

PROYECTO “BARBER CONnECT”



**Carlos Felipe Muñoz Alvarez**

**Ángel Alejandro Oscategui Pineda**

**Damián Tomás Salazar Uribe**

**Ingeniería en Informática**

**Sección: PTY4614-004V**

**Docente: Gonzalo Alejandro Paredes Recabarren**

**Sede Plaza Oeste**

**21 de noviembre 2024**

PRESENTACIÓN FINAL CAPSTONE

# Índice

Tabla de contenido

[Índice 2](#_Toc183295868)

[Resumen/Abstract 5](#_Toc183295869)

[Introducción 6](#_Toc183295870)

[Acta de Constitución del Proyecto 8](#_Toc183295871)

[Información del Proyecto 8](#_Toc183295872)

[Interesados (Stakeholders) 10](#_Toc183295873)

[Planificar la Gestión del Alcance Plan de Gestión del Alcance 11](#_Toc183295874)

[Componentes del Plan de Gestión del Alcance 13](#_Toc183295875)

[Recopilar Requisitos - Documentación de Requisitos 15](#_Toc183295876)

[Definir el Alcance - Declaración del Alcance del Proyecto 17](#_Toc183295877)

[EDT (Estructura de Desglose del Trabajo) 19](#_Toc183295878)

[Planificar la Gestión del Cronograma - Plan de Gestión del Cronograma 21](#_Toc183295879)

[Definir las Actividades - Lista de Actividades 22](#_Toc183295880)

[Secuenciar las Actividades - Diagrama de Secuencia de Actividades 25](#_Toc183295881)

[Estimar los Recursos de las Actividades - Estimación de Recursos 28](#_Toc183295882)

[Estimar la Duración de las Actividades - Estimación de Duración de Actividades 30](#_Toc183295883)

[Desarrollar el Cronograma - Cronograma del Proyecto 31](#_Toc183295884)

[Planificar la Gestión de los Costos - Plan de Gestión de Costos 33](#_Toc183295885)

[Estimar los Costos - Estimación de Costos del Proyecto 34](#_Toc183295886)

[Determinar el Presupuesto - Presupuesto del Proyecto 35](#_Toc183295887)

[Planificar la Gestión de la Calidad - Plan de Gestión de la Calidad 35](#_Toc183295888)

[Planificar la Gestión de los Recursos - Plan de Gestión de Recursos 36](#_Toc183295889)

[Estimar los Recursos de Actividades - Matriz de Asignación de Responsabilidades 37](#_Toc183295890)

[Planificar la Gestión de las Comunicaciones - Plan de Gestión de las Comunicaciones 39](#_Toc183295891)

[Planificar la Gestión de los Riesgos - Plan de Gestión de Riesgos 40](#_Toc183295892)

[Identificar los Riesgos - Registro de Riesgos 41](#_Toc183295893)

[Realizar el Análisis Cualitativo de Riesgos - Análisis Cualitativo de Riesgos 42](#_Toc183295894)

[Realizar el Análisis Cuantitativo de Riesgos - Análisis Cuantitativo de Riesgos 43](#_Toc183295895)

[Planificar la Respuesta a los Riesgos - Estrategias de Respuesta a Riesgos 44](#_Toc183295896)

[Plan de Gestión de las Adquisiciones del Proyecto 45](#_Toc183295897)

[Plan de Gestión de los Interesados del Proyecto 51](#_Toc183295898)

[Registro de Cambios del Proyecto 56](#_Toc183295899)

[Documentación de Lecciones Aprendidas del Proyecto 60](#_Toc183295900)

[Registro de Recursos Adquiridos del Proyecto 65](#_Toc183295901)

[Desarrollar el Equipo - Plan de Desarrollo del Equipo 69](#_Toc183295902)

[Dirigir al Equipo - Evaluación de Desempeño del Equipo 71](#_Toc183295903)

[Gestionar las Comunicaciones - Informe de Estado del Proyecto 72](#_Toc183295904)

[Controlar el Proyecto - Informe de Desempeño 73](#_Toc183295905)

[Gestionar el Riesgo - Plan de Gestión de Riesgos 74](#_Toc183295906)

[Cerrar el Proyecto - Informe de Cierre 75](#_Toc183295907)

[Conclusiones individuales solo en inglés. 79](#_Toc183295908)

[Descripción del aporte del proyecto en el desarrollo de nuestros intereses profesionales 81](#_Toc183295909)

[Evidencias del proyecto 84](#_Toc183295910)

[Arquitectura de Software 84](#_Toc183295911)

[Objetivos y Restricciones de la Arquitectura 88](#_Toc183295912)

[Objetivos de la arquitectura 88](#_Toc183295913)

[Restricciones de la Arquitectura 90](#_Toc183295914)

[Arquitectura Lógica 92](#_Toc183295915)

[Arquitectura de Procesos 96](#_Toc183295916)

[Arquitectura de desarrollo 98](#_Toc183295917)

[Arquitectura física 101](#_Toc183295918)

[Escenarios 102](#_Toc183295919)

[Modelo de Casos de Uso 103](#_Toc183295920)

[Especificación de los Escenarios de Calidad Relevantes 107](#_Toc183295921)

[Tamaño y desempeño 111](#_Toc183295922)

[Identificando Riesgos 113](#_Toc183295923)

[Patrones de Diseño 114](#_Toc183295924)

[¿Y cómo lo podemos aplicar a nuestro código? 115](#_Toc183295925)

[EVALUACION APLICANDO MODEL ATAM: 122](#_Toc183295926)

[Fase 1: 122](#_Toc183295927)

[1.PRESENTACION DE ATAM: 122](#_Toc183295928)

[2.PRESENTACION DE LOS OBJETIVOS DE NEGOCIOS: 122](#_Toc183295929)

[3.PRESENTACION DE LA ARQUITECTURA: 124](#_Toc183295930)

[FASE 2: 128](#_Toc183295931)

[4.IDENTIFICACION DE PATRONES ARQUITECTONICOS: 128](#_Toc183295932)

[5.GENERACION DEL ARBOL DE UTILIDAD: 129](#_Toc183295933)

[6.ANALIZAR LOS ENFOQUES ARQUITECTONICOS 131](#_Toc183295934)

[FASE3: 133](#_Toc183295935)

[7.PRIORIZACION DE ESCENARIOS 133](#_Toc183295936)

[8 ANALISS DE ENFOQUES: 135](#_Toc183295937)

[FASE4: 136](#_Toc183295938)

[9.PRESENTACION DE RESULTADOS: 136](#_Toc183295939)

[Metodología de Modelado 138](#_Toc183295940)

[Diseño del Modelo Entidad-Relación 138](#_Toc183295941)

[Implementación en MySQL 139](#_Toc183295942)

[Definición de Relaciones 140](#_Toc183295943)

[Índices y Optimización 141](#_Toc183295944)

[Resultados y Validación 141](#_Toc183295945)

[Diagrama ER 143](#_Toc183295946)

[Código SQL base de datos 144](#_Toc183295947)

[Reflexión 151](#_Toc183295948)

# Resumen/Abstract

Este proyecto tiene como propósito el desarrollo de una plataforma web para una cadena de barberías en la comuna de Lo Espejo, cuyo objetivo es proporcionar a los clientes una experiencia integral al permitirles realizar reservas de citas, gestionar sus preferencias, visualizar productos en venta y obtener información detallada de cada barbería.

La plataforma incluirá características clave como la visualización de horarios, precios, disponibilidad de barberos, así como la posibilidad de comprar productos para su recogida en tienda. Además, contará con funciones de gestión de reservas, un botón de contacto mediante WhatsApp y un sistema de notificaciones. Este proyecto se basa en tres competencias fundamentales: la gestión de proyectos, según el marco de trabajo PMBOK, el modelado de bases de datos y el diseño de arquitectura de sistemas, que garantizan la eficiencia, escalabilidad y sostenibilidad del sistema en el largo plazo.

A través de la correcta aplicación de estas competencias, se busca no solo cumplir con los requerimientos del cliente, sino también ofrecer una solución que sea capaz de adaptarse al crecimiento de la cadena de barberías y mejorar la experiencia de los usuarios finales.

This project aims to develop a web platform for a chain of barbershops located in the Lo Espejo commune, designed to provide an integrated experience for customers by allowing them to make reservations, manage their preferences, view products for sale, and access detailed information about each barbershop. The platform will feature key functionalities such as displaying schedules, prices, barber availability, and the option to purchase products for in-store pickup. Additionally, it will include reservation management, a WhatsApp contact button, and a notification system. The project focuses on three core competencies: project management, following the PMBOK framework, database modeling, and system architecture design, ensuring the platform's efficiency, scalability, and long-term sustainability. By effectively applying these competencies, the project seeks to not only meet client requirements but also provide a solution that adapts to the growth of the barbershop chain and enhances the end-user experience.

# Introducción

El presente trabajo de investigación tiene como objetivo principal el diseño y desarrollo de una plataforma web para una cadena de barberías en la comuna de Lo Espejo. Esta plataforma permitirá a los usuarios realizar reservas de citas, gestionar sus preferencias, ver productos en venta, y acceder a información detallada sobre los servicios ofrecidos por cada barbería. La plataforma se orientará a optimizar la experiencia del cliente, facilitando la interacción y mejorando la gestión interna de las barberías.

**Contextualización del problema o área de estudio:**

La cadena de barberías se enfrenta a retos relacionados con la gestión de reservas, la atención al cliente, la organización interna y la actualización de la información de manera eficiente. Actualmente, el proceso de reserva es manual y no está centralizado, lo que genera problemas en la coordinación y en la experiencia del cliente. Asimismo, la falta de un sistema integrado impide que la cadena de barberías aproveche de manera óptima las tecnologías disponibles para mejorar su competitividad. Este proyecto busca proporcionar una solución innovadora mediante la implementación de una plataforma web que abarque tanto la interacción con el cliente como la administración de los procesos operativos.

**Justificación de la importancia de la investigación:**

Este estudio es relevante porque responde a la necesidad de modernizar el proceso de gestión de las barberías en la comuna de Lo Espejo, mejorando la calidad del servicio y la satisfacción de los clientes. La implementación de una plataforma web permitirá optimizar la disponibilidad de servicios, los tiempos de espera, y la visibilidad de la oferta, lo que puede traducirse en una mayor fidelización de los clientes y en un aumento de las ventas. Además, la integración de funciones clave como la visualización de horarios, la gestión de reservas, y la compra de productos en línea representa un avance significativo hacia la digitalización de los pequeños negocios, especialmente en un sector tan competitivo como el de las barberías.

**Objetivos de la tesis:**

* Objetivo general: Desarrollar una plataforma web para una cadena de barberías en la comuna de Lo Espejo, que permita gestionar citas, productos, y la información de los servicios ofrecidos de manera eficiente y accesible.
* Objetivos específicos:
  1. Diseñar la arquitectura del sistema para asegurar su escalabilidad y facilidad de uso.
  2. Modelar la base de datos para soportar las operaciones de reservas, pagos y gestión de inventarios.
  3. Implementar un sistema de gestión de reservas que permita a los clientes agendar citas y realizar compras de productos en línea.
  4. Integrar un sistema de notificaciones y de contacto mediante WhatsApp para mejorar la interacción con los clientes.
  5. Evaluar el rendimiento de la plataforma mediante pruebas de usabilidad y eficiencia para asegurar una experiencia óptima para los usuarios finales.

# Acta de Constitución del Proyecto

**Título del Proyecto “BarberConnect”**

Diseño de Arquitectura, Gestión de Proyecto y Modelamiento de Base de Datos para una Plataforma Digital de Gestión de Barberías en la Comuna de Lo Espejo.

# Información del Proyecto

|  |  |
| --- | --- |
| **Empresa / Organización** | Connect IT |
| **Proyecto** | BarberConnect |
| **Fecha de preparación** | 14/08/2024 |
| **Cliente** | Cadena de Barberías Comuna Lo Espejo |
| **Patrocinador principal** | Socios de la cadena de barberías de la comuna |
| **Gerente de Proyecto** | Carlos Muñoz |

**Propósito y Justificación**

**Propósito:** Proveer un diseño detallado de arquitectura, un modelo de base de datos, y un plan de gestión para una futura implementación de una plataforma digital que centralice la gestión de citas y servicios de una cadena de barberías en la comuna.

**Justificación:** Aunque muchas barberías ofrecen servicios locales, no cuentan con una solución digital centralizada para mejorar la eficiencia operativa y la experiencia del cliente. Esta tesis sentará las bases en cuanto a gestión, arquitectura y base de datos, facilitando el desarrollo futuro del sistema con especificaciones claras y un modelo de datos bien definido con el fin de que se agilice la ida a las barberías y sea todo claro y ordenado para los clientes para sobre todo actualizar y mejorar el servicio.

**Alcance del Proyecto**

Abarcará Gestión de Proyectos (PMBOK), Arquitectura de Software, y Modelamiento de Base de Datos, excluyendo la parte de desarrollo.

**Productos Entregables de la Tesis:**

**Plan de Gestión de Proyecto:** Documentación completa que incluye cronograma, análisis de interesados, estimación de recursos y planificación de riesgos.

**Diseño de Arquitectura:** Definición de componentes de software, diagramas de flujo y descripción de la interacción entre módulos.

**Modelo de Base de Datos:** Diagramas y una estructura de base de datos para gestionar citas, usuarios, productos, y membresías de las barberías.

**Limitaciones:**

No se abordará la codificación ni la implementación práctica de la plataforma.

Las especificaciones están orientadas a un prototipo teórico, sin pruebas funcionales ni despliegue.

**Objetivos del Proyecto**

**Objetivo General:** Diseñar los elementos de arquitectura, base de datos y gestión necesarios para una plataforma digital que centralice la gestión de barberías.

**Objetivos Específicos:**

* Desarrollar un plan de gestión de proyecto utilizando los lineamientos PMBOK.
* Elaborar una arquitectura de software detallada que permita una futura implementación.
* Diseñar un modelo de base de datos optimizado para la gestión de citas, productos y usuarios en el contexto de una cadena de barberías.

**Requisitos Generales**

Definiciones precisas en cada área de la tesis (gestión, arquitectura y base de datos).

Enfoque en una plataforma accesible desde dispositivos móviles y de escritorio en el diseño arquitectónico.

Diseño de base de datos relacional escalable.

**Cronograma General**

Fase de Planificación de Proyecto: 2 semanas

Diseño de Arquitectura: 4 semanas

Modelado de Base de Datos: 4 semanas

Revisión y Ajustes en la Documentación: 2 semanas

# Interesados (Stakeholders)

**Propietarios y Administradores de Barberías:** Beneficiarios indirectos, cuyo negocio podría optimizarse a partir de esta plataforma en un futuro mejorando la calidad del servicio para así aumentar las ventas.

**Equipo de Tesis:**

Ángel Oscategui

Damián Salazar

Carlos Muñoz

**Futuro Equipo de Desarrollo:** Aquellos encargados de la implementación práctica de la plataforma digital.

# Planificar la Gestión del Alcance Plan de Gestión del Alcance

El Plan de Gestión del Alcance es un documento fundamental en la gestión de proyectos, ya que define los procesos y herramientas necesarios para asegurar que el proyecto incluya todo el trabajo requerido (y únicamente el trabajo requerido) para completar el proyecto de manera exitosa. En el caso de este proyecto, el propósito es establecer una planificación rigurosa y estructurada para definir y controlar el alcance en el desarrollo de una plataforma web para una cadena de barberías. Esta plataforma permitirá a los usuarios gestionar citas, visualizar productos y proporcionar acceso desde dispositivos móviles.

