circuito.

Implementar un sistema de lectura de tarjetas simulado para representar la interacción con dispositivos NFC.

Sensores y Herramientas:

Simulación de Sensores de Temperatura:

Utilizar sensores de temperatura conectados a la raspberry Pi.

Crear patrones de datos simulados que reflejan variaciones de temperatura.

Integración con Python y Flask:

Crear funciones simuladas en Python para representar la interacción con sensores y la lógica de negocio.

Utilizar Flask para simular la creación de una API y su integración con la interfaz de usuario.

Interfaz de Usuario con Flutter:

Prototipado Visual:

Utilizar herramientas de diseño como Figma para crear prototipos visuales de la interfaz de usuario.

Integrar elementos de Flutter para simular la navegación y la interactividad de la aplicación.

Implementación de la Lógica:

Utilizar código Flutter para implementar la lógica detrás de la interfaz de usuario simulada.

Simular la comunicación con la API y la visualización de datos de asistencia.

Hosteo en la Nube con Azure:

Simulación del Servicio en la Nube:

Utilizar entornos de desarrollo local para simular el hosteo en la nube.

Implementar lógica simulada para responder a las solicitudes como si estuviera alojada en Azure.

Conexión con MongoDB:

Emplear bases de datos locales o en la nube simuladas para representar la conexión con MongoDB.

Simular las operaciones de lectura y escritura en la base de datos.

Simulación de Datos y Valor Agregado:

Generación de Datos Simulados:

Crear scripts que generan datos simulados para mostrar el valor agregado, como patrones de asistencia y estadísticas.

Utilizar archivos de texto o bases de datos temporales para almacenar y recuperar datos simulados.

Nota:

Es fundamental documentar de manera clara que el prototipo es una representación simulada y no refleja el funcionamiento real de los sensores y sistemas. El objetivo es demostrar la viabilidad y la lógica detrás de la solución antes de invertir en implementaciones más complejas y costosas.

UI/UX

Aspecto de la Interfaz con el Usuario:

- El diseño debe ser moderno, limpio y profesional.
- Utilizar una paleta de colores que inspire confianza y sea agradable a la vista.
- Incorporar elementos visuales coherentes con el entorno educativo y empresarial.

Intuitividad de la Interfaz:

- Incluir una navegación sencilla y lógica para facilitar el uso sin necesidad de instrucciones detalladas.
- Utilizar iconos claros y etiquetas descriptivas para indicar la función de cada elemento.
- Mantener consistencia en la disposición de los elementos en todas las secciones.
- Proporcionar retroalimentación visual instantánea sobre las acciones del usuario.

Elementos básicos

- Pantalla inicial de login: Aunque la forma de login puede variar según los requerimientos de la empresa, proponemos iniciar sesión con número de empleado o correo institucional y una contraseña

- Pantalla de registro: De igual manera que la de login puede variar según los requerimientos de la empresa, para registrar se propone, nombre del profesor, número de empleado/correo institucional, imagen de la cara, contraseña.
- Pantalla de home: Para profesores incluye la clase actual según la hora si completó la asistencia de entrada y salida así como de la hora que lo realizó teniendo en cuenta que lo puede hacer 5 minutos antes la entrada y 5 minutos máximo después la salida y para administradores el porcentaje de asistencia de los profesores semanalmente.
- Pantalla de horarios: En esta pantalla cada profesor puede ver las materias que tiene y a qué hora, así como un apartado de sus asistencias o faltas, mostrando un registro, para administradores se muestra la tabla de asistencias según el nombre del profesor.
- Pantalla de configuración: Se puede cambiar la contraseña, tema de la app y cerrar la sesión.
- Pantalla de afluencia en cada salón: Esta pantalla está solo disponible a administradores, donde se puede observar una gráfica de pastel con el porcentaje de uso de todas las aulas.

