



FICHA TÉCNICA

SchoolMe
2900177

Contenido

RESUMEN EJECUTIVO.....	3
Objetivo del proyecto:	3
Resultados esperados	3
Factores de éxito del proyecto	4
Desafíos.....	4
Recomendaciones	4
Conclusión	5
INTRODUCCIÓN.....	5
DESCRIPCION DEL PROYECTO	5
¿Qué es SchoolMe y cuál es su propósito principal?	5
¿Por qué es relevante diseñar una plataforma como SchoolMe para instituciones educativas?.....	5
¿Qué aspectos diferencian a SchoolMe de otros sistemas educativos digitales?.....	6
Objetivos del proyecto	6
REQUISITOS Y CARACTERISTICAS DEL PROYECTO.....	6
Requisitos funcionales.....	6
Requisitos no funcionales	7
Características clave del proyecto	7
Oportunidades de crecimiento	7
Impacto esperado	8
SOLUCION PROPUESTA	8
Arquitectura de la plataforma.....	8
Backend: C# con ASP.NET Core (API RESTful)	8
Frontend: React o Angular (Web) y React Native (Móvil)	8
Base de Datos (SQL Server / PostgreSQL)	9
Desglose de componentes y funcionalidades	10
PLAN DE TRABAJO.....	11
1. Fase de análisis y planificación (1 mes):	12
2. Fase de diseño y prototipado (1 mes).....	12
3. Fase de desarrollo inicial (2 meses):.....	12
4. Fase de desarrollo y prueba (2 meses).....	13
Actividades principales:	13
5. Fase de implementación y lanzamiento (1 mes)	13

• Entregables:	14
Conclusión:	15
Objetivo:	15
1. Estructura del equipo:.....	16

RESUMEN EJECUTIVO

El presente documento describe la especificación de requerimientos para el desarrollo de **SchoolMe**, una plataforma digital orientada a facilitar la gestión académica y administrativa en instituciones educativas, especialmente colegios. Esta iniciativa surge como una respuesta a las dificultades que enfrentan muchas instituciones al depender de herramientas físicas para registrar información importante, como las agendas escolares, y a la necesidad de organizar de manera más eficiente los procesos internos.

SchoolMe se concibe como una solución digital integral que permite centralizar y estructurar procesos clave como la **gestión de personas, organización de grupos académicos, asignación de carga académica y registro diario en agendas escolares digitales**. Estas funcionalidades estarán disponibles principalmente en una **versión web** dirigida a administradores y docentes, mientras que una **versión móvil** estará enfocada en acudientes, quienes podrán visualizar la agenda escolar y mantenerse informados del proceso educativo.

Este documento establece los requerimientos funcionales y no funcionales necesarios para el desarrollo del sistema, junto con las restricciones técnicas y de diseño. Es importante aclarar que esta primera fase contempla únicamente el **diseño y construcción de la plataforma**, sin incluir su implementación o evaluación en un entorno real de uso.

Objetivo del proyecto:

Diseñar y desarrollar la plataforma digital SchoolMe, orientada a digitalizar y optimizar los procesos administrativos y académicos de instituciones educativas, facilitando la gestión integral de inscripciones, matrículas, personal institucional y agenda escolar mediante una solución personalizable que fortalezca la identidad visual y operativa de cada colegio.

Resultados esperados

- **Diseño completo de una plataforma digital multiplataforma**, orientada a la gestión académica y administrativa de instituciones educativas, con una versión web para el personal institucional y una versión móvil para estudiantes y acudientes.

- **Construcción de los módulos funcionales principales** definidos en el alcance del proyecto, incluyendo:
 - Gestión de usuarios (administradores, docentes, estudiantes, acudientes).
 - Creación y asignación de grupos académicos.
 - Asignación de carga académica (materias, docentes, horarios).
 - Registro y visualización de la agenda escolar digital por grado
- **Desarrollo de una interfaz de usuario (UI) intuitiva, accesible y adaptada a cada perfil de usuario**, basada en principios de usabilidad, claridad visual y navegación coherente, compatible con navegadores modernos y dispositivos móviles.
- **Documentación técnica del sistema desarrollado**, incluyendo:
 - Especificación de requerimientos funcionales y no funcionales.
 - Diagramas de arquitectura general del sistema.
 - Recomendaciones para una futura implementación.

