

Proyecto

Universidad Modelo

Proyectos V

Alumnos:

Joaquín Casillas Gonzalez:

Profesor:

Kenia Nayrhovy Osorio López

RESUMEN _____	3
INTRODUCCIÓN _____	3
ANTECEDENTES _____	4
¿Qué son las Páginas Web? _____	4
Servicios Utilizados en el Desarrollo Web _____	4
Herramientas para el Desarrollo Web _____	4
Proceso Actual _____	5
Back-end _____	6
Front-end _____	6
Base de datos _____	7
DEFINICIÓN DEL PROBLEMA _____	7
Problemas Presentes _____	8
Propuesta de Solución _____	8
JUSTIFICACIÓN _____	8
Razón para el Desarrollo del Proyecto _____	9
Beneficios Esperados _____	9
Impacto en los Beneficiarios _____	10
OBJETIVOS _____	10
General _____	10
Específicos _____	10
METODOLOGÍA _____	10
Herramientas a Utilizar _____	11
Resultados Esperados _____	12
DISEÑO CONCEPTUAL _____	12
Descripción Detallada del Proyecto _____	12
Usuarios y Aplicaciones _____	12
Funcionalidades Principales _____	13
Requerimientos del cliente _____	13
Diseño de pantallas _____	14
Herramienta a utilizar _____	16
1. Frontend: _____	16
2. Backend: _____	17
3. Móvil: _____	17

4. Base de Datos:	17
5. Dominios:	18
6. Pasarelas de Pago (Opcional, si se incluye en el proyecto):	18
Características del servidor	18
1. Descripción General	18
2. Características Técnicas	18
Repositorios para el desarrollo del proyecto	20
1. Sistema de Control de Versiones	20
2. Plataformas de Repositorios	20
3. Herramientas de Integración Continua y Entrega Continua (CI/CD)	20
4. Herramientas de Colaboración y Comunicación	20
5. Repositorios de Paquetes y Dependencias	21
6. Herramientas de Documentación	21
7. Gestión de Dependencias y Librerías	21
PLAN DE TRABAJO	21
COSTOS	22
GA(gastos administración):	22
PU (publicidad):	22
GE (ganancias):	22
IM (impuestos):	22
Plan de negocios	23
CONCLUSIONES	23
REFERENCIAS	25

RESUMEN

El proyecto consiste en el desarrollo de una aplicación web diseñada para la gestión de citas con el departamento de odontología dentro de la universidad modelo. La plataforma permitirá a las personas agendar y consultar con estudiantes postulantes a ser médicos cirujanos dentistas.

La página tiene como objetivo principal proporcionar un sistema accesible y gratuito que facilite la comunicación y la planificación de citas con los estudiantes de la universidad. Se espera que la plataforma mejore la eficiencia en la gestión de citas y el acceso a los servidores de salud disponibles, optimizando así la experiencia de los usuarios y promoviendo un entorno más organizado y accesible para el cuidado en la salud.

INTRODUCCIÓN

En la Universidad Modelo, los estudiantes del área de salud ofrecen una variedad de servicios prácticos, tales como clínicas dentales, fisioterapia, nutrición, readaptación deportiva y la Unidad de Atención y Prevención de la Salud. Actualmente, la gestión de citas para estos servicios se realiza predominantemente por teléfono, lo que limita la visibilidad de la disponibilidad de horarios y la información detallada sobre cada servicio.

La falta de una plataforma centralizada para la programación de citas puede resultar en una experiencia menos eficiente y más complicada para los usuarios interesados. La ausencia de un sistema en línea impide a los estudiantes y a la comunidad universitaria acceder fácilmente a la información sobre los servicios ofrecidos y a los horarios disponibles.

Nuestro proyecto busca abordar estas deficiencias mediante el desarrollo de una aplicación web que centralice y digitalice el proceso de agendamiento de citas. La plataforma proporcionará a los usuarios una interfaz intuitiva donde podrán consultar la disponibilidad de horarios y obtener descripciones detalladas de cada servicio. De este modo, esperamos mejorar la visibilidad de los servicios ofrecidos por los estudiantes de la Escuela de Salud y simplificar el proceso de agendado para todos los miembros de la universidad y sus pacientes.

ANTECEDENTES

Una página web es una presencia digital de un negocio o marca, lo cual ayuda a agilizar y crear más impacto lo que ayuda a llegar a más gente. En algunos emprendimientos suelen faltar secciones a dichas páginas para poder cumplir su propósito de agilizar tareas que a la gente luego no le gusta mucho gestionar. Según una socióloga apasionada por el marketing muy conocida “Gisela C.Díaz” han identificado que poder hacer una cita en línea atrae más gente y causa más clientela ya que suele ser algo más privado y sencillo.

¿Qué son las Páginas Web?

Las páginas web son documentos accesibles a través de internet que pueden contener texto, imágenes, videos y otros elementos multimedia. Estas páginas se desarrollan utilizando tecnologías como HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) y JavaScript, y se alojan en servidores web para ser accesibles desde navegadores de internet.