**Objetivo del Plan de Gestión del Alcance**

El objetivo del Plan de Gestión del Alcance es:

1. Documentar y gestionar el trabajo necesario para cumplir con los objetivos del proyecto de una manera organizada y controlada.

2. Evitar la expansión del alcance sin autorización formal (conocido como scope creep), para asegurar que el proyecto se complete dentro del plazo y presupuesto establecidos.

3. Asegurar que todos los entregables cumplen con los requisitos definidos por los interesados y mantienen el estándar de calidad.

**Alcance del Proyecto**

Este plan de gestión se enfoca en tres áreas clave:

- Gestión del Proyecto: Incluye todos los procesos de inicio, planificación, monitoreo y control del proyecto, específicamente en las áreas de alcance, cronograma, y comunicación con los interesados.

- Arquitectura del Sistema: Desarrollo del diseño de alto nivel de la plataforma, incluyendo diagramas de arquitectura, selección de tecnologías, y estructuración de componentes.

- Modelado de la Base de Datos: Diseño de un modelo de datos que soporte todas las funcionalidades del sistema, asegurando la consistencia y escalabilidad de la base de datos para futuras ampliaciones.

# Componentes del Plan de Gestión del Alcance

**Definición del Alcance del Proyecto**

- El alcance está delimitado exclusivamente a la planificación y diseño de la plataforma. Esto significa que se documentarán todos los componentes necesarios para llevar a cabo el proyecto, incluyendo la definición de requisitos y la creación de una arquitectura de sistema robusta. El desarrollo del software (implementación de código) no es parte de este alcance.

**Criterios de Aceptación del Alcance**

- Para que el alcance sea considerado completo, deben cumplirse los siguientes criterios:

- Entrega y aprobación de los documentos de planificación y arquitectura.

- Revisión formal de los entregables con los interesados, donde se evalúa el cumplimiento de los objetivos y se aprueban los componentes definidos.

- La base de datos diseñada debe satisfacer las necesidades del sistema en términos de integridad, eficiencia y escalabilidad.

**Roles y Responsabilidades**

- Gerente del Proyecto: Responsable de supervisar que el alcance se mantenga dentro de los límites establecidos, controlar los cambios y gestionar las expectativas de los interesados.

- Equipo de Arquitectura: Desarrolla y documenta el diseño de la arquitectura, selecciona las tecnologías apropiadas y asegura que el diseño sea escalable y adecuado para futuras implementaciones.

- Equipo de Modelado de Datos: Encargado de diseñar y documentar el modelo de base de datos, el cual soportará todas las funcionalidades de la plataforma, desde la gestión de usuarios hasta el inventario de productos.

**Proceso de Creación de la EDT (Estructura de Desglose del Trabajo)**

- La creación de la EDT permite dividir el proyecto en elementos manejables y estructurados. Para este proyecto, la EDT incluirá:

- Fase de Planificación del Proyecto: Documentos de alcance, cronograma y comunicación.

- Fase de Diseño de Arquitectura: Diagramas de componentes, elección de tecnologías, y estructura de servidores.

- Fase de Modelado de Base de Datos: Diseño de entidad-relación, normalización de datos, y relaciones entre entidades.

**Proceso de Control del Alcance**

- Cualquier cambio en el alcance debe ser documentado y aprobado por el gerente de proyecto y registrado formalmente en un log de cambios.

- Las solicitudes de cambio serán evaluadas en cuanto a su impacto en el tiempo, costo y calidad del proyecto, asegurando que solo se implementen cambios que aporten valor al proyecto.

**Herramientas y Técnicas para la Gestión del Alcance**

- Análisis de Requisitos: Se realizarán entrevistas y reuniones con los interesados para identificar claramente las necesidades del proyecto.

- Reuniones de Revisión: Se llevarán a cabo reuniones periódicas con los interesados para evaluar el progreso, discutir posibles ajustes y asegurar que el alcance siga alineado con sus expectativas.

- EDT y Diagrama de Gantt: Estas herramientas se emplearán para estructurar y visualizar las tareas y el cronograma del proyecto, facilitando la asignación de recursos y el control de tiempos.

# Recopilar Requisitos - Documentación de Requisitos

**Requisitos Funcionales**

**1. Gestión de Citas**

- Los clientes deben poder ver la disponibilidad de citas en tiempo real.

- El sistema debe permitir a los clientes crear, modificar y cancelar sus citas a través de la plataforma.

- La plataforma debe enviar notificaciones de recordatorio antes de la cita.

**2. Visualización de Productos**

- La plataforma debe mostrar el inventario de productos que están a la venta en las barberías, con detalles de cada producto (nombre, descripción, precio, disponibilidad).

- El sistema debe permitir la actualización automática del inventario, reflejando la disponibilidad en tiempo real.

**3. Perfil de Usuario**

- Los usuarios deben poder registrarse y crear un perfil que contenga información personal y su historial de citas.

- Cada perfil de usuario debe estar protegido y solo accesible con credenciales válidas.

**4. Notificaciones de Marketing**

- La plataforma debe tener la capacidad de enviar notificaciones a los usuarios sobre promociones, eventos o productos nuevos.

**Requisitos No Funcionales**

**1. Rendimiento**

- La plataforma debe cargar en menos de 3 segundos en dispositivos móviles y soportar un mínimo de 1000 usuarios simultáneos.

**2. Seguridad**

- Debe emplearse encriptación para proteger los datos de los usuarios, cumpliendo con normativas como el GDPR en caso de que aplique.

**3. Compatibilidad**

- La plataforma debe ser accesible desde dispositivos móviles, tabletas y computadoras de escritorio.

**4. Escalabilidad**

- El diseño del sistema debe permitir la expansión para incluir más barberías y servicios, soportando tanto el crecimiento en usuarios como en datos almacenados.

# Definir el Alcance - Declaración del Alcance del Proyecto

**Nombre del Proyecto: BarberConnect**

Desarrollo de Planificación para el Sistema de Gestión de Barberías

**Propósito del Proyecto**

El propósito de este proyecto es diseñar una plataforma web para una cadena de barberías, permitiendo la gestión de citas, visualización de productos y acceso en dispositivos móviles.

**Descripción del Alcance**

**Este proyecto incluye:**

- Gestión del Proyecto: Definición y documentación de cronograma, roles y procesos necesarios para asegurar el éxito del proyecto.

- Arquitectura del Sistema: Diseño de una arquitectura escalable y eficiente, incluyendo el esquema de componentes y las tecnologías a utilizar.

- Modelado de Base de Datos: Creación de un modelo de datos que permita almacenar y gestionar información de clientes, citas y productos.

**Entregables Principales**

1. Documentación completa de la gestión del proyecto.

2. Diseño detallado de la arquitectura del sistema.

3. Modelo de base de datos, incluyendo diagramas y esquemas de tablas.

**Criterios de Aceptación**

- Aprobación de documentos de gestión, arquitectura y base de datos por los interesados.

- Revisión final y aprobación del modelo de datos para garantizar la integridad y eficiencia.

**Exclusiones del Proyecto**

- Implementación del código y desarrollo de la plataforma.

- Integraciones externas como pasarelas de pago o redes sociales.

**Limitaciones y Supuestos**

- Limitaciones: Plazo de entrega de tres meses y recursos limitados.

- Supuestos: Los interesados estarán disponibles para revisiones periódicas y aprobaciones.

# EDT (Estructura de Desglose del Trabajo)

Nosotros, Ángel Oscátegui, Damián Salazar y Carlos Muñoz, creamos la Estructura de Desglose del Trabajo (EDT) para organizar de manera eficiente cada aspecto de nuestro proyecto de creación de una plataforma web para una cadena de barberías. Este desglose nos permite gestionar cada componente del proyecto en secciones manejables, facilitando así la asignación de recursos, el monitoreo del avance, y la identificación de entregables específicos. La EDT es fundamental para descomponer el proyecto en tareas claras que se alinean con los objetivos y expectativas de los interesados.

**Objetivo de la EDT**

El objetivo de la EDT es estructurar el proyecto en tareas que puedan ser ejecutadas de manera secuencial y eficiente, estableciendo un mapa de ruta claro y comprensible. A través de esta estructura, buscamos maximizar la productividad del equipo y facilitar el seguimiento del progreso. La EDT también actúa como una guía para asegurar que no se omita ningún paso necesario para la culminación exitosa del proyecto.

**Componentes de la EDT**

La EDT se desglosa en tres áreas principales, con múltiples niveles de subtareas, que permiten cubrir todos los aspectos del proyecto:

1. **Gestión del Proyecto**:
   * Planificación del alcance y cronograma.
   * Documentación de los requisitos y definición del alcance del proyecto.
   * Seguimiento y control de las actividades.
2. **Arquitectura del Sistema**:
   * Diseño de la arquitectura general.
   * Selección de tecnologías.
   * Definición de la estructura de servidores y componentes.
3. **Modelado de la Base de Datos**:
   * Desarrollo de diagramas de entidad-relación.
   * Normalización de datos.
   * Definición de relaciones entre las entidades y pruebas de integridad.

Al dividir el trabajo en estas áreas, logramos tener una visión más clara de los entregables específicos y el progreso esperado en cada fase. Esta estructura no solo facilita la coordinación interna, sino que también permite a los interesados comprender el estado y los avances del proyecto en cada etapa.

# Planificar la Gestión del Cronograma - Plan de Gestión del Cronograma

Como equipo planificamos la gestión del cronograma con el fin de establecer un flujo de trabajo estructurado que nos permita cumplir con los tiempos estimados y reducir al mínimo el riesgo de retrasos. Un cronograma detallado nos asegura que cada actividad se ejecutará dentro del marco temporal definido y nos proporciona puntos de referencia para evaluar el avance.

**Objetivo del Plan de Gestión del Cronograma**

Nuestro objetivo es completar cada fase del proyecto dentro del tiempo estimado, mediante un cronograma detallado que nos permita anticipar posibles cuellos de botella y gestionar las dependencias entre tareas. Este plan de cronograma es una herramienta crucial para asignar responsabilidades, definir plazos y coordinar actividades.

**Detalles del Cronograma**

El cronograma del proyecto está estructurado en las siguientes fases:

1. **Inicio y Planificación del Proyecto**:
   * Revisión de los requisitos y definición del alcance.
   * Duración estimada: 1 semana.
2. **Diseño de Arquitectura**:
   * Estructuración de la arquitectura del sistema y elección de tecnologías.
   * Duración estimada: 2 semanas.
3. **Modelado de la Base de Datos**:
   * Desarrollo de esquemas de datos y relaciones.
   * Duración estimada: 1 semana.
4. **Revisión y Validación**:
   * Revisión de la arquitectura y modelo de datos, con ajustes según las observaciones de los interesados.
   * Duración estimada: 3 días.
5. **Documentación Final y Entrega**:
   * Documentación completa del proyecto y presentación a los interesados.
   * Duración estimada: 1 semana.

# Definir las Actividades - Lista de Actividades

Para asegurar la correcta ejecución del proyecto, definimos una lista detallada de actividades que abarca cada paso requerido para completar la plataforma web de manera exitosa. La lista de actividades no solo describe cada tarea, sino que también detalla el tiempo estimado, los responsables, y las relaciones de dependencia, lo cual mejora la coordinación y permite una gestión precisa de los recursos.

**Lista de Actividades**

1. **Revisión de Requisitos**
   * **Duración**: 1 semana
   * **Responsables**: Todo el equipo
   * **Descripción**: Analizamos los requisitos funcionales y no funcionales junto a los interesados para definir claramente los objetivos del proyecto y sus límites. Esto incluye reuniones y revisión de documentos proporcionados por los interesados.
2. **Desarrollo del Plan de Gestión del Alcance**
   * **Duración**: 3 días
   * **Responsable**: Ángel Oscátegui
   * **Descripción**: Creamos el plan que delimita el alcance del proyecto, asegurándonos de que se alineen las expectativas del cliente con nuestras capacidades y los recursos disponibles.
3. **Creación de la Estructura de Desglose del Trabajo (EDT)**
   * **Duración**: 4 días
   * **Responsables**: Damián Salazar y Carlos Muñoz
   * **Descripción**: Dividimos el proyecto en tareas específicas que faciliten la asignación de recursos y el seguimiento del avance. Documentamos cada componente para su comprensión y seguimiento.
4. **Desarrollo del Plan de Gestión del Cronograma**
   * **Duración**: 3 días
   * **Responsables**: Todo el equipo
   * **Descripción**: Establecemos el cronograma general del proyecto, determinando las dependencias entre actividades y asignando tiempos específicos a cada tarea.
5. **Diseño de la Arquitectura del Sistema**
   * **Duración**: 2 semanas
   * **Responsable**: Carlos Muñoz
   * **Descripción**: Definimos la estructura de la plataforma, seleccionando tecnologías y estableciendo la disposición de componentes y servidores. Creamos diagramas de arquitectura para una visualización clara de la estructura del sistema.
6. **Modelado de la Base de Datos**
   * **Duración**: 1 semana
   * **Responsable**: Ángel Oscátegui
   * **Descripción**: Diseñamos el modelo de base de datos, incluyendo entidades, relaciones, y normalización. Desarrollamos el esquema que soporte las funcionalidades de la plataforma y asegura la integridad de los datos.
7. **Revisión y Ajustes de la Arquitectura y Modelo de Datos**
   * **Duración**: 3 días
   * **Responsable**: Damián Salazar
   * **Descripción**: Realizamos una revisión detallada de los diseños de arquitectura y base de datos, haciendo ajustes según los comentarios recibidos por parte de los interesados o los cambios de requisitos.
8. **Elaboración del Documento Final de Planificación y Diseño**
   * **Duración**: 1 semana
   * **Responsables**: Todo el equipo
   * **Descripción**: Compilamos toda la documentación referente a la planificación y diseño, incluyendo el plan de alcance, cronograma, arquitectura y modelo de datos. Preparamos el documento final para su presentación formal a los interesados.
9. **Revisión Final del Proyecto y Entrega**
   * **Duración**: 2 días
   * **Responsables**: Todo el equipo
   * **Descripción**: Hacemos una última revisión del proyecto para verificar que cumple con todos los requisitos definidos y está listo para su entrega a los interesados. Nos aseguramos de que todos los documentos y archivos estén completos y en orden.

# Secuenciar las Actividades - Diagrama de Secuencia de Actividades

La secuenciación de actividades es una etapa esencial en la gestión de proyectos según las buenas prácticas del PMBOK, ya que permite identificar el orden adecuado de ejecución de cada tarea y la relación de dependencia entre ellas. Este proceso es crucial para garantizar que el flujo de trabajo sea eficiente, lógico y bien estructurado, evitando cuellos de botella o tiempos de espera innecesarios. Para nuestro proyecto de creación de una plataforma web para una cadena de barberías, en el cual participamos Ángel Oscátegui, Damián Salazar y Carlos Muñoz, hemos establecido una secuencia de actividades que maximiza la eficiencia del trabajo y se alinea con los tiempos y recursos definidos. El proyecto comenzó la segunda semana de agosto y tiene como fecha de entrega la primera semana de diciembre, lo cual nos da aproximadamente cuatro meses para su ejecución.