Factores de éxito del proyecto

El éxito de SchoolMe dependerá de varios factores clave:

1. **Personalización y flexibilidad:** La plataforma debe adaptarse a las necesidades específicas de cada institución.
2. **Eficiencia operativa:** Debe optimizar la gestión escolar, reduciendo la carga administrativa y mejorando la organización de la información.

Desafíos

El desarrollo de SchoolMe enfrentará retos inherentes a la implementación de soluciones innovadoras en la educación:

- **Interfaz intuitiva y accesible:** Garantizar una experiencia de usuario fluida y eficiente para todos los actores de la comunidad educativa.
- **Competencia en el mercado:** Diferenciarse de otras plataformas de gestión escolar ofreciendo un mayor valor agregado y adaptabilidad.
- **Infraestructura escalable:** Asegurar que el sistema pueda soportar el crecimiento en usuarios sin comprometer la calidad del servicio.

Recomendaciones

Para abordar estos desafíos y garantizar el éxito del proyecto, se proponen las siguientes recomendaciones:

- Enfoque en la usabilidad y diseño centrado en el usuario, priorizando la facilidad

de uso para todos los perfiles de usuarios.

- Pruebas continuas y retroalimentación del usuario, asegurando que el sistema evoluciona según las necesidades reales de los colegios.

Conclusión

El desarrollo de la plataforma **SchoolMe** representa una solución estructurada y adaptable para digitalizar procesos clave en instituciones educativas, como la gestión de personas y el uso de agendas escolares digitales. Este documento define los requerimientos y lineamientos necesarios para su construcción, delimitando un alcance realista centrado en el diseño y desarrollo. Con una arquitectura flexible y una interfaz personalizable, SchoolMe se proyecta como una herramienta base sólida para futuras implementaciones en entornos escolares.

INTRODUCCIÓN

Este documento describe la especificación de requerimientos para el desarrollo de la plataforma digital SchoolMe, la cual está orientada a apoyar la organización de procesos administrativos y académicos en instituciones educativas. La propuesta se centra en estructurar funcionalidades como la gestión de personas, la asignación de carga académica y el registro de información escolar a través de una agenda digital.

La plataforma contará con dos versiones diferenciadas: una versión web, diseñada para el uso de personal institucional como administradores y docentes, y una versión móvil, dirigida a acudientes, con funcionalidades específicas de consulta.

SchoolMe está pensada como una solución tipo plantilla, permitiendo su personalización visual para adaptarse a la identidad institucional de cada colegio. Este documento delimita el desarrollo del sistema en una primera fase, enfocada exclusivamente en su diseño y construcción, sin contemplar procesos de implementación o validación en entornos reales.

DESCRIPCION DEL PROYECTO

¿Qué es SchoolMe y cuál es su propósito principal?

SchoolMe es una propuesta de plataforma digital cuyo propósito principal es facilitar la organización de procesos académicos y administrativos en colegios, mediante el diseño de una solución centralizada y personalizable.

¿Por qué es relevante diseñar una plataforma como SchoolMe para instituciones educativas?

Es importante **diseñar una plataforma como SchoolMe** porque permite estructurar una

solución digital adaptable a las necesidades de los colegios, orientada a organizar procesos como la gestión de personal y agenda escolar, todo desde un entorno centralizado y personalizable.

¿Qué aspectos diferencian a SchoolMe de otros sistemas educativos digitales?

SchoolMe se proyecta como una solución diferenciada por su capacidad de personalización visual para cada institución, su estructura pensada para permitir una administración autónoma de los datos, y su enfoque en ofrecer herramientas que faciliten el seguimiento académico a través del registro y organización de información clave, como la agenda escolar.

Objetivos del proyecto

- **Diseñar un módulo de gestión de personas** que centralice la información de los distintos actores educativos (administradores, docentes, estudiantes, padres y acudientes), permitiendo su fácil administración y seguimiento.
- **Implementar funcionalidades para la creación y gestión de grupos académicos**, así como la asignación de directores de grupo y docentes, asegurando una organización interna clara y estructurada.
- **Construir un sistema de carga académica flexible**, que permita asignar materias, docentes, horarios y grupos, optimizando la planificación del calendario académico.
- **Diseñar una agenda escolar digital personalizable por grado**, que facilite el registro diario de información académica y comportamental de los estudiantes, mediante preguntas predeterminadas y respuestas estructuradas.
- **Desarrollar una versión móvil de la plataforma orientada a acudientes**, que permita consultar el desempeño académico y la agenda escolar diaria, fomentando una participación activa de la familia en el proceso educativo.
- **Garantizar la personalización completa de la plataforma**, permitiendo que cada institución educativa adapte la solución a su identidad visual y marca institucional sin rastros del sistema base.
- **Optimizar la accesibilidad y usabilidad de la plataforma** tanto en su versión web como móvil, asegurando una experiencia de usuario amigable, intuitiva y eficiente para todos los perfiles de usuarios.