Servicios Utilizados en el Desarrollo Web

Para el desarrollo de una aplicación web de citas, se emplean varios servicios y tecnologías:

- **Servicios de Hosting:** Proporcionan el espacio en servidores para alojar la aplicación web. Ejemplos incluyen servicios como AWS (Amazon Web Services), Google Cloud Platform y Microsoft Azure.
- **Sistemas de Gestión de Bases de Datos (DBMS):** Administran el almacenamiento y la gestión de datos. Ejemplos comunes son MySQL, PostgreSQL, y MongoDB.
- **APIs (Interfaces de Programación de Aplicaciones):** Permiten la integración de servicios externos, como servicios de correo electrónico o autenticación de usuarios. Ejemplos incluyen SendGrid para correos electrónicos y Auth0 para autenticación.

Herramientas para el Desarrollo Web

El desarrollo de una aplicación web requiere diversas herramientas para facilitar la codificación, la gestión de proyectos y el control de versiones:

- Editores de Código: Herramientas como Visual Studio Code, Sublime Text y Atom son fundamentales para escribir y editar el código fuente de la aplicación.
- Frameworks y Librerías: Utilizados para simplificar y acelerar el desarrollo. En el front-end, frameworks como React, Angular y Vue.js ofrecen componentes reutilizables y herramientas avanzadas. En el back-end, frameworks como Django (Python), Rails (Ruby), y Laravel (PHP) ayudan a gestionar la lógica del servidor y la base de datos.
- Sistemas de Control de Versiones: Herramientas como Git y plataformas como GitHub o GitLab permiten gestionar el código fuente, colaborar en el desarrollo y realizar seguimiento de cambios.

Proceso Actual

Actualmente, en la Universidad Modelo, los estudiantes acceden a servicios de salud como odontología, fisioterapia y nutrición principalmente a través de llamadas telefónicas. Este método presenta varias limitaciones:

- Falta de visibilidad: Los usuarios no pueden ver horarios disponibles ni detalles específicos sobre los servicios ofrecidos.
- Ineficiencia: La gestión manual de citas a través de llamadas puede ser propensa a errores y es menos eficiente en comparación con un sistema digital.
- Experiencia del usuario: La falta de una plataforma en línea puede hacer que el proceso sea menos accesible y conveniente para los estudiantes.

El proyecto necesitará de 3 importantes pilares para funcionar de forma eficaz: Back-end, Front-end y La base de datos.

Back-end

El desarrollo del back-end es crucial para el funcionamiento de una aplicación web, ya que maneja la lógica del servidor y la base de datos. Las principales tecnologías y lenguajes utilizados incluyen:

- Python: Conocido por su sintaxis sencilla y sus potentes bibliotecas, Python es ampliamente utilizado en el desarrollo de aplicaciones web dinámicas. Su versatilidad y la existencia de frameworks como Django y Flask lo hacen una opción popular.
- JavaScript: Aunque es conocido principalmente por su uso en el front-end, JavaScript también tiene una fuerte presencia en el back-end a través de Node.js, que permite la creación de aplicaciones rápidas y escalables.
- Ruby: Ruby, junto con el framework Ruby on Rails, es conocido por su elegancia y rapidez en el desarrollo de aplicaciones web. Rails proporciona una estructura robusta y eficiente para el desarrollo rápido.
- PHP: Con una larga trayectoria en el desarrollo web, PHP sigue siendo una opción confiable para aplicaciones web. Su amplia adopción y la existencia de frameworks como Laravel lo convierten en una herramienta versátil.
- Java: Utilizado frecuentemente en grandes empresas debido a su robustez y portabilidad. Java ofrece un rendimiento sólido y es ideal para aplicaciones de gran escala.

La elección de la tecnología para el back-end dependerá de diversos factores, como las necesidades específicas del proyecto, la experiencia del equipo y los recursos disponibles.

Front-end

El front-end se centra en la parte visual de la aplicación y la interacción del usuario. Los desarrolladores front-end utilizan tecnologías como:

- HTML y CSS: Lenguajes fundamentales para la estructuración y el diseño de páginas web.

- Frameworks y librerías: Herramientas como React, Angular y Vue.js facilitan la creación de interfaces interactivas y mejoran la experiencia del usuario. Estos frameworks ofrecen componentes reutilizables y herramientas que optimizan el proceso de desarrollo.

Base de datos

La base de datos es esencial para almacenar y gestionar la información de las citas y los usuarios. Algunas opciones populares incluyen:

- MySQL: Una base de datos relacional ampliamente utilizada que es conocida por su fiabilidad y flexibilidad.
- SQL Server: Ofrecido por Microsoft, es conocido por su integración con otras herramientas de Microsoft y su capacidad para manejar grandes volúmenes de datos.
- MariaDB: Un fork de MySQL que ofrece características adicionales y mejoras en el rendimiento, manteniendo la compatibilidad con MySQL.

Elegir la base de datos adecuada dependerá de los requisitos específicos del proyecto, como el volumen de datos y las necesidades de rendimiento.