**Objetivo del Diagrama de Secuencia de Actividades**

El objetivo principal del diagrama de secuencia de actividades es representar de manera visual y estructurada el orden y la interdependencia de cada tarea dentro del proyecto. Este diagrama sirve como una guía para el equipo y los interesados, permitiendo que todos los participantes comprendan el flujo de trabajo. A través de la secuenciación de actividades, buscamos identificar las rutas críticas del proyecto, que son aquellas actividades que deben completarse en un tiempo específico para evitar retrasos en la fecha de finalización prevista. Además, el diagrama permite prever posibles puntos de conflicto o tareas que puedan requerir ajustes de cronograma y recursos adicionales, mejorando así la capacidad de respuesta del equipo ante contingencias.

**Detalles de la Secuencia de Actividades**

Para nuestro proyecto, hemos desarrollado una estructura de fases que optimiza la continuidad de las tareas y la eficiencia del equipo. Las actividades han sido organizadas de la siguiente manera:

**1. Inicio y Planificación del Proyecto**

- Revisión de los requisitos: Del 12 al 18 de agosto. Durante esta fase inicial, analizamos los requisitos funcionales y no funcionales en colaboración con los interesados. Esta tarea es clave para definir los objetivos específicos y establecer el alcance general del proyecto.

- Definición del alcance: Del 19 al 23 de agosto. En esta actividad, desarrollamos el documento de alcance, alineando las expectativas del cliente con las capacidades y recursos del equipo. Este documento es fundamental para evitar cambios de alcance durante la ejecución.

- Creación de la Estructura de Desglose del Trabajo (EDT): Del 24 al 28 de agosto. La EDT es la herramienta clave que descompone el proyecto en tareas específicas, facilitando la asignación de recursos y el seguimiento de avances.

**2. Diseño de Arquitectura y Modelado de la Base de Datos**

- Diseño de la arquitectura general: Del 1 al 14 de septiembre. En esta fase, Carlos Muñoz define la estructura de la plataforma, seleccionando tecnologías adecuadas y estableciendo la disposición de los componentes y servidores.

- Modelado de la base de datos: Del 15 al 21 de septiembre. Damián Salazar se encarga de diseñar el modelo de datos, incluyendo la normalización de datos, la definición de entidades y relaciones para asegurar la integridad y eficiencia del sistema.

**3. Revisión y Ajustes**

- Revisión de la arquitectura y modelo de datos: Del 22 al 25 de septiembre. Todo el equipo realiza una revisión detallada de los diseños de arquitectura y base de datos, implementando ajustes según los comentarios y necesidades del cliente.

**4. Documentación y Entrega**

- Documentación final del proyecto: Del 1 al 20 de Noviembre. En esta fase, todo el equipo recopila la documentación del proyecto, incluyendo el alcance, el cronograma, la arquitectura y el modelo de datos.

- Revisión final del proyecto y entrega: 5 de diciembre. Se realiza una última revisión para asegurar que todo esté conforme a los requisitos y se entrega el proyecto a los interesados.

# Estimar los Recursos de las Actividades - Estimación de Recursos

La estimación de recursos es una etapa clave en la planificación de proyectos según el PMBOK, pues permite identificar y asignar los recursos humanos, materiales y tecnológicos necesarios para ejecutar cada actividad del proyecto. En nuestro caso, es vital para asegurar que cada tarea pueda ser completada eficientemente, dentro de los tiempos y con los niveles de calidad esperados. Dado que el proyecto cuenta con un equipo compuesto por Ángel Oscátegui, Damián Salazar y Carlos Muñoz, cada miembro asumirá roles específicos según sus habilidades, lo cual facilita una asignación eficiente de recursos.

**Objetivo de la Estimación de Recursos**

El objetivo de la estimación de recursos es garantizar que cada actividad cuente con los elementos necesarios para ser ejecutada de forma óptima y eficiente. La correcta estimación de los recursos permite no solo anticipar la cantidad y el tipo de recursos necesarios para cada tarea, sino también prever posibles escaseces o cuellos de botella. Esto es esencial para que el equipo pueda abordar el proyecto de manera organizada y evitar retrasos debido a la falta de recursos.

**Detalles de la Estimación de Recursos**

1. Inicio y Planificación del Proyecto

- Recursos Humanos: Todo el equipo (Ángel, Damián y Carlos) participará en esta fase inicial para asegurar un análisis integral de los requisitos y un alcance bien definido.

- Materiales y Tecnología: Software de gestión de proyectos (Trello, Microsoft Project), herramientas de documentación (Google Docs, Microsoft Word).

**2. Diseño de Arquitectura**

- Recursos Humanos: Carlos Muñoz liderará esta fase con el apoyo del equipo.

- Materiales y Tecnología: Herramientas de modelado arquitectónico como Lucidchart o Visio para los diagramas de arquitectura.

**3. Modelado de la Base de Datos**

- Recursos Humanos: Damián Salazar, con asistencia de Ángel.

- Materiales y Tecnología: Software de modelado de bases de datos (MySQL Workbench, SQL Server Management Studio) para la creación y normalización de esquemas de datos.

**4. Documentación y Entrega**

- Recursos Humanos: Todo el equipo.

- Materiales y Tecnología: Software de edición de documentos y herramientas de revisión de calidad, como Grammarly y Google Docs, para garantizar la calidad y coherencia en la documentación final.

# Estimar la Duración de las Actividades - Estimación de Duración de Actividades

La estimación de la duración de las actividades es una práctica fundamental en la gestión de proyectos, según los lineamientos del PMBOK, que permite establecer el tiempo requerido para completar cada tarea. Este proceso es crucial para la elaboración del cronograma y garantiza que cada fase se complete en el tiempo estimado, evitando retrasos y optimizando la productividad del equipo. Nuestra estimación de duración toma en cuenta la complejidad de las tareas, los recursos disponibles y la fecha límite establecida para la entrega del proyecto en diciembre.

Detalles de la Duración de las Actividades

1. Inicio y Planificación del Proyecto

- Duración estimada: 2 semanas (del 12 al 28 de agosto).

2. Diseño de Arquitectura

- Duración estimada: 2 semanas (del 1 al 14 de septiembre).

3. Modelado de la Base de Datos

- Duración estimada: 1 semana (del 15 al 21 de septiembre).

4. Revisión y Ajustes de la Arquitectura y Modelo de Datos

- Duración estimada: 3 días (del 22 al 25 de septiembre).

5. Documentación Final y Entrega

- Duración estimada: 1 semana

- Revisión final y entrega: 2 días (4 y 5 de diciembre).

La estimación de duración permite a nuestro equipo estructurar el cronograma de manera precisa, garantizando la disponibilidad de recursos en cada fase y facilitando la finalización del proyecto en el tiempo estipulado.

# Desarrollar el Cronograma - Cronograma del Proyecto

Desarrollar el cronograma es crucial para organizar y supervisar la ejecución de cada tarea dentro de un marco temporal claro. En este proyecto estudiantil, hemos planificado un cronograma con recursos limitados que abarca desde la segunda semana de agosto hasta la primera semana de diciembre. Este cronograma se ha diseñado para optimizar la eficiencia y cumplir con los plazos de entrega sin incurrir en gastos innecesarios.

Detalles del Cronograma

1. Inicio y Planificación del Proyecto

- Fecha de inicio: Segunda semana de agosto

- Fecha de finalización: 28 de agosto

- Actividades principales: Análisis de requisitos, definición del alcance y creación de la EDT.

2. Diseño de Arquitectura y Modelado de la Base de Datos

- Fecha de inicio: 1 de septiembre

- Fecha de finalización: 21 de septiembre

- Actividades principales: Diseño de arquitectura (liderado por Carlos Muñoz) y modelado de base de datos (liderado por Damián Salazar).

3. Revisión y Ajustes

- Fecha de inicio: 22 de septiembre

- Fecha de finalización: 25 de septiembre

- Actividades principales: Revisión de la arquitectura y modelo de datos, implementación de ajustes.

4. Desarrollo del Sitio Web

- Fecha de inicio: 1 de octubre

- Fecha de finalización: 30 de noviembre

- Actividades principales: Desarrollo del frontend y backend, integración de funcionalidades.

5. Documentación Final y Entrega

- Fecha de inicio: 1 de diciembre

- Fecha de finalización: 5 de diciembre

- Actividades principales: Documentación del proyecto, revisión final y entrega.

# Planificar la Gestión de los Costos - Plan de Gestión de Costos

La planificación de la gestión de costos es esencial para el éxito de un proyecto estudiantil con presupuesto limitado. Este plan establece los procedimientos necesarios para estimar, asignar y monitorear los costos, enfocándonos en minimizar gastos mediante la utilización de recursos gratuitos o de bajo costo, cuando sea posible.

El objetivo es controlar y optimizar el gasto en cada actividad, evitando sobrecostos y garantizando la viabilidad del proyecto dentro de los recursos limitados. Se priorizarán herramientas y plataformas de acceso gratuito o que estén disponibles en las instituciones educativas.

Componentes del Plan de Gestión de Costos

1. Estimación de Costos: Se utilizarán herramientas y plataformas gratuitas siempre que sea posible. Las actividades de estimación se basarán en precios accesibles o en servicios educativos.

2. Aprobación de Costos: Los costos estimados deberán ser aprobados por el equipo, asegurando que cada gasto sea necesario y justificado.

3. Control de Costos: Se implementará un sistema de control mediante hojas de cálculo compartidas, permitiendo un seguimiento detallado de los gastos.

4. Reporte de Costos: Se generarán reportes mensuales para revisar el gasto acumulado y ajustar el presupuesto en caso de desviaciones.

# Estimar los Costos - Estimación de Costos del Proyecto

La estimación de costos permite determinar el presupuesto necesario para completar cada tarea. En este proyecto, la estimación se basa en una selección de herramientas accesibles y en el uso de recursos gratuitos para reducir el gasto al mínimo.

Objetivo de la Estimación de Costos

El objetivo de esta estimación es establecer un presupuesto asequible para cada actividad, ajustado a las limitaciones económicas de estudiantes, y utilizando herramientas de bajo costo.

Detalles de la Estimación de Costos

1. Recursos Humanos: El trabajo de cada integrante del equipo no tendrá un costo monetario, ya que todos contribuyen como parte del proyecto académico.

2. Materiales y Tecnología: Se estima un presupuesto mínimo para materiales y tecnología, considerando herramientas de bajo costo:

- Licencias de software: $50,000 CLP (se buscarán descuentos o licencias estudiantiles).

- Servicios de hosting y dominio para el sitio web: $25,000 CLP (se usará un plan económico).

3. Contingencia: Se reservará un 10% adicional del costo total para imprevistos, alrededor de $7,500 CLP.

Presupuesto Total Estimado: $82,500 CLP

# Determinar el Presupuesto - Presupuesto del Proyecto

Componentes del Presupuesto

1. Costos de Materiales y Tecnología: $75,000 CLP

2. Contingencia (10%): $7,500 CLP

3. Presupuesto Total: $82,500 CLP

Este presupuesto ha sido diseñado para cubrir las necesidades esenciales del proyecto sin exceder los recursos disponibles.

# Planificar la Gestión de la Calidad - Plan de Gestión de la Calidad

El plan de gestión de la calidad establece los procedimientos necesarios para asegurar que el proyecto cumpla con los requisitos de calidad. En un contexto de recursos limitados, se priorizan prácticas que aseguren la funcionalidad y facilidad de uso del producto final.

El objetivo es garantizar que el producto final cumpla con los requisitos mínimos de calidad, optimizando los recursos sin comprometer el rendimiento.

Componentes del Plan de Gestión de la Calidad

1. Criterios de Calidad: Cumplimiento de los requisitos funcionales básicos y estabilidad del sistema.

2. Métricas de Calidad: Pruebas de carga y de estabilidad en entornos de bajo recurso.

3. Control de Calidad: Se realizarán pruebas periódicas y revisiones internas para garantizar la calidad.

# Planificar la Gestión de los Recursos - Plan de Gestión de Recursos

Componentes del Plan de Gestión de Recursos

1. Asignación de Recursos Humanos: Cada miembro del equipo tiene un rol específico para maximizar el uso de sus habilidades y minimizar gastos adicionales.

2. Disponibilidad y Adquisición de Materiales: Se buscarán herramientas gratuitas o con licencias educativas.

3. Seguimiento de Recursos: Se implementará un seguimiento básico mediante una hoja de cálculo para monitorear el uso de los recursos en tiempo real.

| Actividad | Recurso Humano Asignado | Tipo de Recurso | Cantidad de Recursos | Duración Estimada | Comentarios |
| --- | --- | --- | --- | --- | --- |
| Investigación de requisitos | Analista de sistemas | Humano | 1 | 2 semanas | Necesario para definir el alcance del proyecto |
| Diseño de arquitectura | Arquitecto de software | Humano | 1 | 3 semanas | Crear la arquitectura inicial para el sistema |
| Desarrollo del backend | Desarrollador backend | Humano, Hardware | 2 | 8 semanas | Desarrollar y probar el backend del sistema |
| Pruebas y validación | Ingeniero de calidad | Humano | 1 | 4 semanas | Validar y asegurar la calidad del sistema |
| Implementación de servidor | Administrador de sistemas | Humano, Hardware | 1 | 1 semana | Configurar el servidor para producción |

# Estimar los Recursos de Actividades - Matriz de Asignación de Responsabilidades

**Matriz de Asignación de Responsabilidades (RAM)**

| **Actividad** | **Responsable (R)** | **Aprobador (A)** | **Consultado (C)** | **Informado (I)** |
| --- | --- | --- | --- | --- |
| **1. Análisis de requisitos** | Carlos Muñoz | Carlos Muñoz | Propietarios de barberías | Equipo de desarrollo |
| **2. Diseño de la arquitectura del sistema** | Ángel Oscátegui | Carlos Muñoz | Equipo de desarrollo | Propietarios de barberías |
| **3. Modelado de la base de datos** | Damián Salazar | Carlos Muñoz | Ángel Oscátegui | Propietarios de barberías |
| **4. Desarrollo del sistema de gestión de reservas** | Equipo de desarrollo | Carlos Muñoz | Propietarios de barberías | Usuarios finales |
| **5. Integración de Google Calendar** | Ángel Oscátegui | Carlos Muñoz | Proveedores de tecnología | Usuarios finales |
| **6. Implementación de página de administración** | Damián Salazar | Carlos Muñoz | Equipo de desarrollo | Propietarios de barberías |
| **7. Pruebas de funcionalidad** | Equipo de desarrollo | Carlos Muñoz | Usuarios finales | Propietarios de barberías |
| **8. Corrección de errores** | Equipo de desarrollo | Carlos Muñoz | Propietarios de barberías | Usuarios finales |
| **9. Entrenamiento a usuarios** | Carlos Muñoz | Carlos Muñoz | Usuarios finales | Propietarios de barberías |
| **10. Cierre del proyecto** | Carlos Muñoz | Carlos Muñoz | Propietarios de barberías | Equipo de desarrollo |

* **R (Responsable):** Persona que ejecuta la actividad o tarea. Es quien toma las acciones para completarla.
* **A (Aprobador):** Persona encargada de revisar y aprobar los resultados del trabajo.
* **C (Consultado):** Persona que proporciona información, asesoramiento o experiencia para realizar la tarea.
* **I (Informado):** Persona que debe ser notificada sobre el progreso o los resultados, pero no participa directamente en la ejecución.