REQUISITOS Y CARACTERISTICAS DEL PROYECTO

Requisitos funcionales

El sistema SchoolMe deberá cumplir con las siguientes funcionalidades mínimas, de acuerdo con su alcance:

- Gestión de usuarios: creación, modificación y organización de datos de administradores, docentes, estudiantes y acudientes.
- Creación y administración de grupos académicos.
- Asignación de carga académica (materias, docentes y horarios).

- Diseño y gestión de agendas escolares por grado.
- Visualización de la agenda desde la versión móvil para estudiantes por medio del acudiente.

Requisitos no funcionales

- La interfaz debe ser intuitiva, clara y accesible para diferentes perfiles de usuario.
- El sistema debe ser adaptable a distintos dispositivos (responsive design).
- La plataforma debe permitir la personalización visual por institución (colores, nombre, logotipo).
- La arquitectura debe ser modular para facilitar futuras extensiones del sistema.
- El rendimiento debe permitir tiempos de respuesta aceptables en acciones comunes.

Características clave del proyecto

- Plataforma dividida en **versión web** (para personal institucional) y **versión móvil** (acudientes).
- Desarrollo basado en tecnologías modernas:
 - **Back-end:** C# con ASP.NET Core.
 - **Front-end web:** React o Angular.
 - **Móvil:** React Native.
- Personalización visual completa según la identidad institucional de cada colegio.
- Diseño de una interfaz basada en componentes reutilizables y con buenas prácticas de accesibilidad.
- Estructura pensada para futuras integraciones, sin comprometer la estabilidad del sistema base.

Oportunidades de crecimiento

SchoolMe, al ser una propuesta técnica en fase de diseño y desarrollo, presenta varias vías para una futura expansión. Estas oportunidades se centran en la **versatilidad** y la **capacidad de adaptación** del sistema que se está construyendo:

- **Ampliar el Alcance Educativo:** El diseño modular de SchoolMe permitirá que la plataforma pueda **adaptarse** a otros niveles educativos más allá de los colegios, como jardines infantiles o institutos técnicos. También se **considerará** su escalabilidad para ser implementada en múltiples instituciones o redes educativas, incluso a nivel nacional.
- **Desarrollar Módulos Complementarios:** Se puede **planificar el diseño** de funcionalidades adicionales que aumenten el valor de la plataforma. Esto incluiría un **módulo de gestión financiera básica** (para pagos de matrículas y pensiones), **herramientas de comunicación integral** (mensajería interna, foros, notificaciones personalizables), y **funcionalidades avanzadas para el seguimiento académico** (historial de desempeño detallado y reportes analíticos).
- **Fortalecer la Personalización y UX:** Además de la personalización visual, se puede **diseñar la plataforma** para ofrecer mayor flexibilidad en la configuración de la agenda (permitiendo a los colegios definir sus propias categorías de observaciones) y crear interfaces aún más adaptadas a las necesidades específicas de cada tipo de

usuario, especialmente los estudiantes.

- **Preparar para Integraciones Externas:** Desde el **diseño de la arquitectura**, se pueden **establecer las bases** para futuras integraciones. Esto implica **diseñar y documentar APIs** que permitan a SchoolMe comunicarse con otros sistemas, como plataformas contables o de aprendizaje virtual, facilitando así futuras expansiones sin comprometer la fase actual del proyecto.

Impacto esperado

El diseño y desarrollo de SchoolMe apunta a generar un impacto positivo en la gestión educativa, centrándose en los beneficios de una solución digital bien concebida:

- **Información Centralizada y Accesible:** La plataforma busca ser un **punto único para toda la información académica y administrativa**, facilitando el acceso rápido y organizado para todos los usuarios.
- **Mejor Seguimiento de Estudiantes:** Al registrar observaciones y datos en un formato digital estructurado, se facilitará un **seguimiento más coherente del desempeño y comportamiento** de cada estudiante.
- **Identidad Institucional Reforzada:** La personalización visual **permitirá que cada colegio mantenga su propia marca**, fortaleciendo su imagen digital.