DEFINICIÓN DEL PROBLEMA

Actualmente, los estudiantes de la Universidad Modelo enfrentan desafíos significativos al intentar agendar citas con los servicios de salud proporcionados por sus compañeros de la Escuela de Salud. Estos servicios incluyen atención en áreas como odontología, fisioterapia, nutrición, y más. La ausencia de un sistema centralizado para la gestión de citas causa problemas como:

- Ineficiencia en la Organización: La falta de una plataforma unificada lleva a una gestión desorganizada de las citas. Los estudiantes deben comunicarse a través de llamadas telefónicas o correos electrónicos, lo que resulta en una coordinación poco eficiente y en la posibilidad de errores.
- Pérdida de Tiempo: Tanto los estudiantes que buscan servicios de salud como aquellos que los ofrecen pierden tiempo debido a la necesidad de coordinar citas manualmente. Esto afecta la disponibilidad y el aprovechamiento del tiempo de ambos grupos.

- **Acceso Limitado:** Los estudiantes tienen dificultades para acceder a información actualizada sobre la disponibilidad de horarios y los detalles de los servicios ofrecidos, lo que puede llevar a la frustración y una menor utilización de los servicios disponibles.

Problemas Presentes

Los problemas descritos se presentan específicamente en la Universidad Modelo y afectan a la población estudiantil que busca servicios de salud proporcionados por sus compañeros en la Escuela de Salud. Estos problemas están acotados en tiempo, ya que se experimentan diariamente, y en lugar, se circunscriben al entorno universitario.

Propuesta de Solución

Para abordar estos problemas, se propone desarrollar una aplicación web gratuita destinada a los estudiantes de la Universidad Modelo. Esta plataforma digital permitirá:

- **Gestión Simplificada de Citas:** Los estudiantes podrán agendar, modificar y cancelar citas de manera rápida y sencilla a través de una interfaz en línea, eliminando la necesidad de llamadas telefónicas o correos electrónicos.
- **Acceso a Información Actualizada:** La aplicación proporcionará detalles en tiempo real sobre la disponibilidad de horarios y descripciones de los servicios ofrecidos, facilitando la planificación y mejorando la experiencia del usuario.
- **Mejora en la Coordinación y Eficiencia:** Con un sistema centralizado, se optimizará la organización de citas y se reducirá el tiempo perdido en la coordinación, beneficiando a toda la comunidad universitaria.

Esta herramienta no solo mejorará la accesibilidad y la eficiencia en la gestión de citas, sino que también contribuirá a una mejor organización y aprovechamiento de los recursos de salud disponibles en la universidad.

JUSTIFICACIÓN

Razón para el Desarrollo del Proyecto

Actualmente, los estudiantes de la Universidad Modelo enfrentan barreras significativas para acceder a los servicios de salud ofrecidos por sus compañeros de la Escuela de Salud. La falta de un sistema accesible y centralizado para la gestión de citas implica que los estudiantes deben acudir personalmente al área de salud para obtener información y agendar una cita, o hacerlo a través de llamadas telefónicas. Este proceso no solo es ineficiente, sino que también crea inconvenientes y demoras innecesarias para los estudiantes y para el personal encargado.

Beneficios Esperados

El desarrollo de una aplicación web para la gestión de citas tiene el potencial de transformar significativamente la manera en que los estudiantes acceden a los servicios de salud. Los principales beneficios incluyen:

- **Accesibilidad Mejorada:** La plataforma permitirá a los estudiantes consultar la disponibilidad de horarios, conocer detalles sobre los servicios ofrecidos y agendar citas en línea desde cualquier lugar y en cualquier momento. Esto elimina la necesidad de desplazarse físicamente al área de salud, reduciendo barreras de acceso y facilitando la planificación.
- **Eficiencia en la Gestión de Citas:** Un sistema digital centralizado optimizará el proceso de agendamiento, reduciendo el tiempo perdido en la coordinación manual y minimizando los errores en la programación de citas. Esto permitirá un uso más eficiente de los recursos del área de salud.
- **Mejora en la Experiencia del Usuario:** Al proporcionar una interfaz intuitiva y accesible, la aplicación mejorará la experiencia general de los estudiantes al interactuar con los servicios de salud. La facilidad para consultar horarios y servicios contribuirá a una mayor satisfacción y aprovechamiento de los servicios disponibles.
- **Optimización de Recursos:** La implementación del sistema permitirá una mejor gestión y seguimiento de las citas, lo que a su vez puede contribuir a una mayor eficiencia en el uso de los recursos de salud. Los estudiantes y el personal podrán beneficiarse de una coordinación más efectiva y menos tiempos de espera.

Impacto en los Beneficiarios

El proyecto beneficiará directamente a los estudiantes de la Universidad Modelo, facilitando su acceso a servicios de salud esenciales. Asimismo, el personal de salud también se beneficiará al contar con una herramienta que simplifica la gestión de citas y reduce la carga administrativa. En última instancia, la implementación de esta aplicación web contribuirá a una mejora general en la eficiencia y accesibilidad de los servicios de salud universitarios, beneficiando a toda la comunidad académica.

OBJETIVOS

General

Desarrollar una aplicación web de citas médicas, estas citas médicas pueden ser de odontología, fisioterapia, nutrición, consulta psicológica.