# Planificar la Gestión de las Comunicaciones - Plan de Gestión de las Comunicaciones

| **Tipo de Comunicación** | **Frecuencia** | **Participantes** | **Medio** | **Responsable** | **Objetivo** |
| --- | --- | --- | --- | --- | --- |
| Reunión de seguimiento semanal | Semanal | Equipo de proyecto | Presencial / Video | Líder de proyecto | Revisar el avance y resolver dudas |
| Reporte de estado | Quincenal | Cliente, Stakeholders | Documento / Email | Líder de proyecto | Informar del progreso del proyecto |
| Actualización técnica | Mensual | Equipo técnico | Documento compartido | Arquitecto de software | Reportar cambios técnicos o problemas |
| Revisión de hitos | Al completar hitos | Equipo, Cliente | Presencial / Video | Líder de proyecto | Validar el cumplimiento de los hitos |
| Comunicación de riesgos | Según necesidad | Equipo de riesgos | Documento / Email | Gestor de riesgos | Informar sobre nuevos riesgos y planes de respuesta |

**Explicación:**

El plan de gestión de las comunicaciones establece cómo, cuándo y con quién se deben realizar las comunicaciones. Esto garantiza que todos los involucrados estén al tanto del progreso y puedan colaborar efectivamente. Este plan define la estructura de las comunicaciones en función de la frecuencia, los participantes, el medio, y el objetivo de cada tipo de comunicación.

# Planificar la Gestión de los Riesgos - Plan de Gestión de Riesgos

| **Componente del Plan de Gestión de Riesgos** | **Descripción** |
| --- | --- |
| Identificación de riesgos | Procedimientos para identificar riesgos potenciales en las fases de desarrollo y operación |
| Análisis cualitativo | Evaluación de la probabilidad e impacto de cada riesgo para priorizar los más relevantes |
| Análisis cuantitativo | Cálculo del impacto financiero y temporal de los riesgos identificados |
| Planificación de respuestas | Estrategias para mitigar, transferir, aceptar o evitar los riesgos según su naturaleza |
| Seguimiento y control | Procesos para monitorizar riesgos durante el proyecto y ajustar estrategias cuando sea necesario |

**Explicación:**

Este plan detalla cómo el equipo manejará los riesgos durante el proyecto. Involucra la identificación y evaluación de riesgos, la planificación de respuestas y el monitoreo continuo para minimizar el impacto de los riesgos en el proyecto.

# Identificar los Riesgos - Registro de Riesgos

| **Identificación** | **Riesgo** | **Descripción** | **Probabilidad** | **Impacto** | **Estrategia de Mitigación** |
| --- | --- | --- | --- | --- | --- |
| R1 | Atrasos en el desarrollo | *Problemas técnicos con integración* | *Alta* | *Alta* | Crear un plan de contingencia y pruebas intermedias |
| R2 | Falta de recursos | *Recursos humanos o materiales insuficientes* | *Media* | *Alta* | Asegurar disponibilidad de recursos antes de cada fase |
| R3 | Riesgo financiero | *Exceso de costos en recursos adicionales* | *Baja* | *Media* | Definir un presupuesto con margen de reserva |
| R4 | Problemas de comunicación | *Retraso en la transmisión de información* | *Media* | *Alta* | Implementar reuniones y reportes frecuentes |
| R5 | Cambios en los requisitos | Cambios solicitados por el cliente | Alta | Media | Definir un proceso de gestión de cambios |

**Explicación:**

El registro de riesgos contiene todos los riesgos identificados, su descripción, probabilidad e impacto, junto con las estrategias de mitigación planificadas. Esto permite al equipo prepararse para responder a posibles problemas de manera proactiva.

# Realizar el Análisis Cualitativo de Riesgos - Análisis Cualitativo de Riesgos

| **Riesgo** | **Probabilidad** | **Impacto** | **Nivel de Riesgo** | **Justificación** |
| --- | --- | --- | --- | --- |
| Atrasos en el desarrollo | Alta | Alta | Crítico | Problemas técnicos recurrentes |
| Falta de recursos | Media | Alta | Alto | Dependencia de recursos externos |
| Riesgo financiero | Baja | Media | Medio | Control de costos moderado |
| Problemas de comunicación | Media | Alta | Alto | Información crucial se puede retrasar |
| Cambios en requisitos | Alta | Media | Alto | Cambios frecuentes en el alcance del proyecto |

**Explicación:**

El análisis cualitativo de riesgos clasifica los riesgos en función de su probabilidad e impacto, y les asigna un nivel de riesgo. Este análisis permite priorizar los riesgos y enfocar los esfuerzos en los de nivel más crítico.

# Realizar el Análisis Cuantitativo de Riesgos - Análisis Cuantitativo de Riesgos

| **Riesgo** | **Impacto Financiero Estimado** | **Impacto Temporal Estimado** | **Estrategia** |
| --- | --- | --- | --- |
| Atrasos en desarrollo | $10,000 | 2 semanas | Mitigación |
| Falta de recursos | $5,000 | 1 semana | Mitigación |
| Riesgo financiero | $3,000 | Ninguno | Aceptación |
| Problemas de comunicación | $1,500 | 3 días | Mitigación |
| Cambios en requisitos | $7,000 | 1 semana | Mitigación y negociación |

**Explicación:**

El análisis cuantitativo de riesgos estima el impacto financiero y temporal de cada riesgo, ayudando a decidir cuánto se debería invertir en estrategias de mitigación. Este análisis es útil para definir el presupuesto de reserva y calcular el costo total del proyecto considerando los riesgos.

# Planificar la Respuesta a los Riesgos - Estrategias de Respuesta a Riesgos

| **Riesgo** | **Estrategia** | **Acción Específica** |
| --- | --- | --- |
| Atrasos en el desarrollo | Mitigación | Asignar recursos adicionales y establecer puntos de control |
| Falta de recursos | Mitigación | Coordinar con proveedores y asegurar stock con antelación |
| Riesgo financiero | Aceptación | No se tomarán medidas adicionales |
| Problemas de comunicación | Mitigación | Implementar un sistema de gestión de comunicación eficaz |
| Cambios en requisitos | Mitigación | Crear un proceso formal para evaluar y aprobar cambios |

**Explicación:**

En esta tabla, cada riesgo tiene una estrategia de respuesta específica, con acciones detalladas para reducir su impacto o aceptarlo si el costo de mitigación supera el beneficio. Esto permite manejar los riesgos de manera proactiva y controlada durante el proyecto.

# Plan de Gestión de las Adquisiciones del Proyecto

**1. Identificación de las Necesidades de Adquisición**

**Objetivo:** Definir los bienes, servicios y recursos necesarios para el éxito del proyecto.

En este proyecto, es necesario adquirir los siguientes recursos para desarrollar y mantener el sitio web de la cadena de barbershops de la comuna de Lo Espejo:

* **Licencias de Software:**
  + Licencias para las herramientas de desarrollo y plataformas como Django, Python, y cualquier software adicional necesario para la administración del sitio web.
* **Servicios de Hosting:**
  + Proveedor de servicios de alojamiento web, incluyendo el servidor para la base de datos, plataformas para el sitio web y almacenamiento.
* **Herramientas de Desarrollo:**
  + Herramientas de análisis de datos, seguridad, y monitoreo del sitio.
* **Recursos Humanos:**
  + Desarrolladores adicionales para completar el trabajo en los plazos establecidos.

**Ejemplo de Necesidades de Adquisición:**

| **Recurso** | **Descripción** | **Cantidad Estimada** |
| --- | --- | --- |
| Licencia de Django | Licencia para la plataforma Django | 1 |
| Servidor de Hosting | Servicio de alojamiento web | 1 |
| Licencia de Seguridad | Licencia para herramientas de seguridad (SSL, Firewall) | 1 |
| Herramienta de Análisis | Plataforma de análisis de datos (Google Analytics, herramientas de SEO) | 1 |
| Desarrolladores Adicionales | Recursos humanos adicionales | 1 |

**2. Estrategia de Adquisición**

**Objetivo:** Establecer cómo se adquirirán los bienes y servicios identificados en la fase anterior.

* **Licencias de Software:**
  + **Estrategia:** Compra directa mediante proveedores autorizados para asegurar la legalidad y actualización continua del software.
* **Servicios de Hosting:**
  + **Estrategia:** Contratar un proveedor de servicios de hosting basado en la relación costo-calidad, con un contrato a largo plazo para asegurar la estabilidad del servicio.
* **Herramientas de Desarrollo:**
  + **Estrategia:** Suscripción mensual o anual a herramientas específicas de análisis de datos y seguridad, según las necesidades.
* **Recursos Humanos:**
  + **Estrategia:** Contratación de dos desarrolladores adicionales a través de una agencia especializada para completar el equipo de desarrollo.

**3. Selección de Proveedores**

**Objetivo:** Establecer los criterios para seleccionar proveedores y evaluar sus propuestas.

Para la selección de proveedores se utilizarán los siguientes criterios:

1. **Costo:**
   * Evaluar el costo total de la adquisición en relación con el presupuesto disponible.
2. **Calidad:**
   * Asegurar que los proveedores seleccionados ofrezcan productos y servicios de alta calidad que cumplan con los requisitos del proyecto.
3. **Reputación:**
   * Considerar la experiencia previa y la reputación de los proveedores en el mercado.
4. **Tiempo de Entrega:**
   * Asegurar que el proveedor pueda cumplir con los plazos de entrega establecidos en el proyecto.

**Tabla de Evaluación de Proveedores:**

| **Proveedor** |  | **Tiempo de Entrega** | **Calidad** | **Condiciones** |
| --- | --- | --- | --- | --- |
| **Proveedor A** |  | 15 días | Alta | Plazo flexible |
| **Proveedor B** |  | 10 días | Media | Condiciones estándar |
| **Proveedor C** |  | 12 días | Alta | Garantía incluida |

**4. Gestión de Contratos**

**Objetivo:** Establecer los tipos de contrato que se utilizarán para las adquisiciones, y garantizar que todos los términos sean claros y equitativos.

* **Tipo de Contrato:**
  + Para los servicios de **hosting**, se utilizará un contrato de **precio fijo** por el período acordado.
  + Para las **licencias de software**, se empleará un contrato de **suscripción anual**.
  + Para los **recursos humanos**, se contratarán bajo un contrato de **empleado por tiempo determinado** a través de una agencia externa.

**Términos de los Contratos:**

* **Alcance:** El contrato especificará las tareas y responsabilidades de cada parte.
* **Fechas de entrega:** Los proveedores deberán cumplir con las fechas de entrega acordadas.
* **Condiciones de pago:** Los pagos se realizarán conforme a hitos establecidos en los contratos.

**5. Monitoreo de Adquisiciones**

**Objetivo:** Asegurar que las adquisiciones se realicen de acuerdo con los acuerdos establecidos en los contratos.

* **Monitoreo Regular:**
  + Se establecerán reuniones de seguimiento mensuales con los proveedores para evaluar el cumplimiento de los plazos y la calidad de los servicios entregados.
* **Control de Calidad:**
  + Se realizarán inspecciones periódicas para garantizar que los productos y servicios adquiridos cumplan con los estándares especificados.
* **Control de Plazos:**
  + Se usará un sistema de gestión de proyectos (como **Trello** o **Asana**) para monitorear los avances y asegurar que los plazos se cumplan.

**6. Cierre de Adquisiciones**

**Objetivo:** Finalizar las adquisiciones una vez que los productos o servicios sean entregados y aprobados.

* **Aceptación Formal:**
  + Una vez entregados los productos o servicios, se llevará a cabo una **revisión final** y una **aceptación formal** con los responsables del proyecto para verificar que todo esté conforme al contrato.
* **Documentación:**
  + Se documentarán todas las adquisiciones, indicando si se cumplieron los términos, el costo final y las lecciones aprendidas. Esto ayudará a mejorar el proceso de adquisiciones en proyectos futuros.
* **Lecciones Aprendidas:**
  + Al finalizar la adquisición, se realizará una **sesión de retroalimentación** con los interesados clave para discutir qué funcionó bien, qué podría haberse hecho de manera diferente y cómo mejorar el proceso en el futuro.

# Plan de Gestión de los Interesados del Proyecto

El objetivo de este Plan es identificar a los interesados clave, analizar sus necesidades y expectativas, y desarrollar una estrategia para involucrarlos en el proyecto, garantizando una comunicación efectiva y un manejo adecuado de sus expectativas.

**1. Identificación de los Interesados**

Los interesados son aquellos individuos, grupos u organizaciones que pueden tener un impacto en el proyecto o verse impactados por él. La identificación de estos es crucial para el éxito del proyecto, ya que su nivel de apoyo o resistencia puede afectar directamente los resultados.

**Interesados en este Proyecto:**

* **Propietarios de las Barberías:** Son los dueños de las barberías que utilizan la plataforma para gestionar sus citas, productos y servicios.
* **Clientes (Usuarios finales):** Los clientes que usarán la plataforma para agendar citas, ver productos y pagar servicios.
* **Proveedores de Tecnología:** Incluye a los desarrolladores que trabajarán en el sistema y proveedores de servicios de infraestructura (hosting, licencias de software, herramientas de análisis de datos, etc.).
* **Equipo de Desarrollo:** El equipo interno o subcontratado que se encarga del desarrollo, mantenimiento y actualización de la plataforma.
* **Reguladores de Seguridad (si aplica):** Entidades o normativas encargadas de garantizar la seguridad y privacidad de los datos de los usuarios, como regulaciones de privacidad de datos o normativas de comercio electrónico.

**2. Análisis de los Interesados**

Para gestionar correctamente a los interesados, es fundamental analizar su poder e interés en el proyecto. La **Matriz de Poder-Interés** es una herramienta útil para evaluar el nivel de involucramiento que requiere cada grupo de interesados.

**Matriz de Poder-Interés:**

| **Interesado** | **Poder** | **Interés** | **Estrategia de Gestión** |
| --- | --- | --- | --- |
| **Propietarios de Barberías** | Alto | Alto | Gestionar estrechamente. Asegurar que estén involucrados en todas las fases del proyecto. Realizar reuniones quincenales y proporcionar reportes detallados de progreso. |
| **Clientes (Usuarios finales)** | Bajo | Alto | Mantener informado. Enviar actualizaciones periódicas sobre avances y promociones. Utilizar notificaciones en la plataforma. |
| **Proveedores de Tecnología** | Alto | Medio | Gestionar de cerca. Mantener comunicación continua, reuniones semanales para garantizar el cumplimiento de plazos y calidad. |
| **Equipo de Desarrollo** | Alto | Medio | Mantener comunicado. Coordinar reuniones semanales y brindar soporte constante para asegurar la eficiencia del trabajo. |
| **Reguladores de Seguridad** | Bajo | Bajo | Mantener informado. Asegurar que las normativas se cumplan y que no haya problemas de seguridad. Reportes anuales o cuando se requiera. |

**3. Desarrollo del Plan de Comunicación**

La comunicación efectiva con los interesados es esencial para el éxito del proyecto. El Plan de Comunicación debe definir cómo se comunicará el progreso del proyecto, la información relevante y los cambios que ocurran en el camino.