SOLUCION PROPUESTA

SchoolMe es una **plataforma digital integral** que se propone como la respuesta a los desafíos de gestión académica y administrativa que enfrentan los colegios. Su desarrollo se centrará en la creación de un sistema robusto y adaptable, capaz de centralizar y simplificar procesos clave que, tradicionalmente, han sido manejados de forma manual y fragmentada.

Arquitectura de la plataforma

Backend: C# con ASP.NET Core (API RESTful)

La arquitectura de SchoolMe se concebirá como una **solución moderna basada en una arquitectura de capas bien definida (N-Capas)** para el backend, y un enfoque de **componentes desacoplados** en el frontend. Esto asegurará la mantenibilidad, escalabilidad y la adherencia a los principios de diseño limpio.

Frontend: React o Angular (Web) y React Native (Móvil)

Ambos frameworks fomentan una arquitectura basada en **componentes**. El objetivo es crear componentes **reutilizables, desacoplados y con responsabilidad única**.

Estructura Basada en Componentes/Módulos:

- La aplicación se dividirá en módulos (ej. Módulo de Gestión de Usuarios) y, dentro de cada módulo, en componentes (ej. TablaUsuarios).
- Se diferenciarán los componentes de presentación (dumb components), que solo reciben datos y exponen eventos, de los componentes contenedores (smart components), que gestionan el estado y la lógica.
- *Relación con SOLID*: Se busca aplicar el SRP (un componente hace una cosa bien) y el OCP (extensible con nuevos componentes sin modificar los existentes).

Base de Datos (SQL Server / PostgreSQL)

- Se optará por una base de datos relacional robusta que soporte las necesidades de la aplicación.
- El **diseño de la base de datos** será normalizado para asegurar la integridad y consistencia de los datos.
- La interacción se realizará a través de **Entity Framework Core** en el backend, lo que facilita el mapeo objeto-relacional y permite una abstracción de la base de datos.

Desglose de componentes y funcionalidades

1. Componente: Gestión de Personas

Funcionalidades:

Registro de Personas: Crear, modificar y eliminar registros de docentes, padres, niños y acudientes.

Validación de Datos: Asegurar que los datos ingresados estén completos y sean correctos.

Búsqueda y Filtrado: Permitir la búsqueda de personas por nombre, identificación o rol.

2. Componente: Registro de Grados y Creación de Grupos

Funcionalidades:

Registro de Grados: Crear grados (ejemplo: Primero, Segundo, Tercero, Cuarto).

Creación de Grupos: Crear grupos asociados a cada grado (ejemplo: Primero A, Primero B).

Asociación de Grupos a Grados: Vincular cada grupo a un grado específico.

Edición: Modificar grados y grupo.

3. Componente: Agenda Diaria del Docente

Funcionalidades:

Visualización de Agenda: Mostrar las clases programadas para el día actual, según los grupos asignados.

Detalles de Clase: Incluir información como tema, actividades y recursos necesarios.

Recordatorios: Notificar al docente sobre tareas pendientes o actividades importantes.

Navegación entre Días: Permitir la visualización de la agenda para días anteriores o siguientes.

4. Componente: Asignación de Dirección de Grupo y Carga Académica

Funcionalidades:

Asignación de Dirección de Grupo: Designar un docente como director de grupo para un curso específico.

Configuración de Carga Académica: Asignar asignaturas, docentes y horarios para cada curso.

Validación de Horarios: Evitar la superposición de horarios para docentes y grupos.

Reportes: Generar reportes de carga académica y dirección de grupo.

5. Componente: Seguridad y Permisos

Funcionalidades:

Roles y Permisos: Asignar roles (administrador, docente, acudiente) y permisos específicos.

Autenticación: Implementar un sistema de inicio de sesión seguro.

Auditoría: Registrar acciones críticas realizadas por los usuarios (ejemplo: creación, modificación, eliminación de datos).

6. Componente: Reportes y Análisis

Funcionalidades:

Generación de Reportes: Crear reportes personalizados (matrículas, calificaciones, asistencia, etc.).

Gráficos y Estadísticas: Mostrar datos en formato gráfico para facilitar el análisis.

Exportación: Permitir la exportación de reportes en formatos como PDF, Excel o CSV.