Específicos

1. Desarrollar una interfaz atractiva y dinámica
2. Investigar los servicios que ofrece la escuela de salud
3. Utilizar herramientas que faciliten el desarrollo de la aplicación web

METODOLOGÍA

- **Análisis de Requerimientos:** Reunión con los stakeholders (estudiantes, personal de salud, etc.) para identificar las necesidades y requisitos del sistema de gestión de citas. Esto incluye definir qué funcionalidades serán necesarias en la aplicación.
- **Diseño del Sistema:** Creación de diagramas y esquemas para el diseño de la arquitectura de la aplicación. Esto incluye el diseño de la interfaz de usuario (UI) y la estructura de la base de datos.

- Desarrollo del Back-end: Implementación del servidor y la lógica del negocio utilizando tecnologías como Python, JavaScript (Node.js), Ruby, PHP o Java. Esto incluye el desarrollo de APIs para la gestión de citas, autenticación y acceso a datos.
- Desarrollo del Front-end: Creación de la interfaz de usuario utilizando HTML, CSS y JavaScript. Se emplearán frameworks como React, Angular o Vue.js para construir una interfaz interactiva y receptiva.
- Integración y Pruebas: Integración del front-end y back-end, y realización de pruebas para asegurar que la aplicación funcione correctamente. Esto incluye pruebas de funcionalidad, usabilidad y rendimiento.
- Despliegue y Mantenimiento: Publicación de la aplicación en un entorno de producción y monitoreo continuo para resolver problemas y realizar mejoras según sea necesario. Esto incluye la implementación en un servidor web y la configuración de bases de datos.

Herramientas a Utilizar

- Herramientas de diseño: Herramientas como Figma o Adobe XD para el diseño de interfaces y prototipos.
- Lenguajes y Frameworks:
 - Back-end: Python (Django, Flask), JavaScript (Node.js), Ruby (Rails), PHP (Laravel), Java (Spring Boot).
 - Front-end: HTML, CSS, JavaScript, y frameworks/librerías como React, Angular, Vue.js.
- Bases de datos: MySQL, PostgreSQL, MariaDB para el almacenamiento y gestión de datos.
- Sistemas de control de versiones: Git y plataformas como GitHub o GitLab para el control de versiones y la colaboración.
- Herramientas de pruebas: Herramientas como Jest, Mocha para pruebas de JavaScript, y Postman para pruebas de APIs.
- Servicios de hosting: Plataformas como AWS, Google Cloud Platform, o Microsoft Azure para el despliegue de la aplicación.

Resultados Esperados

- **Aplicación Web Funcional:** Un sistema operativo y accesible para la gestión de citas, que permita a los estudiantes agendar, modificar y cancelar citas de manera eficiente.
- **Mejora en la accesibilidad:** Aumento en la facilidad de acceso a servicios de salud y reducción de la necesidad de coordinar citas manualmente.
- **Optimización de recursos:** Mejoras en la eficiencia de la gestión de citas y en la utilización de recursos de salud disponibles.
- **Satisfacción del usuario:** Incremento en la satisfacción de los usuarios con respecto a la accesibilidad y la experiencia general de uso.

DISEÑO CONCEPTUAL

Descripción Detallada del Proyecto

- El proyecto consiste en el desarrollo de una plataforma web para la gestión de citas en el área de salud de la Universidad Modelo. La página será una extensión del sitio web institucional, diseñada para facilitar el acceso a los servicios de salud disponibles, así como la programación y monitoreo de citas por parte de los usuarios.

Usuarios y Aplicaciones

- **Cliente:**
 - **Descripción:** El usuario (estudiante o personal de la universidad) podrá acceder a la página y:
 - Ver los servicios que ofrece la escuela de salud.
 - Consultar el perfil de las diferentes carreras y de algunos estudiantes representantes.
 - Agendar, monitorear y cancelar citas según sus necesidades.
 - Acceder a la sección de preguntas frecuentes y enviar consultas.
- **Operador:**
 - **Descripción:** El operador (personal de salud o administrativo) podrá:
 - Ver y gestionar las citas agendadas.
 - Actualizar información sobre los servicios y profesionales disponibles.
 - Monitorear el flujo de pacientes que llegan para atención.
 - Proporcionar asistencia a los usuarios en caso de dudas o inconvenientes.
- **Administrador:**

- **Descripción:** El administrador (personal encargado del sistema) tendrá las siguientes funciones:
 - Gestionar y configurar la base de datos de usuarios, servicios y citas.
 - Supervisar el rendimiento de la plataforma y realizar ajustes según sea necesario.
 - Generar reportes sobre la utilización de los servicios de salud y la gestión de citas.
 - Responder a preguntas y gestionar la sección de preguntas frecuentes.

Funcionalidades Principales

- **Interfaz de Usuario Amigable:** Diseño intuitivo que facilita la navegación.
- **Sistema de Agendamiento:** Permite a los usuarios seleccionar servicios, fechas y horas disponibles.
- **Gestión de Citas:** Herramientas para que los operadores y administradores puedan modificar, cancelar y supervisar citas.
- **Base de Datos:** Almacenamiento seguro de información de usuarios, citas y servicios.
- **Sección de FAQ:** Respuestas a preguntas comunes y un formulario para consultas.