**Métodos de Comunicación:**

| **Interesado** | **Método de Comunicación** | **Frecuencia** | **Responsable** |
| --- | --- | --- | --- |
| **Propietarios de Barberías** | Reuniones y reportes detallados. | Quincenales | Carlos Muñoz (PM) |
| **Clientes (Usuarios finales)** | Notificaciones a través de la plataforma y correos electrónicos. | Mensuales o según actualizaciones | Ángel Oscátegui (Comunicaciones) |
| **Proveedores de Tecnología** | Reuniones de seguimiento y correos electrónicos. | Semanales | Damián Salazar (TI) |
| **Equipo de Desarrollo** | Reuniones de equipo y actualizaciones internas. | Semanales | Ángel Oscátegui (Tareas) |
| **Reguladores de Seguridad** | Reportes de cumplimiento y revisiones de seguridad. | Anuales o cuando se requiera | Carlos Muñoz (PM) |

**4. Manejo de Expectativas**

Gestionar las expectativas de los interesados es crucial para mantener una relación sana y productiva. A medida que el proyecto avanza, es necesario monitorear las expectativas de los interesados para asegurarse de que estén alineadas con los avances del proyecto.

**Estrategias para Manejar Expectativas:**

* **Propietarios de Barberías:** Asegurar que las expectativas sobre el alcance de las funcionalidades sean realistas. Proporcionar una línea directa de comunicación para resolver cualquier problema de inmediato.
* **Clientes (Usuarios finales):** Gestionar sus expectativas relacionadas con la funcionalidad de la plataforma, garantizando una experiencia de usuario fluida y libre de errores. Actualizar sobre las nuevas funcionalidades.
* **Proveedores de Tecnología:** Asegurarse de que comprendan claramente los plazos y los requisitos técnicos, gestionando posibles cambios de alcance o retrasos.
* **Equipo de Desarrollo:** Crear un ambiente de trabajo abierto donde se resuelvan dudas de manera continua y se mantenga una carga de trabajo equilibrada.

**5. Monitoreo de la Participación**

La participación activa de los interesados a lo largo del proyecto es esencial para su éxito. Es necesario evaluar de manera continua su nivel de involucramiento y ajustar las estrategias de comunicación e involucramiento conforme avanza el proyecto.

**Métodos para Monitorear la Participación:**

* **Revisión mensual del involucramiento:** Se realizarán reuniones de evaluación del proyecto para discutir el nivel de participación de cada grupo de interesados.
* **Encuestas a clientes:** Al finalizar ciertas fases del proyecto, se enviarán encuestas a los usuarios finales para obtener retroalimentación sobre su experiencia con la plataforma.
* **Informe de avance al equipo:** El equipo de desarrollo será evaluado semanalmente para asegurar que se mantenga comprometido con los plazos y objetivos del proyecto.
* **Evaluación de proveedores:** Evaluar periódicamente la calidad del servicio y cumplimiento de plazos por parte de los proveedores de tecnología.

# Registro de Cambios del Proyecto

El objetivo de este proceso es asegurar que cualquier cambio que ocurra durante el ciclo de vida del proyecto sea gestionado de manera controlada y documentada. Esto permite minimizar el impacto sobre los objetivos del proyecto, manteniendo el control sobre el alcance, tiempo y costo del proyecto.

**1. Proceso de Control de Cambios**

La gestión de cambios es esencial para el buen desarrollo del proyecto. Establecer un proceso claro y formal de control de cambios ayuda a garantizar que las modificaciones al proyecto se gestionen de forma adecuada, evitando desviaciones imprevistas que puedan afectar negativamente el cumplimiento de los objetivos.

**Pasos del Proceso de Control de Cambios:**

1. **Solicitud de Cambio:**
   * Cualquier cambio propuesto será registrado formalmente, ya sea por el equipo de trabajo, un interesado o el cliente. La solicitud debe contener una descripción detallada del cambio propuesto, los motivos del mismo y el área del proyecto que se verá afectada.
   * Ejemplo: Un cambio en la funcionalidad de la plataforma debido a un nuevo requisito de los propietarios de las barberías.
2. **Evaluación del Impacto:**
   * Una vez recibida la solicitud, se evaluará el impacto del cambio en varios aspectos del proyecto: tiempo, costo y alcance. Esto se realiza con la participación de los líderes de equipo y, en algunos casos, de los interesados clave para valorar el efecto potencial.
   * Ejemplo: Si se solicita una nueva funcionalidad, se evaluará cuánto costará agregarla, cuánto tiempo tomará implementarla y si altera los entregables previamente definidos.
3. **Aprobación o Rechazo:**
   * Después de evaluar el impacto, la solicitud será presentada al Comité de Proyecto o al responsable del proyecto para su aprobación. El cambio será aceptado si se justifica por su beneficio o si se considera necesario para el éxito del proyecto. En caso contrario, se rechazará.
   * Ejemplo: Si el impacto en el costo es significativo, el comité podría rechazar el cambio si no hay justificación suficiente o si excede el presupuesto aprobado.
4. **Implementación y Comunicación del Cambio:**
   * Una vez aprobado el cambio, el equipo de trabajo procederá con su implementación. Posteriormente, se comunicará a todos los interesados sobre el cambio realizado, detallando los ajustes realizados en el proyecto y su impacto.
   * Ejemplo: Se actualizarán los cronogramas, los presupuestos y se informará a los proveedores o clientes sobre las nuevas características que se están incorporando.

**2. Registro de Cambios**

Mantener un **registro detallado de todos los cambios** solicitados, aprobados y implementados es crucial para llevar un control transparente de todas las modificaciones que se realizan en el proyecto. Este documento no solo facilita el seguimiento, sino que también proporciona evidencia de las decisiones tomadas en relación con los cambios.

El registro debe incluir los siguientes detalles:

* **ID de Cambio:** Un identificador único para cada cambio.
* **Descripción:** Una descripción breve y precisa del cambio solicitado.
* **Impacto en Costo:** Una estimación del impacto del cambio en los costos del proyecto.
* **Impacto en Tiempo:** La cantidad de tiempo adicional o reducido que el cambio podría implicar.
* **Fecha de Aprobación:** La fecha en que el cambio fue aprobado.
* **Aprobado por:** Nombre del responsable que aprobó el cambio (puede ser el Comité de Proyecto o el gerente del proyecto).

**Ejemplo de Registro de Cambios:**

| **ID Cambio** | **Descripción** | **Impacto en Costo** | **Impacto en Tiempo** | **Fecha de Aprobación** | **Aprobado por** |
| --- | --- | --- | --- | --- | --- |
| 001 | Ampliar funcionalidad de notificaciones | $1000 | 1 semana | 10/01/2024 | Comité de Proyecto |
| 002 | Integración con sistema de pagos externo | $3000 | 2 semanas | 15/01/2024 | Carlos Muñoz (PM) |
| 003 | Mejora en la interfaz de usuario para móviles | $1500 | 1 semana | 20/01/2024 | Comité de Proyecto |

**3. Implementación y Monitoreo de Cambios**

Una vez aprobado un cambio, su implementación debe ser monitoreada de cerca para asegurar que no afecte negativamente a otras partes del proyecto. Los responsables de la implementación deben reportar sobre el progreso de cada cambio y cualquier desviación que ocurra durante su ejecución.

**Plan de Monitoreo:**

* **Revisiones periódicas:** Se deben realizar reuniones de seguimiento donde se verifique el avance de la implementación de los cambios, los problemas que puedan surgir y los ajustes necesarios.
* **Actualización de Documentación:** Los cambios aprobados deben reflejarse inmediatamente en la documentación oficial del proyecto, como el cronograma, el presupuesto y las especificaciones del proyecto.

El **Registro de Cambios** es una herramienta esencial en la gestión de proyectos, ya que asegura que todas las modificaciones sean tratadas de manera controlada y alineada con los objetivos del proyecto. Implementar un proceso claro de control de cambios no solo permite gestionar las expectativas de los interesados, sino también optimizar los recursos del proyecto, evitando aumentos innecesarios en los costos o retrasos en los plazos establecidos.

Este proceso será revisado y actualizado periódicamente para garantizar que se mantenga la efectividad del control de cambios, adaptándose a las necesidades cambiantes del proyecto y los requisitos de los interesados.

# Documentación de Lecciones Aprendidas del Proyecto

El objetivo de la **Documentación de Lecciones Aprendidas** es capturar y compartir los aprendizajes obtenidos a lo largo del ciclo de vida del proyecto, con el fin de mejorar los procesos en proyectos futuros. Este proceso permite reflexionar sobre los éxitos y las dificultades enfrentadas durante la ejecución, de modo que las buenas prácticas se mantengan y los errores se eviten en proyectos venideros.

La gestión del conocimiento y la documentación de lecciones aprendidas son vitales para cualquier organización que busque mejorar continuamente su desempeño y gestionar proyectos con una mayor eficiencia.

**1. Captura de Lecciones Aprendidas**

A lo largo del ciclo de vida del proyecto, es fundamental **capturar las lecciones aprendidas** en cada una de sus fases: inicio, planificación, ejecución y cierre. Este proceso ayuda a identificar lo que funcionó bien, lo que no funcionó como se esperaba, y lo que podría haberse hecho de manera diferente para obtener mejores resultados.

**Métodos para Capturar Lecciones Aprendidas:**

1. **Reuniones Retrospectivas:**
   * Las reuniones retrospectivas son una práctica común en metodologías ágiles y otras disciplinas de gestión de proyectos. Estas reuniones se realizan al final de cada fase del proyecto y se centran en reflexionar sobre los resultados obtenidos. Los equipos discuten qué aspectos fueron exitosos, qué problemas surgieron y qué cambios podrían haberse hecho para mejorar.
   * Ejemplo de una pregunta clave: "¿Qué funcionó bien y qué no durante esta fase?"
2. **Encuestas a Miembros del Equipo y Otros Interesados:**
   * Realizar encuestas o entrevistas a los miembros del equipo, así como a los interesados del proyecto (clientes, proveedores, etc.), permite obtener una perspectiva más amplia sobre el desempeño del proyecto. A través de estas encuestas, se pueden recoger opiniones y sugerencias que de otro modo podrían no ser expresadas en un entorno de equipo.
   * Las encuestas pueden incluir preguntas como: "¿Qué desafíos enfrentaron durante la ejecución del proyecto?" o "¿Cómo podríamos haber gestionado mejor el tiempo o los recursos?"
3. **Análisis de Documentación del Proyecto:**
   * Revisar los informes del proyecto, los registros de cambios y los informes de seguimiento puede proporcionar información valiosa sobre los problemas que surgieron, las soluciones aplicadas y los resultados obtenidos. Los documentos de control como los cronogramas, presupuestos y reportes de calidad también ofrecen un panorama claro sobre las áreas de mejora.
4. **Revisión de Riesgos y Problemas Resueltos:**
   * Identificar los riesgos que se materializaron y los problemas que surgieron durante el proyecto es otra fuente importante de lecciones aprendidas. Analizar cómo se gestionaron estos eventos y qué tan efectivos fueron los planes de contingencia es clave para mejorar la preparación y las respuestas ante situaciones similares en el futuro.

**2. Documentación de las Lecciones Aprendidas**

Cada lección aprendida debe ser **documentada cuidadosamente** para garantizar que el conocimiento adquirido se pueda compartir y utilizar en el futuro. Esta documentación debe ser precisa y contener información detallada sobre los éxitos, los fracasos, las acciones correctivas y las recomendaciones.

**Elementos Clave de la Documentación:**

1. **Descripción de la Lección Aprendida:**
   * Cada lección debe comenzar con una descripción clara y concisa de lo sucedido. Esto incluye el contexto, los factores involucrados y cómo el equipo o los interesados se enfrentaron al desafío o aprovecharon la oportunidad.
   * Ejemplo: “Los tiempos de entrega del software fueron más largos de lo esperado debido a la escasez de recursos humanos y a la falta de planificación de capacidad.”
2. **Impacto en el Proyecto:**
   * Se debe describir el impacto que tuvo la lección aprendida en el proyecto, tanto positivo como negativo. Este apartado también debe incluir una evaluación del impacto en el cronograma, el presupuesto, el alcance y los objetivos del proyecto.
   * Ejemplo: "La demora en el desarrollo causó un retraso en la entrega del producto final, lo que afectó la satisfacción del cliente y provocó una sobrecarga de trabajo en el equipo durante las fases finales."
3. **Acciones Correctivas o Preventivas:**
   * Es importante documentar las acciones tomadas para corregir la situación o prevenir que el mismo problema ocurra en el futuro. Estas acciones deben estar claramente definidas y ser accesibles para futuras referencias.
   * Ejemplo: “Se decidió contratar personal adicional al inicio de la fase de desarrollo para garantizar que no hubiera escasez de recursos. También se implementó un sistema de planificación de capacidad para mejorar la previsibilidad del trabajo.”
4. **Recomendaciones para el Futuro:**
   * Las recomendaciones deben estar orientadas a cómo mejorar los procesos para proyectos futuros. Este apartado proporciona sugerencias útiles que pueden ayudar a otros equipos a evitar los mismos errores o a mejorar en áreas específicas.
   * Ejemplo: “Es recomendable que se asignen recursos con antelación para evitar retrasos imprevistos. Además, se debe revisar el presupuesto con más frecuencia para ajustarse a los cambios durante la ejecución.”

**Ejemplo de Lección Aprendida Documentada:**

| Lección Aprendida | Descripción | Impacto en el Proyecto | Acciones Correctivas/Preventivas | Recomendaciones para el Futuro |
| --- | --- | --- | --- | --- |
| **Retraso en la entrega del software** | Durante la fase de desarrollo, los tiempos de entrega fueron más largos de lo esperado debido a la escasez de recursos. | El retraso afectó la entrega a tiempo del producto, causando insatisfacción en los clientes y sobrecarga de trabajo en el equipo. | Se contrató personal adicional al inicio de la fase de desarrollo y se implementó un plan de capacidad para asegurar la disponibilidad de recursos. | Asegurar la contratación de personal necesario al inicio del proyecto y establecer revisiones periódicas del presupuesto y la asignación de recursos. |

**3. Utilización de las Lecciones Aprendidas**

Las lecciones aprendidas no deben quedar archivadas en un documento; deben ser **compartidas** con todo el equipo y con los interesados clave para asegurar que todos se beneficien del conocimiento adquirido. Es fundamental crear un entorno de aprendizaje continuo donde las experiencias pasadas se utilicen para mejorar la ejecución de proyectos futuros.

**Métodos para Compartir las Lecciones Aprendidas:**

1. **Reuniones de Compartición de Conocimientos:**
   * Organizar reuniones periódicas donde se presenten y discutan las lecciones aprendidas de proyectos anteriores. Esto facilita la difusión del conocimiento dentro de la organización.
2. **Plataformas de Gestión del Conocimiento:**
   * Utilizar plataformas digitales (como wikis, bases de datos de proyectos, o herramientas de gestión del conocimiento) para almacenar y hacer accesibles las lecciones aprendidas a todo el equipo y a futuros proyectos.
3. **Capacitación y Formación:**
   * Incorporar las lecciones aprendidas en programas de capacitación y formación para que los miembros del equipo adquieran las habilidades necesarias para abordar problemas similares de manera más eficaz en el futuro.

# Registro de Recursos Adquiridos del Proyecto

El objetivo del **Registro de Recursos Adquiridos** es gestionar la adquisición de los recursos necesarios para el desarrollo y ejecución exitosa del proyecto. Estos recursos incluyen tanto los **recursos humanos**, como los **tecnológicos** y **materiales**, todos esenciales para cumplir con los objetivos establecidos en el plan de proyecto. La correcta adquisición y seguimiento de estos recursos garantiza que el proyecto se desarrolle conforme a lo planificado y dentro de los plazos y presupuestos estipulados.