Resumen de Componentes Principales:

- Gestión de Personas
- Gestión de Colegios
- Registro de Grados y Creación de Grupos
- Agenda Diaria del Docente
- Asignación de Dirección de Grupo y Carga Académica
- Seguridad y Permisos
- Reportes y Análisis

PLAN DE TRABAJO

El plan de trabajo de ScholaWeb está estructurado en varias fases clave, cada una de ellas orientada a cumplir con los objetivos establecidos de forma eficiente, organizada y dentro de los plazos previstos. A continuación, se presenta un desglose detallado del plan, que incluye la metodología de trabajo, las fases del proyecto, los entregables y los plazos estimados para su desarrollo.

- Metodología de trabajo:

ScholaWeb se desarrollará utilizando la metodología ágil Scrum, lo que permite una planificación iterativa y flexible, con ciclos de trabajo cortos e incrementales. Esta metodología fomenta la colaboración continua entre el equipo de trabajo y los interesados, lo que facilita la adaptación a cambios durante el proceso de desarrollo. El enfoque está en la calidad del producto desde el inicio y en la mejora continua mediante ciclos de retroalimentación.

- Elementos clave de la metodología Scrum:

Sprints: Se dividirá el trabajo en ciclos cortos de desarrollo (sprints) de 2 a 4 semanas.

Reuniones diarias (Daily Standups): El equipo se reunirá brevemente cada día para revisar avances, identificar bloqueos y definir prioridades.

Revisión de Sprint: Al final de cada sprint, se revisarán los avances y se harán ajustes para mejorar el siguiente ciclo.

Retrospectivas: Se realizarán revisiones periódicas para analizar qué salió bien, qué se puede mejorar y cómo aplicar mejoras continuas.

1. Fase de análisis y planificación (1 mes):

- Objetivo: Definir todos los requisitos del proyecto, los objetivos de cada funcionalidad y la infraestructura técnica.
- Actividades principales:
- Mockup
- Documento de requisitos y objetivos.
- Mapa de funcionalidades del sistema.
- Cronograma detallado del proyecto.

2. Fase de diseño y prototipado (1 mes)

- Objetivo Crear el diseño visual y las interfaces de usuario (UI/UX) para la plataforma. Esto incluye la definición de las pantallas, los flujos de usuario y la experiencia interactiva.
- Actividades principales:
- Diseño de mockups.
- Prototipo funcional de la interfaz de usuario.
- Entregables:
 - mockups aprobados.

3. Fase de desarrollo inicial (2 meses):

- **Objetivo:** Desarrollar el backend y frontend del sistema, integrando las

funcionalidades básicas de la plataforma.

- Actividades principales:
 - Desarrollo de la infraestructura técnica (servidores, bases de datos, APIs).
 - Programación de las funcionalidades principales (registro de usuario, modos de juego, sistema de puntuación).
 - Integración de las primeras pantallas y pruebas de usabilidad.
 - Entregables
 - Backend funcional con gestión de usuarios y datos.
 - Interfaz de usuario básica implementada.
 - Prototipo funcional con los primeros módulos de la plataforma.

4. Fase de desarrollo y prueba (2 meses)

Objetivo: Continuar con el desarrollo de las funcionalidades avanzadas y realizar pruebas exhaustivas de la plataforma.

Actividades principales:

- Desarrollo de características avanzadas, como la agenda
 - Implementación de seguridad y pruebas de integración.
 - Pruebas de usabilidad y retroalimentación continua.
 - Realización de pruebas unitarias y pruebas de carga para garantizar el rendimiento.
- Entregables:
 - Funcionalidades completas del sistema (juegos, modos, progresión).
 - Resultados de pruebas de calidad.
 - Revisión de la plataforma según feedback recibido.

5. Fase de implementación y lanzamiento (1 mes)

Objetivo: Desplegar la plataforma a producción, realizar la puesta en marcha y asegurar el buen funcionamiento del sistema.

- Actividades principales:
 - Despliegue de la plataforma en servidores en vivo
 - Monitoreo del sistema para identificar posibles errores.
 - Resolución de posibles problemas detectados en producción.

- Capacitación inicial para los administradores de la plataforma y usuarios.
- **Entregables:**
 - Plataforma en vivo accesible para los usuarios.
 - Manual de usuario y guía de administración.
 - Informe de lanzamiento con los primeros resultados de uso.