Requerimientos del cliente

Los requerimientos establecidos por el cliente se muestran en la Tabla 1. (Descripción).

Tabla 1. Requerimientos del cliente.

ELEMEN TO	PROCESO	REQUERIMIENTO	USUARIO	CRITERIOS/NOT AS
Cita	Agendar cita	El usuario podrá agendar una cita de forma fácil y accesible	Cliente	Citas No duplicadas
	Asignar folio	Se generará un folio para que el cliente pueda monitorear, editar o cancelar su cita.	Cliente	Folio único
	Cancelar cita	El cliente puede cancelar citas	Cliente	
Bugs	Citas duplicadas	No se debería asignar una cita si no hay cupo	Cliente	

	No disponibilidad	No se debería poder asignar una cita los días que no se atienda.	Cliente	
--	-------------------	--	---------	--

Diseño de pantallas



Figura1. Pagina de inicio



Figura2. Página "Nuestros profesionales"

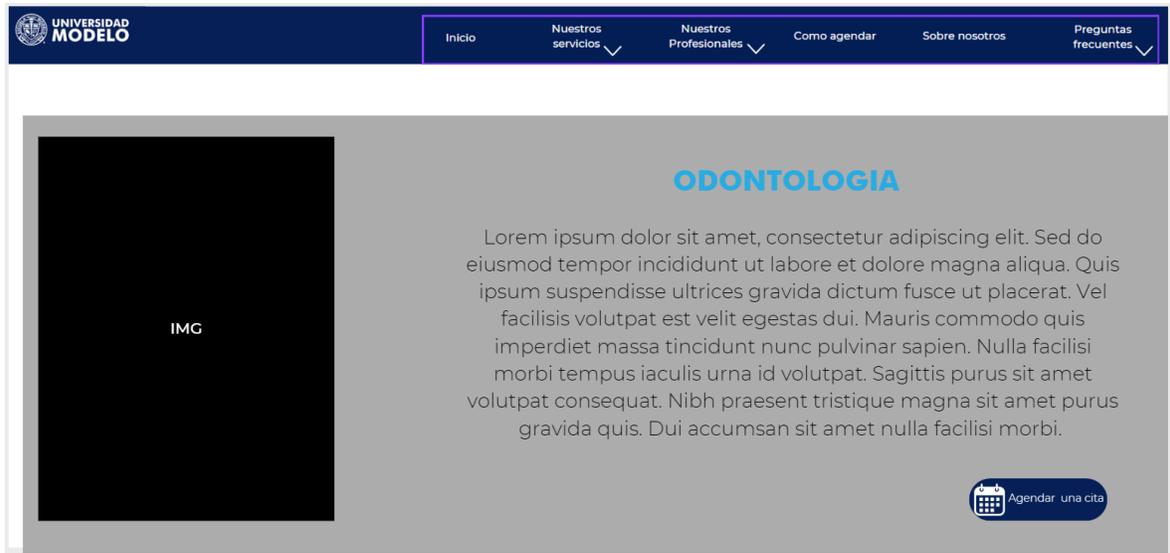


Figura3. Página "Agendar cita"



Figura4. Página " Como agendar"



Figura5. Página "Sobre nosotros"



Figura6." Formulario para agendar"

Herramienta a utilizar

1. Frontend:

- **HTML5, CSS3 y JavaScript:** Para la estructura y diseño de la página web, asegurando compatibilidad con navegadores modernos.

- **Frameworks de Frontend (React o Angular):** Herramientas para crear una interfaz de usuario interactiva y responsiva, facilitando la gestión del estado y la renderización eficiente.
- **Bootstrap o Tailwind CSS:** Frameworks CSS para asegurar un diseño moderno y adaptable, optimizando la experiencia del usuario en diferentes dispositivos (móvil, tablet, escritorio).

2. Backend:

- **Node.js con Express:** Plataforma para el desarrollo del backend, permitiendo la creación de una API REST que gestionará las solicitudes y respuestas de los usuarios y operadores.
- **Autenticación JWT (JSON Web Tokens):** Para manejar sesiones seguras y autenticación de usuarios (clientes, operadores y administradores).
- **Framework de pruebas (Jest o Mocha):** Para realizar pruebas unitarias y asegurar que el backend funcione correctamente.

3. Móvil:

- **React Native o Flutter:** Frameworks que permiten desarrollar aplicaciones móviles multiplataforma (iOS y Android), basadas en el código web, para que los usuarios puedan gestionar citas desde dispositivos móviles.
- **Progressive Web App (PWA):** Alternativa que permitirá que la plataforma web funcione como una aplicación móvil sin necesidad de instalación, brindando una experiencia nativa en cualquier dispositivo.

4. Base de Datos:

- **MongoDB o MySQL:** Dependiendo de la necesidad del proyecto:
 - **MongoDB:** Para una estructura flexible de datos, ideal para manejar los registros de usuarios, citas y servicios de manera escalable.
 - **MySQL:** Para una base de datos relacional, si se necesita una estructura más definida para los registros.
- **Mongoose (para MongoDB) o Sequelize (para MySQL):** ORM (Object-Relational Mapping) que facilita la interacción entre la aplicación y la base de datos.