**1. Tipo de Recursos**

Para llevar a cabo este proyecto, se deben identificar y adquirir una variedad de recursos. Estos recursos pueden clasificarse en tres categorías principales:

* **Recursos Humanos:** Incluye personal con las habilidades y competencias necesarias para llevar a cabo las actividades del proyecto, tales como desarrolladores, diseñadores, especialistas en infraestructura y demás roles clave.
  + Ejemplo: Ingenieros de software, técnicos de soporte, consultores de infraestructura.
* **Recursos Tecnológicos:** Engloba los equipos y herramientas tecnológicas necesarias para el desarrollo y la ejecución del proyecto, tales como servidores, software, licencias, y herramientas especializadas.
  + Ejemplo: Servidores web, licencias de software, herramientas de desarrollo (IDEs), plataformas de gestión de proyectos.
* **Recursos Materiales:** Son los elementos tangibles que se requieren para las operaciones del proyecto, aunque este tipo de recursos puede no ser tan prominente en proyectos tecnológicos, es importante tenerlos en cuenta en ciertas fases.
  + Ejemplo: Equipos de oficina, computadoras, dispositivos de red, etc.

**2. Monitoreo de los Recursos Adquiridos**

El seguimiento adecuado de los recursos adquiridos es esencial para asegurar que los mismos sean utilizados conforme a lo planificado, sin desviaciones innecesarias. Este monitoreo permite garantizar que los recursos estén disponibles en los momentos clave del proyecto, evitando retrasos o sobrecostos debido a la falta de recursos.

**Registro de Recursos Adquiridos:**

Es fundamental mantener un **registro detallado** de todos los recursos adquiridos, para lo cual se utiliza una tabla que permite documentar los elementos esenciales de cada recurso: tipo, cantidad, fecha de adquisición, responsable y su uso planeado dentro del proyecto.

**Tabla de Registro de Recursos Adquiridos:**

| **Recurso** | **Cantidad** | **Fecha de Adquisición** | **Responsable** | **Uso Planeado** |
| --- | --- | --- | --- | --- |
| Servidor Web | 2 | 01/02/2024 | Carlos Muñoz | Hosting Plataforma |
| Licencia Software | 10 | 01/03/2024 | Damián Salazar | Desarrollo de la Plataforma |
| Equipo de Desarrollo | 5 | 15/02/2024 | Ángel Oscátegui | Desarrollo Backend y Frontend |
| Herramienta de Diseño | 1 | 01/03/2024 | Carlos Muñoz | Diseño de Interfaz de Usuario (UI) |

**3. Proceso de Adquisición y Monitoreo**

El proceso de adquisición y monitoreo de recursos se lleva a cabo en los siguientes pasos:

1. **Identificación de la Necesidad de Recursos:**
   * Antes de adquirir cualquier recurso, se debe identificar qué recursos son esenciales para cada fase del proyecto, según el plan de trabajo establecido. Esta identificación incluye la cantidad de recursos requeridos y el momento en que deben estar disponibles.
2. **Adquisición de Recursos:**
   * Se procede con la adquisición de los recursos a través de proveedores aprobados o recursos internos, como la contratación de personal o la compra de equipos. Los proveedores se seleccionan en función de la calidad, costo, y capacidad de entrega a tiempo.
3. **Registro de los Recursos Adquiridos:**
   * Una vez que los recursos son adquiridos, se documentan en el registro de recursos, donde se incluyen todos los detalles relevantes: tipo de recurso, fecha de adquisición, cantidad, responsable, y el uso previsto.
4. **Monitoreo del Uso de los Recursos:**
   * A lo largo del ciclo de vida del proyecto, se realiza un seguimiento continuo de los recursos adquiridos para asegurarse de que estén siendo utilizados conforme a lo planeado. Si es necesario, se ajustan las asignaciones o se adquieren más recursos para cumplir con las demandas del proyecto.
5. **Evaluación y Ajuste:**
   * Regularmente se evalúa la disponibilidad y uso de los recursos. En caso de desviaciones, se realiza un ajuste en la planificación para asegurar que el proyecto se mantenga dentro de los plazos y presupuesto previstos.

**4. Importancia del Registro de Recursos Adquiridos**

El **registro de recursos adquiridos** es crucial para varios aspectos del proyecto:

* **Gestión del Presupuesto:** Permite tener un control detallado sobre los gastos asociados a los recursos adquiridos, ayudando a identificar desviaciones en los costos del proyecto.
* **Cumplimiento de Plazos:** Garantiza que los recursos estén disponibles cuando sean necesarios, evitando retrasos debido a la falta de equipos, personal o tecnología.
* **Optimización de Recursos:** Facilita la gestión eficiente de los recursos, evitando tanto el exceso como la escasez de los mismos, lo que contribuye a la optimización de los costos y tiempos.
* **Trazabilidad:** En caso de que se necesiten realizar auditorías o evaluaciones post-proyecto, el registro de recursos proporciona una fuente confiable para rastrear la adquisición y el uso de los recursos durante el ciclo de vida del proyecto.

# Desarrollar el Equipo - Plan de Desarrollo del Equipo

El objetivo del **Plan de Desarrollo del Equipo** es asegurar que el equipo de trabajo esté motivado, capacitado y cohesionado, para que pueda cumplir con los objetivos del proyecto de manera eficiente. Un equipo bien formado y comprometido es crucial para el éxito de cualquier proyecto, ya que permite optimizar los recursos, mejorar la colaboración y garantizar que se cumplan los plazos establecidos.

**Plan de Desarrollo del Equipo:**

1. **Entrenamiento y Capacitación:**
   * Es fundamental planificar sesiones de **capacitación** para mejorar las habilidades técnicas y de gestión del equipo. Esto debe incluir:
     + Capacitación técnica en áreas clave del proyecto (como desarrollo de software, uso de herramientas tecnológicas, etc.).
     + Desarrollo de habilidades blandas (comunicación, liderazgo, gestión de conflictos).
     + Talleres de integración para fomentar el trabajo en equipo y la cohesión.
   * **Plan de Acción:** Se organizarán talleres bimestrales de capacitación y sesiones de intercambio de conocimientos para abordar cualquier brecha de habilidades que surja.
2. **Manejo de Conflictos:**
   * Los conflictos pueden surgir en cualquier equipo, y su gestión efectiva es esencial para mantener un ambiente de trabajo positivo y productivo.
   * Establecer un proceso para la resolución de conflictos que contemple:
     + **Proceso de mediación:** Un sistema estructurado para resolver disputas internas mediante reuniones individuales y de grupo.
     + **Entrenamiento en resolución de conflictos:** Capacitación continua en técnicas de negociación y manejo de conflictos.
   * **Plan de Acción:** Designar un facilitador de conflictos dentro del equipo y fomentar la comunicación abierta.
3. **Evaluación de Desempeño:**
   * Para asegurar que el equipo se mantenga enfocado y motivado, se deben realizar evaluaciones periódicas del desempeño, tanto de manera formal como informal.
     + **Evaluaciones de habilidades técnicas:** Revisión de las capacidades técnicas de cada miembro en relación con los requerimientos del proyecto.
     + **Evaluaciones de habilidades blandas:** Evaluar la colaboración, liderazgo y comunicación dentro del equipo.
   * **Plan de Acción:** Evaluaciones trimestrales y retroalimentación constructiva, con un enfoque en el desarrollo continuo de cada miembro.

# Dirigir al Equipo - Evaluación de Desempeño del Equipo

La **Evaluación de Desempeño del Equipo** busca medir el rendimiento global del equipo, identificar áreas de mejora y aplicar las medidas correctivas necesarias para asegurar que el proyecto siga su curso exitosamente.

**Evaluación de Desempeño:**

1. **Criterios de Evaluación:**
   * El desempeño del equipo será evaluado con base en los siguientes criterios:
     + **Habilidades Técnicas:** Evaluación del conocimiento y la ejecución técnica de los miembros.
     + **Colaboración y Trabajo en Equipo:** Capacidad para trabajar juntos de manera efectiva.
     + **Cumplimiento de Plazos:** La habilidad de cumplir con los entregables y los plazos establecidos.
     + **Comunicación:** Eficiencia y claridad en las interacciones internas y con los interesados.
   * **Plan de Acción:** Evaluaciones semestrales de cada miembro y del equipo en su conjunto, con discusiones abiertas sobre áreas de mejora.
2. **Acciones Correctivas:**
   * Las **acciones correctivas** que se tomarán en caso de bajo desempeño incluyen:
     + **Retroalimentación constructiva**: Reuniones individuales para abordar áreas de mejora.
     + **Formación adicional**: Programas de capacitación si se detectan deficiencias en habilidades técnicas o interpersonales.
     + **Reasignación de tareas**: Si un miembro no está cumpliendo con sus responsabilidades, se considerará la reasignación de tareas para optimizar la carga de trabajo del equipo.

# Gestionar las Comunicaciones - Informe de Estado del Proyecto

El **Informe de Estado del Proyecto** tiene como objetivo garantizar que todos los interesados del proyecto reciban información precisa y actualizada sobre el progreso, los problemas y las acciones correctivas necesarias.

**Informe de Estado del Proyecto:**

1. **Contenido del Informe:** El informe debe incluir los siguientes elementos clave:
   * **Estado Actual de las Tareas:** Un resumen del avance de las tareas programadas, con un enfoque en las que están atrasadas o avanzadas según lo planeado.
   * **Problemas y Riesgos Identificados:** Descripción de cualquier obstáculo o desafío que haya surgido, junto con los riesgos identificados que puedan afectar el proyecto.
   * **Recomendaciones de Acciones Correctivas:** Sugerencias y pasos a seguir para abordar los problemas y mitigar los riesgos. Este punto incluye tanto la solución técnica como las acciones de gestión necesarias para volver a encaminar el proyecto.
   * **Frecuencia:** El informe será emitido quincenalmente o mensualmente, dependiendo de la fase del proyecto.

# Controlar el Proyecto - Informe de Desempeño

El **Informe de Desempeño** tiene como objetivo proporcionar una visión clara y precisa de cómo se está desempeñando el proyecto en términos de tiempo, costo, calidad y otros indicadores clave, permitiendo así la toma de decisiones informadas para mantener el proyecto en el camino correcto.

**Informe de Desempeño:**

1. **Indicadores Clave de Desempeño (KPIs):** Los **KPIs** que se monitorearán a lo largo del proyecto incluyen:
   * **Tiempo:** Comparación entre el cronograma planificado y el real, identificando posibles retrasos.
   * **Costo:** Control de los gastos reales frente al presupuesto aprobado.
   * **Calidad:** Evaluación de los entregables en relación con los estándares de calidad establecidos.
   * **Satisfacción del Cliente:** Medición del nivel de satisfacción de los clientes o usuarios finales mediante encuestas o entrevistas.
   * **Frecuencia:** Los informes de desempeño se realizarán mensualmente y se compartirán con los interesados clave del proyecto.

# Gestionar el Riesgo - Plan de Gestión de Riesgos

El **Plan de Gestión de Riesgos** tiene como objetivo identificar los riesgos del proyecto, evaluar su impacto y desarrollar estrategias para mitigar esos riesgos, asegurando que el proyecto continúe avanzando según lo planeado.

**Plan de Gestión de Riesgos:**

1. **Identificación de Riesgos:** Se deben identificar y clasificar los posibles riesgos que podrían afectar el proyecto. Ejemplos de riesgos incluyen:
   * **Riesgos Técnicos:** Problemas con la infraestructura, fallos en el software o hardware, cambios en la tecnología.
   * **Riesgos de Cronograma:** Retrasos en las entregas, problemas de programación.
   * **Riesgos de Recursos:** Escasez de personal, falta de habilidades técnicas, recursos financieros limitados.
2. **Mitigación de Riesgos:** Una vez identificados los riesgos, se deben desarrollar estrategias para mitigar su impacto:
   * **Planes de contingencia:** Desarrollar planes alternativos en caso de que ocurra un riesgo.
   * **Asignación de responsabilidades:** Definir roles claros para gestionar cada tipo de riesgo.
   * **Revisión continua:** Monitorear y evaluar continuamente los riesgos durante todo el ciclo de vida del proyecto.

# Cerrar el Proyecto - Informe de Cierre

**Resumen de Logros**

A lo largo del proyecto, se lograron cumplir los objetivos planteados en el plan de gestión y desarrollo. A continuación, se detallan los principales logros:

| **Objetivo** | **Estado** | **Comentarios** |
| --- | --- | --- |
| Diseño de la arquitectura del sistema | Completado | La arquitectura desarrollada fue validada por expertos. |
| Modelado y desarrollo de la base de datos | Completado | Incluye gestión de usuarios, reservas y barberías. |
| Implementación de Google Calendar | Completado | Sincronización funcional con horarios de barberías. |
| Creación de la página de administración | Completado | Los propietarios pueden gestionar reservas y usuarios. |
| Plataforma responsiva para dispositivos móviles | Completado | Pruebas de compatibilidad exitosas. |
| Capacitación a usuarios finales | Completado | Realizadas sesiones de entrenamiento en línea. |

**Impacto General:** El proyecto logró ofrecer una plataforma funcional y moderna que facilita la reserva de servicios, la gestión de barberos y la integración con herramientas tecnológicas como Google Calendar.

**2. Lecciones Aprendidas**

| **Aspecto** | **Lección** | **Recomendación** |
| --- | --- | --- |
| Gestión del Tiempo | La planificación inicial no consideró suficiente tiempo para pruebas de compatibilidad. | Incluir mayores márgenes de tiempo para pruebas y correcciones. |
| Comunicación con los Interesados | La comunicación con los propietarios tuvo retrasos al inicio del proyecto. | Establecer puntos de contacto frecuentes desde las primeras etapas. |
| Gestión de Riesgos | No se identificó oportunamente la necesidad de recursos adicionales para la infraestructura. | Realizar evaluaciones de riesgos más exhaustivas en la etapa de planificación. |
| Integración de Tecnología | La integración de Google Calendar presentó desafíos técnicos no previstos. | Investigar más a fondo las dependencias técnicas antes de comenzar el desarrollo. |

**3. Satisfacción de los Interesados**

| **Interesado** | **Nivel de Satisfacción (1-5)** | **Comentarios** |
| --- | --- | --- |
| Propietarios de barberías | 5 | Expresaron satisfacción con la funcionalidad y la interfaz de usuario. |
| Clientes finales | 4 | Usuarios valoraron la facilidad de uso, aunque sugirieron añadir más métodos de pago. |
| Equipo de desarrollo | 4 | Buen desempeño general, aunque se identificaron oportunidades para mejorar la gestión. |

**4. Frecuencia y Procedimientos**

El **Informe de Cierre** se emitió tras:

1. **Entrega de los entregables:** Todos los componentes del sistema fueron entregados, probados y aceptados por los interesados clave.
2. **Evaluación de satisfacción:** Se recopilaron datos mediante encuestas y reuniones de retroalimentación.

**5. Recomendaciones Finales**

* Documentar de forma completa los aspectos técnicos y funcionales de la plataforma para facilitar su mantenimiento.
* Establecer una línea de soporte para garantizar la continuidad operativa de la plataforma.
* Considerar futuras iteraciones del proyecto para incluir funcionalidades adicionales solicitadas por los usuarios.

# Conclusiones individuales solo en inglés.

Carlos Muñoz

My participation in the project to develop a website for the barbershop chain in the Lo Espejo commune has been a rewarding and challenging experience. As the **project manager**, I focused my efforts on ensuring that each phase was executed in alignment with the principles of the PMBOK framework, meeting the objectives on time and within scope.