Acceso del Usuario:

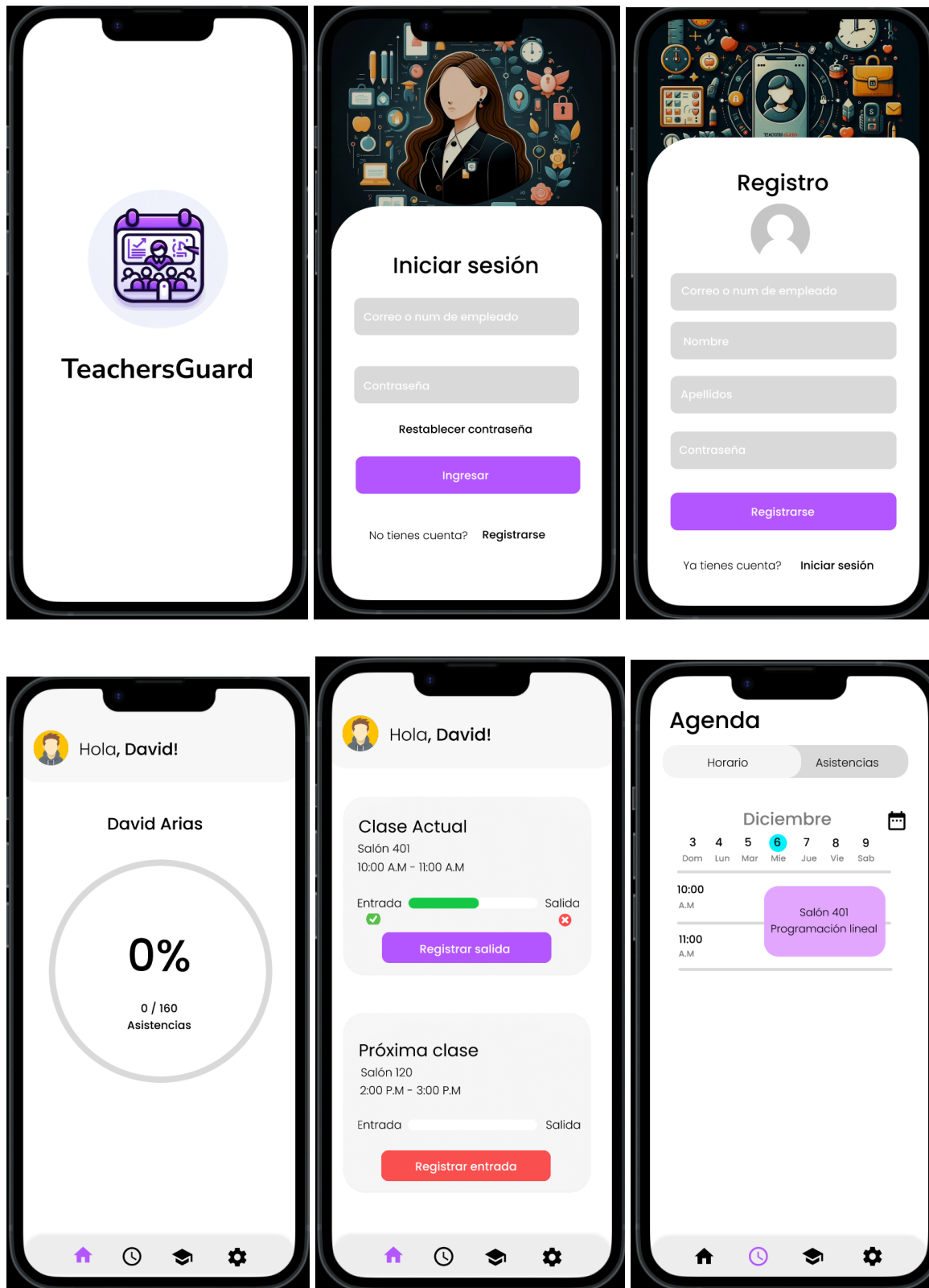
- Permitir el acceso tanto desde dispositivos móviles como desde computadoras.
- Implementar un diseño responsive que se adapte a diferentes tamaños de pantalla.