5. Dominios:

- **Proveedor de Dominios (GoDaddy, Namecheap, etc.):** Para el registro del dominio web asociado a la extensión de la página de la universidad.
- **Certificado SSL (Let's Encrypt):** Para asegurar la navegación con HTTPS y proteger la información de los usuarios.

6. Pasarelas de Pago (Opcional, si se incluye en el proyecto):

- **Stripe o PayPal:** En caso de que la plataforma requiera procesar pagos, estas pasarelas permitirán gestionar pagos seguros por servicios de salud o consultas especializadas.
- **Integración con el backend:** A través de API de pago, se vinculará con el sistema para registrar y confirmar transacciones.

Características del servidor

1. Descripción General

El servidor es la infraestructura central que soporta la operación de la plataforma web de gestión de citas del área de salud de la Universidad Modelo. Su función principal es manejar las solicitudes de los usuarios, operadores y administradores, asegurando que la información fluya de manera segura y eficiente entre el frontend y el backend. El servidor también se encarga del almacenamiento de los datos, la autenticación de usuarios, y el procesamiento de las operaciones administrativas.

2. Características Técnicas

4. Tipo de Servidor:

- **Servidor en la nube (Cloud Server):** Se recomienda usar un servidor en la nube como AWS (Amazon Web Services), Google Cloud, o Microsoft Azure para asegurar escalabilidad y alta disponibilidad. Esto permitirá ajustar los recursos según la demanda de los usuarios (picos de tráfico) y asegurar que la plataforma esté siempre accesible.
- **Servidor dedicado o VPS (Virtual Private Server):** Alternativa viable si se prefiere mayor control sobre los recursos y configuraciones del servidor.

5. Sistema Operativo:

- **Linux (Ubuntu o CentOS):** Se recomienda un sistema operativo basado en Linux debido a su robustez, seguridad y compatibilidad con la mayoría de las tecnologías web (Node.js, bases de datos, etc.).
 - **Seguridad:** Configuraciones adicionales de firewall (iptables), fail2ban y actualizaciones automáticas de seguridad.
- 6. Capacidad de Almacenamiento:**
- **Espacio en disco:** Se recomienda un disco SSD (Solid State Drive) de al menos 100 GB para asegurar un acceso rápido a los archivos y bases de datos.
 - **Backup automático:** Implementar un sistema de copias de seguridad automatizadas para prevenir la pérdida de datos en caso de fallos.
- 7. Capacidad de Procesamiento:**
- **CPU:** Para un servidor de tamaño medio, un procesador con al menos 4 núcleos (4 vCPUs) garantizará un rendimiento adecuado al manejar múltiples solicitudes simultáneas.
 - **RAM:** Se recomienda al menos 8 GB de RAM para manejar la carga operativa de la plataforma sin interrupciones, sobre todo en momentos de alta demanda.
- 8. Conectividad y Red:**
- **Ancho de banda:** Para asegurar una buena experiencia de usuario, se recomienda al menos una conexión de red de 1 Gbps. Esto garantiza tiempos de respuesta rápidos y evita cuellos de botella en el tráfico de datos.
 - **CDN (Content Delivery Network):** Implementar una CDN como Cloudflare o Amazon CloudFront para mejorar la velocidad de carga al distribuir el contenido estático (imágenes, CSS, JavaScript) desde servidores más cercanos al usuario.
- 9. Escalabilidad y Disponibilidad:**
- **Autoescalado (Auto-scaling):** En caso de usar un servidor en la nube, configurar reglas de autoescalado que permitan incrementar o reducir recursos automáticamente según el tráfico.
 - **Alta disponibilidad (High Availability):** Configurar un sistema de balanceo de carga (load balancer) para distribuir las solicitudes entre múltiples instancias del servidor y evitar caídas del sistema en caso de fallos.
- 10. Seguridad del Servidor:**
- **Certificado SSL:** Implementación de un certificado SSL para garantizar que todas las comunicaciones entre el servidor y el cliente se realicen de forma encriptada (HTTPS).
 - **Autenticación de doble factor (2FA):** Para los operadores y administradores, implementar un sistema de autenticación de doble factor para mejorar la seguridad en el acceso a la plataforma.
 - **Firewall y protección DDoS:** Asegurar el servidor con firewalls y sistemas de protección ante ataques de denegación de servicio (DDoS), como los ofrecidos por servicios como Cloudflare o AWS Shield.
- 11. Mantenimiento y Monitoreo:**
- **Monitoreo de servidores (Prometheus, Grafana):** Implementar herramientas para el monitoreo constante del rendimiento del servidor (CPU, RAM, tráfico de red), así como para recibir alertas en caso de posibles fallos.

- **Mantenimiento periódico:** Realizar actualizaciones de software y parches de seguridad regularmente para evitar vulnerabilidades.

Repositorios para el desarrollo del proyecto

1. Sistema de Control de Versiones

- **Git:** Se utilizará **Git** como el sistema de control de versiones para gestionar el código fuente del proyecto. Esto permitirá a los desarrolladores colaborar de manera eficiente, llevar un historial de cambios, revertir modificaciones si es necesario y trabajar en múltiples ramas (features) simultáneamente.