Throughout this process, I oversaw the planning, organization, and coordination of the team, ensuring effective communication among members and proper task allocation based on their roles and responsibilities. Additionally, I worked closely on defining the system architecture and database modeling, ensuring the proposed solution was scalable, functional, and aligned with the client’s needs.

This project not only allowed me to solidify my technical knowledge in areas such as software architecture and database design but also to strengthen key skills like leadership, problem-solving, and decision-making. I am satisfied with the results achieved and consider this project an excellent opportunity to apply the knowledge acquired during my training as a computer engineer, contributing to a final product that meets the expected quality standards.”

**Damián Salazar**

My role in the project allowed me to strengthen my skills in project management, teamwork, and problem-solving. By coordinating tasks and aligning them with the project's objectives, I gained practical experience in leadership and decision-making.

This project also enhanced my understanding of system architecture and database modeling, providing valuable insights into creating scalable and user-focused solutions. Overall, it was an enriching experience that reinforced my professional growth and readiness for real-world challenges.”

**Angel Oscategui**

“Participating in this project was a valuable opportunity to develop my skills in project management, system design, and collaboration. As the project manager, I focused on coordinating tasks, ensuring effective communication within the team, and aligning our work with the overall objectives. This experience deepened my understanding of leadership, time management, and the importance of maintaining a clear vision throughout the development process.

Working on the system architecture and database modeling allowed me to explore scalable solutions that address real-world needs, reinforcing my technical knowledge and problem-solving abilities. Additionally, the project highlighted the importance of adaptability, as we encountered challenges that required creative thinking and teamwork to resolve.

Beyond the technical aspects, this project emphasized the value of designing user-focused solutions that are both functional and intuitive. It provided a practical setting to apply theoretical concepts and demonstrated how technology can make a positive impact when aligned with user needs. This experience has strengthened my confidence and prepared me for future professional challenges.**”**

# Descripción del aporte del proyecto en el desarrollo de nuestros intereses profesionales

El desarrollo del sitio web para la cadena de barberías en la comuna de Lo Espejo representó una oportunidad invaluable para todos los integrantes del equipo, permitiéndonos alinear nuestras habilidades y roles con nuestros intereses profesionales. Este proyecto no solo fue un ejercicio académico, sino también una simulación práctica de cómo se trabaja en el ámbito profesional, enfrentando desafíos técnicos y de gestión en cada etapa del proceso.

Desde una perspectiva grupal, el proyecto nos permitió desarrollar y reforzar nuestras áreas de interés específicas:

* **Gestión de Proyectos**: Como encargado de esta área, me enfoqué en liderar y coordinar al equipo, utilizando herramientas y principios del marco PMBOK. Esto incluyó la planificación detallada de tareas, el control del cronograma y la gestión de riesgos, contribuyendo a mi interés por desempeñar roles de liderazgo en proyectos tecnológicos.
* **Arquitectura y Diseño de Software**: Ángel Oscátegui jugó un papel clave en el diseño de la estructura del sistema, asegurándose de que fuera modular, escalable y adecuado para las necesidades futuras del cliente. Esta experiencia reforzó su interés por el diseño y la planificación de soluciones tecnológicas robustas.
* **Modelado de Bases de Datos**: Damián Salazar se especializó en la construcción y optimización del sistema de bases de datos, garantizando la integridad y la eficiencia de las operaciones del sistema. Esto le permitió profundizar en su interés por los sistemas de gestión de información y el análisis de datos.

El proyecto nos ofreció la oportunidad de integrar nuestras habilidades técnicas con un enfoque práctico y colaborativo, fortaleciendo nuestra capacidad para trabajar como un equipo cohesionado. Además, nos permitió experimentar de primera mano la importancia de atender las necesidades del cliente y de los usuarios finales, lo que refuerza nuestro interés profesional en diseñar soluciones tecnológicas que generen un impacto positivo en el mundo real.

**Competencias del perfil de egreso evidenciadas en el desarrollo de nuestro proyecto**

A lo largo del desarrollo del proyecto, nuestro equipo demostró competencias fundamentales del perfil de egreso que reflejan nuestra preparación integral como futuros ingenieros en computación. Estas competencias incluyen:

1. **Gestión y Liderazgo de Proyectos**: A través del uso de metodologías organizativas y de gestión, como el marco PMBOK, logramos planificar y ejecutar cada fase del proyecto, asegurando el cumplimiento de los plazos y objetivos. Este proceso evidenció nuestra capacidad para liderar proyectos multidisciplinarios.
2. **Diseño y Arquitectura de Sistemas**: Diseñamos un sistema estructurado y escalable que permite integrar funcionalidades como reservas en línea, gestión de usuarios, integración con Google Calendar y un sistema de administración para las barberías. Esta competencia asegura que podemos crear soluciones alineadas con estándares de la industria.
3. **Optimización de Bases de Datos**: Creamos un modelo de datos eficiente que permite manejar grandes volúmenes de información con integridad y seguridad, abordando áreas clave como reservas, usuarios y productos.
4. **Colaboración y Trabajo en Equipo**: Como grupo, trabajamos de manera colaborativa, aprovechando nuestras fortalezas individuales para lograr los objetivos colectivos. Esta sinergia se tradujo en un producto final que reflejó el aporte de cada miembro del equipo.
5. **Resolución de Problemas**: Enfrentamos y resolvimos desafíos técnicos, como la integración de funcionalidades avanzadas y la adaptación de la página web a diferentes dispositivos. Esto evidenció nuestra capacidad para analizar problemas y desarrollar soluciones creativas y efectivas.
6. **Enfoque en el Usuario**: Diseñamos un sistema centrado en la experiencia del usuario, garantizando una navegación intuitiva, accesibilidad y funciones útiles que respondieran a las necesidades tanto de los clientes como de los barberos.
7. **Adaptabilidad y Aprendizaje Continuo**: Durante el proyecto, tuvimos que aprender y aplicar nuevas herramientas y tecnologías, lo que demostró nuestra capacidad de adaptarnos rápidamente a los cambios y de mantenernos actualizados en un campo en constante evolución.

En conjunto, estas competencias reflejan nuestra capacidad para desempeñarnos profesionalmente en proyectos tecnológicos complejos y para generar soluciones que integren tanto aspectos técnicos como humanos. Este proyecto fue una prueba de nuestro compromiso, esfuerzo y preparación para los desafíos del futuro.

# Evidencias del proyecto

# Arquitectura de Software

La arquitectura del sistema *BarberConnect* está representada siguiendo el enfoque del framework 4+1 y las recomendaciones del proceso unificado. Las vistas incluidas en esta versión del documento son:

* **Vista de Casos de Uso y Escenarios de Calidad**: Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.

**Caso de uso “registro de usuario”:**

Para realizar el registro de usuario en *BarberConnect*, interactúan tres actores:

* **Usuario**: Persona que desea agendar una cita en una barbería.
* **Sistema**: Plataforma que valida los datos y registra al usuario.
* **Google (opcional)**: Permite el registro rápido a través de la cuenta de Google.

El usuario puede registrarse ingresando directamente sus datos en el sistema (nombre, correo electrónico, contraseña y número de teléfono) o utilizar su cuenta de Google. El sistema valida la información ingresada y crea el perfil del usuario en la base de datos. En el caso de utilizar Google, el sistema solicita los permisos necesarios para extraer la información de la cuenta y proceder con el registro

**Caso de uso “Agendar una Cita”:**

Para agendar una cita en una barbería participan dos actores:

* **Usuario**: Persona que busca un servicio en una barbería.
* **Sistema**: Plataforma que gestiona la agenda de la barbería.

El usuario selecciona una barbería de la lista, elige un barbero disponible, y el sistema le muestra las fechas y horarios disponibles. Tras elegir una hora, el sistema valida la disponibilidad en tiempo real y confirma la reserva. El sistema envía una notificación al usuario confirmando la cita y otra a la barbería para registrar la solicitud en su agenda.

**Escenario de calidad “Usabilidad”:** En este escenario se evalúa la facilidad de uso de la plataforma *BarberConnect*. Se implementa una interfaz sencilla e intuitiva, donde los usuarios pueden realizar sus reservas con pocos clics. Las pruebas realizadas con usuarios nuevos mostraron que podían interactuar con el sistema y agendar una cita sin dificultades ni capacitación previa.

**Escenario de calidad “portabilidad”:** El escenario de calidad basado en la portabilidad verifica la compatibilidad de *BarberConnect* en distintas plataformas y dispositivos. Las pruebas mostraron que el sistema funcionaba correctamente en navegadores móviles y de escritorio, sin afectar la funcionalidad o experiencia de usuario. Esto asegura que los usuarios puedan agendar citas desde cualquier dispositivo con acceso a internet.

* **Vista de Metas y Restricciones**: Describe las restricciones tecnológicas y normativas que influyen en las decisiones arquitectónicas del sistema:

**Tecnológicas**: El sistema debe ser compatible con dispositivos móviles y de escritorio, por lo que se utilizó un diseño responsive en el frontend (desarrollado en React). La base de datos debe ser escalable y segura, optando por MySQL con optimización de consultas.

**Restricciones Normativas**: Se cumplen con las normativas de protección de datos personales (GDPR/LPDP) al almacenar la información de los usuarios y barberos. Todo el tráfico se asegura mediante HTTPS.

* **Vista Lógica**: Describe la arquitectura lógica del sistema, presentando los módulos principales y sus responsabilidades. Esta vista usa el tipo de vista "Módulos" para representar la estructura lógica y "Componentes y Conectores" para describir el comportamiento del sistema.

**Módulo de Usuarios**: Gestiona el registro, autenticación y perfil de los usuarios.

**Módulo de Barberías**: Maneja la información de cada barbería, incluyendo su disponibilidad y servicios ofrecidos.

**Módulo de Reservas**: Coordina las citas entre usuarios y barberías, validando la disponibilidad y confirmando las reservas.

**Módulo de Notificaciones**: Gestiona el envío de notificaciones automáticas por correo electrónico o SMS sobre las reservas realizadas y recordatorios.

* **Vista de Procesos**: Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.

**Proceso de Agendamiento de Citas**: Cuando un usuario selecciona una barbería, elige un barbero disponible y selecciona un horario, el sistema verifica en tiempo real la disponibilidad de ese barbero en el horario seleccionado. Si el slot está disponible, el sistema confirma la reserva y envía notificaciones automáticas tanto al usuario como a la barbería. El proceso se sincroniza con la base de datos para evitar overbooking y asegurar que no haya conflictos de horarios entre diferentes barberos.

**Proceso de Notificaciones**: Se ejecuta en segundo plano y asegura que tanto los usuarios como las barberías reciban notificaciones sobre la reserva o recordatorios previos a la cita. Este proceso está sincronizado con el sistema de reservas para garantizar la exactitud de la información, y también incluye detalles específicos del barbero seleccionado para cada cita.

* **Vista de Implementación**: Describe los componentes de deployment construidos y sus dependencias.

**Backend**: Implementado en **Python** usando el framework **Django**, alojado en un **servidor de hosting** que nos proporciona el espacio para alojar la página web. Este componente se comunica con la base de datos **MySQL** para gestionar la información de usuarios, barberías y reservas.

**Frontend**: Desarrollado en **JavaScript** usando **React**, implementado como una aplicación web responsive. Esto asegura que la plataforma funcione correctamente en cualquier dispositivo, adaptándose a pantallas móviles, tabletas y computadoras.

**Base de Datos**: **MySQL**, también alojada en el **servidor de hosting**, gestiona todas las entidades del sistema, como usuarios, barberías, horarios y reservas. La base de datos está integrada directamente con el backend para asegurar que toda la información esté sincronizada y disponible en tiempo real.

# Objetivos y Restricciones de la Arquitectura

A continuación, se revisan las metas y restricciones de la arquitectura de software para *BarberConnect*, una página web de agendamiento diseñada para facilitar la conexión entre los clientes y las barberías, permitiendo reservar citas de manera rápida y eficiente.

# Objetivos de la arquitectura

*BarberConnect* tiene como objetivo principal ofrecer una plataforma ágil y confiable que permita a los usuarios gestionar sus citas en barberías, asegurando una experiencia fluida tanto para clientes como para barberos. El sistema busca optimizar la comunicación, aumentar la eficiencia en la gestión de citas y proporcionar una interfaz accesible y fácil de usar.

* **Desempeño**:

El desempeño es crucial para asegurar la satisfacción de los usuarios. En el caso de *BarberConnect*, el sistema debe responder de manera rápida y eficiente a las solicitudes de reserva, especialmente en un entorno con múltiples usuarios simultáneos.

Para lograr esto, utilizamos una base de datos relacional (MySQL) optimizada con índices que garantizan un acceso rápido a la información. La aplicación está diseñada para minimizar los tiempos de carga, especialmente en la búsqueda de barberías y en la validación de la disponibilidad de horarios.

* **Tolerancia a fallos**:

Es fundamental que *BarberConnect* continúe operando incluso en caso de fallos en algunos de sus componentes. El sistema se ha diseñado con redundancia en los puntos críticos, como la base de datos y el servidor de backend, asegurando que, ante un fallo parcial, el sistema pueda seguir funcionando sin interrupciones.

Para garantizar esto, se utiliza un **entorno de hosting** que incluye mecanismos de respaldo adecuados proporcionados por el proveedor de hosting. Esto asegura que cualquier fallo en el servidor no afecte la disponibilidad general de la plataforma, manteniendo la estabilidad y continuidad del servicio.

* **Seguridad**:

La seguridad es un aspecto clave para proteger los datos personales de los usuarios y barberos que interactúan con la plataforma. *BarberConnect* implementa HTTPS para todas las comunicaciones, asegurando que los datos viajen de manera encriptada entre el cliente y el servidor.

Además, utilizamos autenticación basada en **JWT (JSON Web Tokens)** para gestionar el acceso seguro a las funcionalidades del sistema. Se han implementado pruebas de seguridad, como **fuzzing**, para identificar posibles vulnerabilidades que podrían comprometer los datos de los usuarios

* **Modificabilidad/Reusó**:

La arquitectura de *BarberConnect* está diseñada para ser altamente modificable, lo que permite la fácil incorporación de nuevas funcionalidades o ajustes basados en cambios en los requisitos. Este enfoque modular asegura que los componentes individuales del sistema puedan ser reemplazados o actualizados sin necesidad de reestructurar toda la aplicación.

Además, la reutilización de componentes es clave. El código del backend y el frontend se ha desarrollado utilizando patrones modulares, lo que facilita el mantenimiento y las futuras expansiones de la plataforma, como la integración de pagos en línea o la ampliación a nivel regional.

* **Operatividad**:

El sistema debe cumplir con su propósito básico: permitir la gestión de citas de forma eficiente. *BarberConnect* está diseñado para ofrecer una experiencia de usuario fluida, con una interfaz sencilla y accesible para cualquier usuario, independientemente de su nivel de habilidad técnica.

La operatividad se asegura mediante la implementación de pruebas continuas, tanto en el frontend como en el backend, garantizando que el sistema produzca los resultados esperados bajo distintas condiciones de uso. Además, se prioriza la facilidad de uso, con una interfaz clara y bien organizada que minimiza el esfuerzo requerido por el usuario para completar tareas.