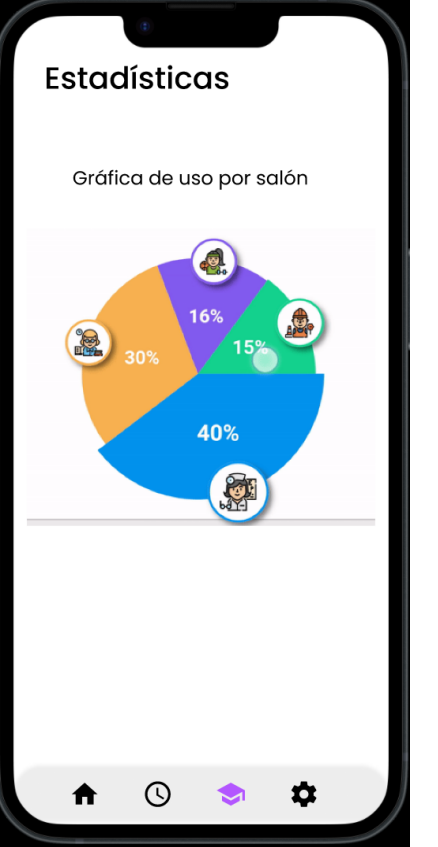
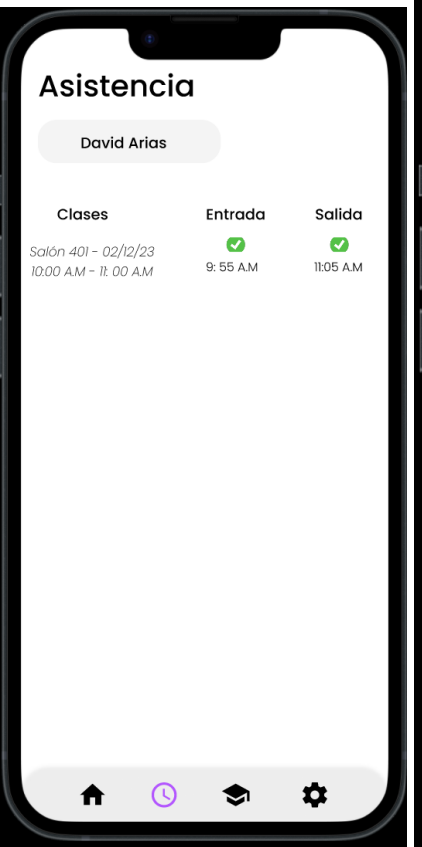
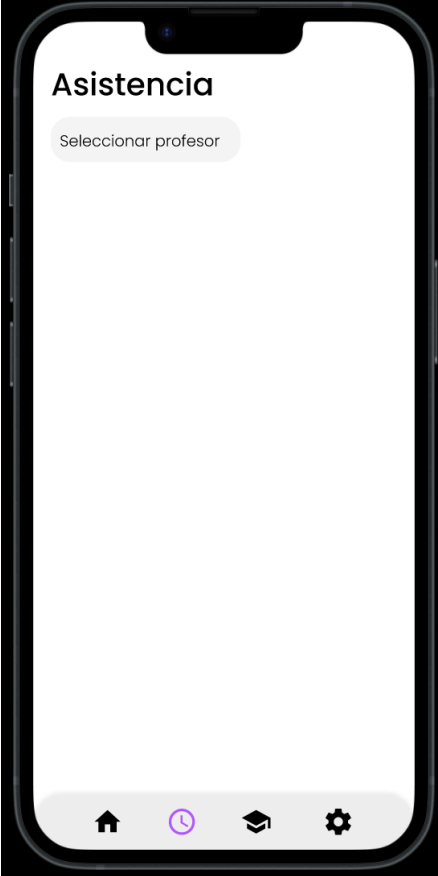
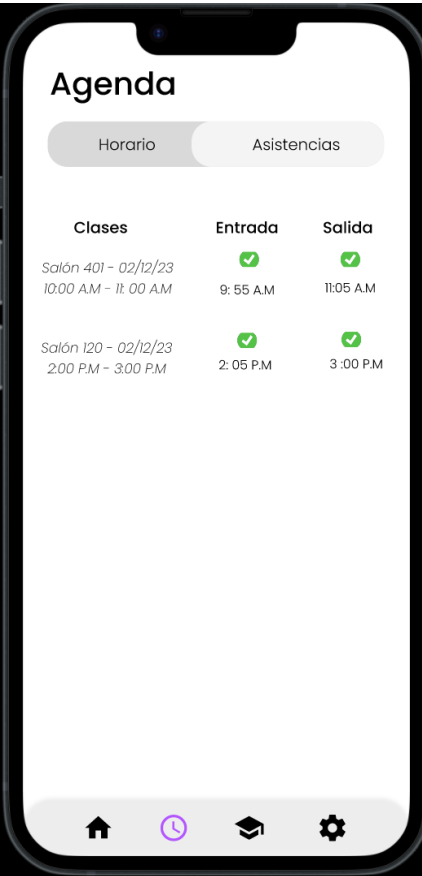
Forma de Visualización:

- Desarrollar una aplicación Flutter para dispositivos móviles, garantizando una experiencia intuitiva y móvil.
- Considerar también una versión web para permitir el acceso desde computadoras, proporcionando versatilidad y accesibilidad.

Mockup:

El siguiente mockup está diseñado para dispositivos móviles pero en esencia se verá igual para web.





Historia de Caso de Uso: Mejora en la Gestión de Asistencia Educativa y Empresarial con Nuestra Solución

1) Contexto de las Personas Involucradas:

En un entorno educativo, directores de instituciones, profesores y personal administrativo enfrentan desafíos para gestionar eficientemente la asistencia de profesores y estudiantes. En el ámbito empresarial, los departamentos de recursos humanos luchan por un control preciso de la asistencia de los empleados.

2) Problema y Requerimientos para la Solución:

El problema radica en la falta de un sistema centralizado y tecnológico para el registro de asistencia. Se requiere una solución que permita registrar la asistencia de manera automática, precisa y que sea escalable para adaptarse a diferentes contextos y necesidades.

3) Razones por las que no se ha Resuelto:

Las soluciones actuales pueden ser manuales, propensas a errores y carecer de integración con tecnologías modernas. La falta de una plataforma integral que utilice sensores avanzados y tecnologías de la nube ha limitado la eficiencia en la gestión de la asistencia.

4) Desarrollo del Prototipo y Valor Agregado:

La creación del prototipo se originó al reconocer la necesidad de una solución integral. Se incorporan sensores de temperatura, cámaras, lector NFC y reconocimiento facial, junto con tecnologías como Python y Flutter para proporcionar una interfaz de usuario eficiente. La elección de la nube, utilizando Azure, garantiza escalabilidad y seguridad.

La solución no solo aborda la asistencia, sino que también ofrece un valor adicional al proporcionar datos analíticos, como patrones de asistencia, carga de trabajo de los profesores y eficiencia en el uso de salones. Además, la versatilidad de la aplicación Flutter permite el acceso desde dispositivos móviles y computadoras, mejorando la accesibilidad y la experiencia del usuario.

5) Situaciones del Usuario y Resolución de Problemas:

Situación 1: Registro de Asistencia Rápido:

Antes: Proceso manual que consumía tiempo.

Ahora: Nuestra solución automatiza el registro de asistencia, ahorrando tiempo y reduciendo errores.

Situación 2: Gestión de Salones Eficiente:

Antes: Dificultad para optimizar la ocupación de salones.

Ahora: La visualización gráfica de la asistencia por salón permite una gestión eficiente del espacio.

Situación 3: Evaluación del Rendimiento Docente:

Antes: Información limitada sobre la carga de trabajo de los profesores.

Ahora: Datos detallados sobre la asistencia y carga de trabajo proporcionan herramientas para evaluar el rendimiento docente.

Situación 4: Adaptabilidad a Diferentes Contextos:

Antes: Soluciones estáticas sin capacidad de escalabilidad.

Ahora: La solución es escalable y adaptable a entornos educativos y empresariales, cumpliendo diferentes requisitos.