2. Plataformas de Repositorios

- **GitHub:** Plataforma principal para alojar los repositorios del proyecto. GitHub facilita la colaboración mediante herramientas como pull requests, revisión de código y la integración continua (CI/CD). Se puede configurar un repositorio privado para mantener el proyecto seguro hasta que esté listo para su publicación.
- **GitLab:** Alternativa a GitHub que ofrece características similares con mayor flexibilidad para configuraciones de CI/CD integradas y hosting privado si se necesita mayor control sobre los datos del proyecto.
- **Bitbucket:** Otra alternativa que ofrece integración con Jira (herramienta de gestión de proyectos) y permite manejar repositorios privados con equipos pequeños de manera gratuita.

3. Herramientas de Integración Continua y Entrega Continua (CI/CD)

- **GitHub Actions (o GitLab CI/CD):** Estas herramientas permitirán automatizar las pruebas y el despliegue del código. Cada vez que se realicen cambios en el repositorio, GitHub Actions o GitLab CI/CD podrán ejecutar automáticamente pruebas unitarias, realizar un análisis de calidad del código y, eventualmente, desplegar el prototipo en un entorno de pruebas o producción.
- **Jenkins:** Como una opción adicional para la integración continua, Jenkins puede ser utilizado para gestionar el ciclo de vida del desarrollo del proyecto, desde el build hasta el despliegue automático.

4. Herramientas de Colaboración y Comunicación

- **Trello o Jira:** Herramientas para la gestión del proyecto y las tareas. Se utilizarán para crear tableros kanban o listas de tareas para organizar las actividades del equipo de

desarrollo. Esto facilita la planificación de sprints, seguimiento de tareas y gestión del backlog.

- **Slack o Microsoft Teams:** Para la comunicación del equipo, integrando notificaciones automáticas de los repositorios y herramientas de CI/CD, de manera que el equipo esté siempre actualizado con el progreso del proyecto.

5. Repositorios de Paquetes y Dependencias

- **npm (Node Package Manager):** Se utilizará para gestionar las dependencias del proyecto, instalando y actualizando paquetes necesarios para el desarrollo frontend y backend (por ejemplo, React, Express, etc.).
- **Docker Hub:** Si se opta por contenerizar la aplicación, Docker Hub será utilizado para almacenar imágenes de contenedores Docker, facilitando el despliegue y la replicación del entorno de desarrollo en diferentes máquinas y entornos.

6. Herramientas de Documentación

- **Markdown en GitHub/GitLab:** Los archivos de documentación, como el README.md, serán escritos en formato Markdown para proporcionar instrucciones claras sobre el uso y desarrollo del proyecto. GitHub y GitLab ofrecen visualización de estos archivos de manera clara dentro del repositorio.
- **Confluence:** Si se requiere una herramienta más robusta para la documentación, Confluence puede utilizarse para crear una wiki interna donde se detallen los aspectos técnicos y organizativos del proyecto.

7. Gestión de Dependencias y Librerías

- **SonarQube:** Para el análisis estático del código y la revisión de su calidad. SonarQube puede integrarse con GitHub, GitLab o Bitbucket para realizar un análisis en cada commit o pull request y detectar posibles errores, vulnerabilidades o problemas de calidad.

PLAN DE TRABAJO

Tabla 2. Diagrama de Gantt con las actividades a realizar para el desarrollo de la aplicación...

Actividades	Plan de acción (2024-2025) Ago2024-jun2025																																															
	Sep				Oct				Nov				Dic				Ene				Feb				Mar				Abr				May				Jun											
Semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Análisis de requerimientos (A,C)																																																
Diseño del sistema (I,C,A,A)																																																
Desarrollo Base de datos (C,G)																																																
Desarrollo Back-end (G,C)																																																
Desarrollo Front-end (I,A)																																																
Testing Intergral (A,I)																																																
Prototipo (I,A)																																																
Continuacionn Back-end (G,C)																																																
Continuacionn Front-end (I,A)																																																
Chequeo de base de datos (A,I)																																																
Finalización Back-end (G,C)																																																
Finalización Front-end (I,A)																																																
Testing Intergral (A,I)																																																
Testing Aceptación (A,I)																																																
Preparacion despliegue (I,A,G)																																																
Despliegue (I,G,A,C,A)																																																

COSTOS

GA(gastos administración):

- € Host: \$40.00
- € Cloud: \$224
- € Sueldo: \$44,000

PU (publicidad):

Ningún gasto

GE (ganancias):

- € \$44,264
- € \$50,000

IM (impuestos):

- € ISR: 30% sobre utilidades.

Costo Mensual= 109,264

$$\text{Costo} = [\text{GE} * (1 + \text{IM}) + \text{GA} + \text{PU}] / (\text{plazo})$$

Plan de negocios

Se tienen 2 posibles opciones que proporcionen a los clientes, una sería vender el producto, y manejándolo como sesiones y calcularlo conforme a cuánto tiempo se espera terminar el proyecto y cuantos recursos se utilizan para la creación del proyecto y de ahí se saca el precio del producto y se le vende al cliente y después de venderlo ya no se tiene ninguna interacción.