Entregables claves en cada fase:

- **Fase de análisis y planificación:** Documento de requisitos y cronograma detallado.

- **Fase de diseño y prototipado:** Wireframes, mockups y prototipo funcional.
- **Fase de desarrollo inicial:** Backend funcional y primeras pantallas implementadas.
- **Fase de desarrollo y prueba:** Sistema completamente funcional con pruebas de calidad.
- **Fase de implementación y lanzamiento** Plataforma en producción y manual de usuario.
- **Fase de mantenimiento y mejoras continuas:** Actualizaciones periódicas y soporte técnico.

Conclusión:

El plan de trabajo de **ScholaWeb** está diseñado para garantizar un desarrollo estructurado y eficiente, asegurando el cumplimiento de cada fase dentro de los plazos establecidos. Desde el análisis y la planificación hasta la implementación y mantenimiento, cada etapa del proyecto cuenta con objetivos y entregables claves que facilitarán su correcta ejecución. Adaptarnos rápidamente a cualquier cambio o necesidad del proyecto, optimizando constantemente los resultados y garantizando una experiencia exitosa para el usuario final.

PRESENTACIÓN DEL EQUIPO DE TRABAJO

PROYECTO:


- SCHOLAWEB


Objetivo:


ScholaWeb es una plataforma digital flexible y personalizable diseñada para optimizar la gestión académica y administrativa de las instituciones educativas. Su objetivo principal es brindar autonomía a los colegios, mejorar la toma de decisiones mediante el análisis de datos y aprovechar la inteligencia artificial para impulsar el rendimiento educativo. La plataforma permite a las instituciones configurar su funcionalidad y diseño según sus necesidades específicas, facilitando la gestión, horarios, asignación de docentes. ScholaWeb fomenta la autonomía institucional al ofrecer un sistema adaptable que otorga a los colegios el control total de sus procesos, reduce la carga administrativa mediante una interfaz intuitiva y promueve una gestión escolar moderna y eficaz, generando confianza en la comunidad educativa.

1. Estructura del equipo:

Nombre completo	Rol	Responsabilidades
Jesus Fernando Carvajal Anacona	Líder del proyecto	coordinación general, planificación y seguimiento del equipo
Juan Felipe Guerrero Fandiño	Desarrollo Backend & Frontend	Colaborar en la integración de backend y frontend. Desarrollo de módulos o componentes específicos (backend y frontend).
Santiago Chaparro Riaño	Desarrollo Frontend & Backend	Colaborar en la integración de backend y frontend. Desarrollo de módulos o componentes específicos (backend y frontend).
(Todo el equipo de trabajo)	UX/UI	Diseño de interfaces y mejora de la experiencia del usuario

Fotografía	Datos personales
	<p>Nombres: Jesus Fernando</p> <p>Apellidos: Carvajal Anacona</p> <p>Edad: 18 años.</p> <p>Habilidades técnicas:</p> <ul style="list-style-type: none"> • Html & Css • Javascript • Git • Php • Sql <p>Experiencia previa:</p> <ul style="list-style-type: none"> • Técnico programación de software <p>Rol:</p> <p>➤ Líder del proyecto</p>
Responsabilidades	
<ul style="list-style-type: none"> • Planificación y seguimiento del cronograma del proyecto. • Asignación de tareas y recursos a los miembros del equipo. • Coordinación y comunicación entre todas las áreas del proyecto. • Supervisión del progreso y resolución de problemas. • Asegurarse de que los objetivos del proyecto se cumplan dentro del tiempo y presupuesto establecidos. • Desarrollo del proyecto • Despliegue y mantenimiento • Documentación 	

Fotografía	Datos personales
	<p>Nombres: Juan Felipe</p> <p>Apellidos: Guerrero Fandiño</p> <p>Edad: 18 años.</p> <p>Habilidades técnicas:</p> <ul style="list-style-type: none"> • Html & Css • Javascript • Git • Php • Sql <p>Experiencia previa:</p> <ul style="list-style-type: none"> • Técnico programación de software <p>Rol:</p> <ul style="list-style-type: none"> ➤ Desarrollador backend & Frontend
Responsabilidades	
<ul style="list-style-type: none"> • Documentación • Diseño de la interfaz de usuario • Experiencia de usuario • Integración con el backend • Visualización de datos • Adaptabilidad y compatibilidad • Pruebas y retroalimentación • Diseño y gestión de la base de dato • Lógica de negocio • Integración de inteligencia artificial • Seguridad y autenticación • Despliegue y mantenimiento 	