La otra posibilidad de vender el producto sería, venderlo como un servicio, el cual se cobraría según los recursos utilizados y el personal que trabaje en ello, y se cobraría una mensualidad al cliente. Este modelo trae ventajas como posicionamiento, análisis y personalización en tiempo real con el cliente.

CONCLUSIONES

Joaquín: El proyecto de gestión de citas no solo simplificará el acceso a los servicios de salud, sino que también fomentará una mayor interacción y confianza entre estudiantes y personal. Más allá de cumplir con sus objetivos técnicos, la plataforma mejorará la comunicación y optimizará la organización en el área de salud, creando un entorno más eficiente y accesible para toda la comunidad universitaria.

Angel: El proyecto de gestión de citas para el área de salud de la Universidad Modelo promete hacer la vida más fácil tanto para los estudiantes como para el personal. Con una plataforma web fácil de usar, se podrá agendar, modificar y cancelar citas sin complicaciones, lo que hará que acceder a los servicios de salud sea mucho más sencillo.

La idea de contar con una interfaz intuitiva y herramientas como una sección de preguntas frecuentes es genial para mejorar la experiencia del usuario. Además, asegurarnos de que la plataforma sea accesible desde cualquier dispositivo hará que más personas la utilicen sin problemas.

Espero que, al final, tengamos una plataforma que no solo sea útil, sino que también mejore la experiencia de nuestros compañeros en la universidad.

Camila León: Trabajar en el proyecto de la aplicación web para la gestión de citas en servicios de salud dentro de la Universidad Modelo fue una experiencia nueva y difícil ya que yo no tengo experiencia en el ámbito del desarrollo web. A lo largo del proyecto, aprendí cómo integrar tecnologías de front-end y un poco de back-end para construir una plataforma eficiente y accesible, lo cual me ayudó a mejorar mis habilidades tanto en desarrollo web como en la gestión de bases de datos.

También pudimos poner en práctica lo aprendido en la materia de investigación de mercados para conocer las necesidades de los usuarios de nuestra página web.

La experiencia de trabajar en equipo, como siempre fue retadora, pero fundamental para el éxito del proyecto. Colaborar con diferentes áreas, desde mis compañeros hasta estudiantes de salud y administrativos, me permitió adquirir una visión más amplia de la importancia de la comunicación y la coordinación en proyectos complejos.

Mis expectativas para el desarrollo futuro del proyecto son muy positivas. Estoy convencida de que la aplicación no solo mejorará la experiencia de los usuarios al agendar citas, sino que también servirá como una base sólida para futuras implementaciones y mejoras. Espero que el proyecto se pueda concluir de manera correcta y pueda ser funcional.

Gonzalo Gamboa: El proyecto de gestión de citas para el área de salud de la Universidad Modelo tiene como objetivo fundamental mejorar el acceso y la eficiencia en la prestación de servicios de salud para los estudiantes y el personal. Más allá de los aspectos técnicos, la visión para la plataforma es crear una herramienta integral que no solo permita a los usuarios gestionar citas de manera sencilla y accesible, sino que también fortalezca la conexión entre los estudiantes y los servicios profesionales disponibles.

La interfaz intuitiva y el acceso desde cualquier dispositivo aseguran que el proceso de agendar, modificar, y cancelar citas sea lo más fluido posible. Asimismo, se busca ofrecer una experiencia óptima para el usuario, integrando funcionalidades adicionales, como una sección de preguntas frecuentes, para facilitar el uso de la plataforma.

Ana Gómez: El proyecto de gestión de citas para el área de salud de la Universidad Modelo tiene como objetivo mejorar la accesibilidad y eficiencia en la prestación de servicios. A través de una plataforma web con una interfaz intuitiva y accesible desde cualquier dispositivo, se facilitará la gestión de citas para estudiantes y personal, permitiendo agendar, modificar y cancelar citas de manera sencilla. Además, se incluyen funciones como una sección de preguntas frecuentes para mejorar la experiencia del usuario.

La colaboración entre diferentes áreas ha sido clave, permitiendo el desarrollo de una solución integral que no solo responde a las necesidades técnicas, sino también a las expectativas de los usuarios. Los participantes del proyecto han adquirido nuevas habilidades y destacan la importancia

de la comunicación y el trabajo en equipo para superar los desafíos. En general, se espera que esta plataforma no solo cumpla con su propósito, sino que también sirva como base para futuras mejoras y evoluciones tecnológicas.

REFERENCIAS

(Fuentes citadas en el texto. Libros, manuales, páginas web... En APA).

KeepCoding. (2024). *Lenguajes de programación backend más usados*. KeepCoding. <https://keepcoding.io/blog/lenguajes-de-programacion-backend-mas-usados/>

Calaverita. (2024). *Claves del desarrollo frontend en 2024: Herramientas y estrategias esenciales*. Calaverita. <https://calaverita.dev/es/blog/claves-del-desarrollo-frontend-en-2024-herramientas-y-estrategias-esenciales>

HubSpot. (2023). *Sistemas gestores de bases de datos: Qué son y cómo elegir el adecuado*. HubSpot. <https://blog.hubspot.es/sales/sistemas-gestores-bases-de-datos>