Fotografía	Datos personales
	<p>Nombres: Santiago</p> <p>Apellidos: Chaparro Riaño</p> <p>Edad: 18 años.</p> <p>Habilidades técnicas:</p> <ul style="list-style-type: none"> • Html & Css • Javascript • Git • Php • Sql <p>Experiencia previa:</p> <ul style="list-style-type: none"> • Técnico programación de software <p>Rol:</p> <ul style="list-style-type: none"> ➤ Desarrollador backend & Frontend
Responsabilidades	
<ul style="list-style-type: none"> • Documentación • Diseño de la interfaz de usuario • Experiencia de usuario • Integración con el backend • Visualización de datos • Adaptabilidad y compatibilidad • Pruebas y retroalimentación • Diseño y gestión de la base de dato • Lógica de negocio • Integración de inteligencia artificial • Seguridad y autenticación • Despliegue y mantenimiento 	

Documentación de responsabilidades

Responsabilidades Generales del Frontend

1. Diseño de la interfaz de usuario:

- Crear una interfaz atractiva, intuitiva y fácil de usar para docentes, estudiantes y administradores.
- Asegurar que el diseño sea consistente en toda la plataforma.

2. Experiencia de usuario:

- Asegurar que los usuarios puedan realizar sus tareas de manera eficiente (ejemplo: matricularse, ver horarios, consultar calificaciones).
- Probar la plataforma con usuarios reales para identificar y corregir problemas.

3. Integración con el backend:

- Conectar la interfaz de usuario con las APIs del backend para mostrar y gestionar datos en tiempo real.
- Asegurar que la comunicación entre el frontend y el backend sea fluida y sin errores.

4. Visualización de datos:

- Mostrar la información de manera clara y útil (ejemplo: gráficos de rendimiento, tablas de horarios, boletines de notas).
- Asegurar que los datos sean fáciles de entender y estén bien organizados.

5. Adaptabilidad y compatibilidad:

- Asegurar que la plataforma funcione correctamente en diferentes dispositivos (escritorio, tablet, móvil) y navegadores.
- Optimizar la plataforma para que sea rápida y eficiente.

6. Pruebas y retroalimentación:

- Realizar pruebas para asegurar que la plataforma funcione correctamente y sea fácil de usar.
- Recibir y aplicar retroalimentación de los usuarios para mejorar la plataforma.

Colaboración entre Backend y Frontend

1. Comunicación constante:

- Mantener una comunicación fluida entre ambos equipos para asegurar que el frontend y el backend estén alineados.

2. Definición de APIs:

- Acordar cómo serán las APIs y qué datos se enviarán y recibirán.

3. Pruebas de integración:

- Realizar pruebas conjuntas para asegurar que el frontend y el backend funcionen correctamente juntos.

4. Control de versiones:

- Usar un sistema de control de versiones para gestionar el código y facilitar la colaboración.

Responsabilidades Generales del Backend

1. Diseño y gestión de la base de datos:

- Crear y mantener la estructura de la base de datos para almacenar información de estudiantes, docentes, horarios, calificaciones, etc.
- Asegurar que los datos estén organizados, seguros y sean fáciles de acceder.

2. Desarrollo de APIs:

- Crear endpoints para que el frontend pueda interactuar con el backend (ejemplo: obtener datos de estudiantes, enviar calificaciones, gestionar horarios).
- Asegurar que las APIs sean eficientes, seguras y fáciles de usar.

3. Lógica de negocio:

- Implementar las reglas y procesos que rigen la plataforma (ejemplo: cómo se asignan los horarios, cómo se calculan los promedios, cómo se generan los boletines).
- Asegurar que la lógica sea clara, eficiente y esté bien documentada.

4. Integración de inteligencia artificial:

- Implementar herramientas para analizar el rendimiento de los estudiantes y generar recomendaciones basadas en datos.
- Asegurar que los análisis sean precisos y útiles para los usuarios.

5. Seguridad y autenticación:

- Implementar sistemas para proteger la plataforma y los datos de los usuarios.
- Gestionar el acceso de los usuarios (ejemplo: docentes, estudiantes, administradores) de manera segura.

6. Despliegue y mantenimiento:

- Asegurar que la plataforma esté disponible en línea y funcione correctamente.
- Realizar actualizaciones y correcciones según sea necesario.