# Restricciones de la Arquitectura

Existen restricciones que han sido levantadas con los stakeholders del proyecto *BarberConnect*, las cuales se presentan a continuación:

* **Tiempo de construcción**: El plazo disponible para la implementación de la arquitectura es limitado, contando con un máximo de 6 semanas para su desarrollo completo según la planificación establecida para el proyecto. Este tiempo restringe el alcance de las funcionalidades a las más esenciales en esta primera versión.
* **Infraestructura**: El sistema se desplegará utilizando un servicio de **hosting tradicional**, lo que proporciona el espacio necesario para alojar los archivos y la base de datos del sistema sin necesidad de una infraestructura en la nube más compleja. Este hosting ofrece soporte para los lenguajes de programación **Python** (backend) y **JavaScript** (frontend)
* **Otros componentes de software**: No se considerará la adquisición de software propietario o con licencias de pago. Todo el software utilizado será de código abierto, incluyendo frameworks como **Django** (Python) para el backend y **React** (JavaScript) para el frontend. Esto permite reducir costos y mantener la flexibilidad del sistema.
* **Base de datos:** Se utilizará una base de datos **relacional** soportada por el servicio de hosting, como **MySQL**, para gestionar la información de usuarios, barberías, horarios y citas. Esta elección se debe a la necesidad de un sistema que maneje datos estructurados con integridad relacional, y es compatible con el entorno de hosting seleccionado
* **Lenguaje de programación**: El backend de *BarberConnect* será desarrollado en **Python**, utilizando el framework **Django** para manejar la lógica de negocio y la comunicación con la base de datos. Para el frontend, se utilizará **JavaScript** con el framework **React**, asegurando una interfaz de usuario interactiva y responsiva. Esta combinación de tecnologías es compatible con el hosting elegido y facilita la escalabilidad futura del sistema.

# Arquitectura Lógica

A continuación, se presenta una vista lógica de la aplicación expresado en dos diagramas, uno de ellos que muestra la parte estructural o estática de la aplicación (módulos), y otra vista que representa la parte dinámica (componentes y conectores).

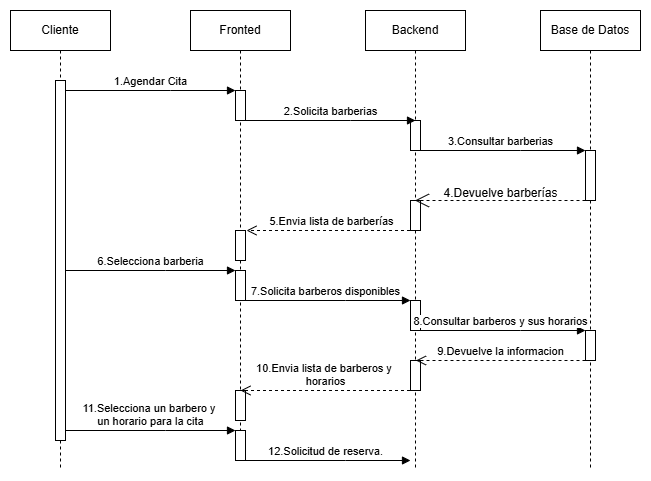
**Parte Estructural**

En el siguiente diagrama de clases se observa que el principal módulo….

***Ilustración 1: Diagrama de clases***

**Parte Dinámica**

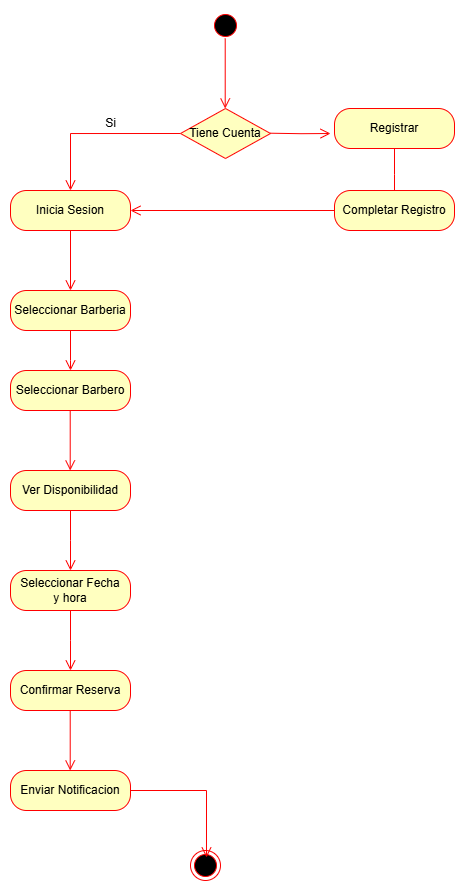
La parte dinámica….



***Ilustración 2: Diagrama de secuencia***

# Arquitectura de Procesos

A continuación, se muestra una vista de procesos, en la cual se observa que:



***Ilustración 3: Diagrama de actividad***

# Arquitectura de desarrollo

En esta vista se aprecia que existirán dos módulos principales que contendrán distintas funcionalidades de la aplicación. A continuación, se describen:

**A continuación se presenta el diagrama de componentes:**

La relación entre las interfaces, las clases y los datos.

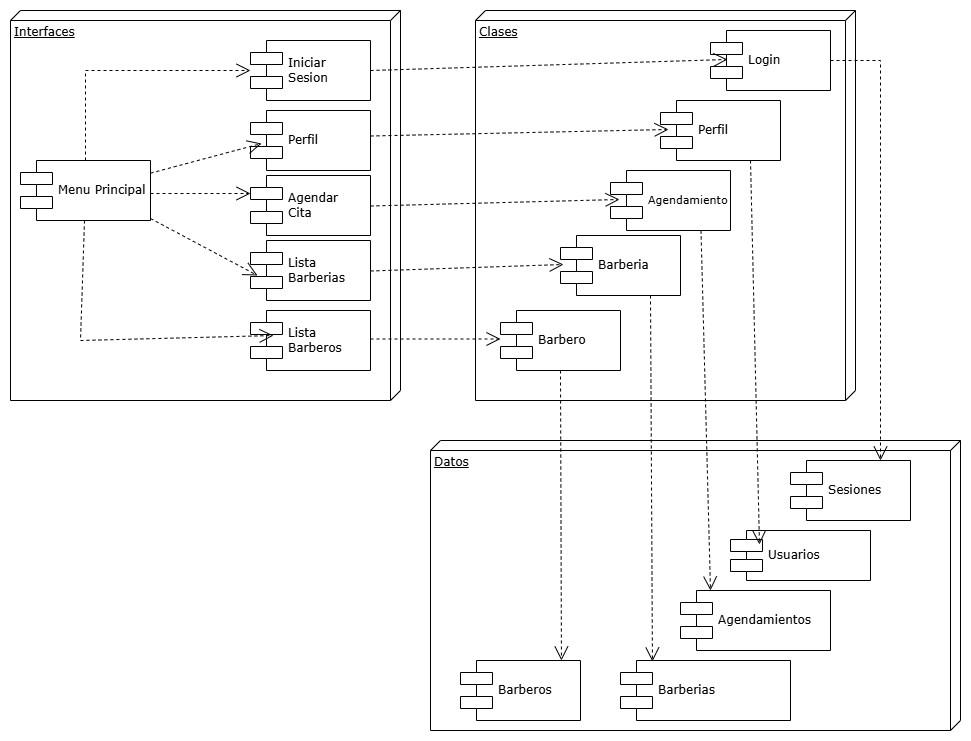
La interfaz de iniciar sesión se comunica con la clase iniciar sesión para que haga la búsqueda

en la tabla de usuarios.

Luego la interfaz llamaría al menú, el menú llamara a las interfaces según donde le damos clic luego.

la interfaz de perfil se comunica con la clase perfil para ver la edad que tengo.

la interfaz de barberías se comunica con la clase barberías para consultar los locales disponibles.

la interfaz de Agendar Cita se comunica con la clase Agendar Cita, para ver las citas disponibles.

***Ilustración 4: Vista de Implementación (Diagramas de componentes o paquetes)***

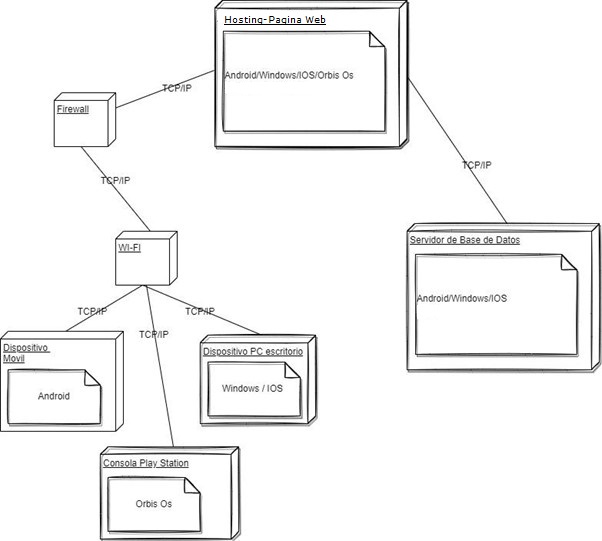
# Arquitectura física

En esta vista se despliegan los nodos que participan con el sistema. Los nodos principales son los nodos Servidor de Integración. Características a continuación:

**1.A continuación vemos el diagrama de despliegue:**

El dispositivo móvil tiene que conectarse al wifi, pasa el muro de seguridad del firewall

y luego ingresa al servidor de la aplicación y a su vez se conectará a la base de datos y este es el concepto de MVC modelo vista controlador.

**

***Ilustración 6: Diagrama de Despliegue***

# Escenarios

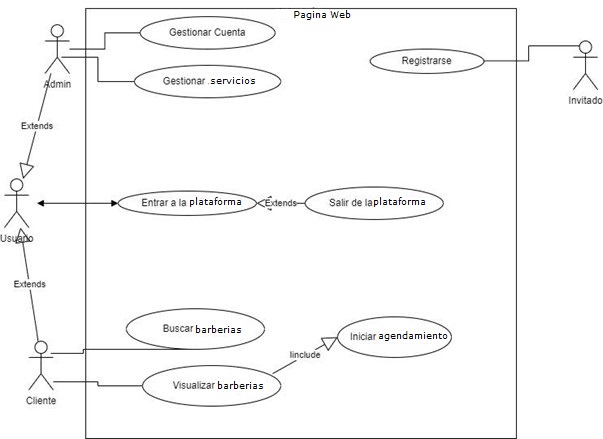
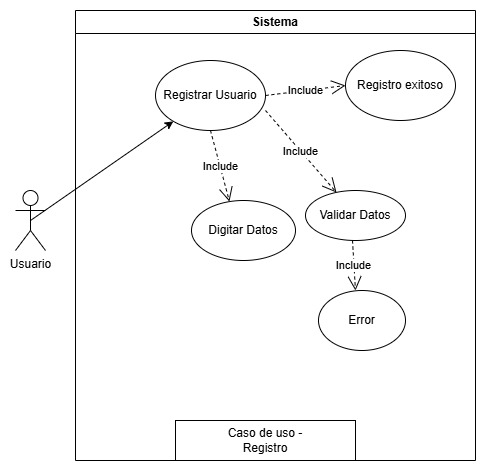
Los escenarios son útiles para especificar los atributos de calidad, es un requerimiento especifico a una sola categoría de atributo de calidad compuesto por 6 partes, se tiene varios escenarios de atributos de calidad, de acuerdo con que atributo queremos atacar del sistema; estos tienen ciertos pasos que d sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis.

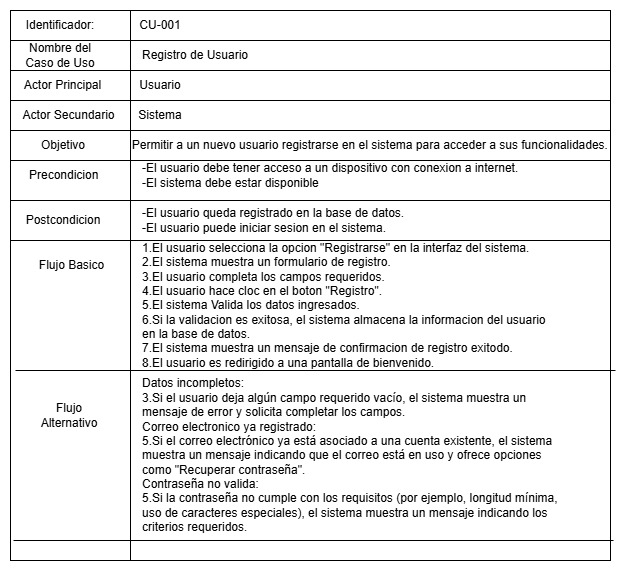
Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

# Modelo de Casos de Uso

El modelo de casos de uso puede ser encontrado en el documento “Casos de Uso”.

***En este primer caso se aborda las funcionalidades más relevantes de los distintos tipos de usuarios.***

**



**Especificación de Casos de Uso Relevantes**

Los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados. Los criterios usados para dicha determinación fueron:

* Su implementación implica varios nodos de la vista de despliegue.
* Su implementación es de alto riesgo.
* Incluye muchos conceptos y relaciones del dominio.
* Incluye posibles escenarios críticos de calidad.

# Especificación de los Escenarios de Calidad Relevantes

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

Los escenarios están compuestos de 6 pasos y se especifica cada uno de ellos, en los siguientes escenarios de calidad asignados para poder especificar de manera más detalla los atributos de calidad.

Los escenarios seleccionados son los siguientes:

**Escenario de calidad: Portabilidad de la aplicación.**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

|  |  |
| --- | --- |
| Escenario de Calidad N° | |
| Atributo de Calidad Asociado (Característica): Portabilidad.  Subcaracterística: Adaptabilidad. | |
| Descripción: La aplicación de celular debe ser compatible en las distintas plataformas existentes. | |
| Fuente del Estímulo: | Usuario. |
| Estimulo: | Acceder a la aplicación de celular. |
| Artefacto: | Aplicación. |
| Ambiente: | Explotación. |
| Respuesta: | Debe funcionar de manera correcta en cualquier tipo de dispositivo y plataformas de softwares distintas existentes. |
| Medida de Respuesta: | Tiene que ser probado en distintos, dispositivos, navegadores y sistemas operativos. |

**Escenario de calidad: Buscador.**

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente**

|  |  |
| --- | --- |
| Escenario de Calidad N° | |
| Atributo de Calidad Asociado (Característica): Eficiencia de desempeño.  Subcaracterística: Comportamiento temporal. | |
| Descripción: El usuario usa el buscador para encontrar barberías. | |
| Fuente del Estímulo: | Usuario. |
| Estimulo: | Realizar una consulta en el buscador. |
| Artefacto: | En todo el sistema. |
| Ambiente: | Aplicación finalizada y operativa. |
| Respuesta: | Muestra el resultado de la consulta. |
| Medida de Respuesta: | Tiempo no mayor a 4 segundos. |

Escenario de calidad: Ingreso fallido a la cuenta.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

|  |  |
| --- | --- |
| Escenario de Calidad N° | |
| Atributo de Calidad Asociado (Característica): Seguridad.  Subcaracterística: Confidencialidad. | |
| Descripción: Un usuario desconocido intento acceder a la cuenta de otra persona con fines maliciosos. | |
| Fuente del Estímulo: | Usuario desconocido. |
| Estimulo: | Introduce el nickname y contraseñas incorrectas. |
| Artefacto: | Operación normal. |
| Ambiente: | Condición Normal. |
| Respuesta: | Registro de intento en la base de datos y bloqueo de la cuenta. |
| Medida de Respuesta: | Después de 4 intentos. |

# Tamaño y desempeño

Las principales decisiones arquitectónicas para *BarberConnect* se tomaron considerando la restricción de **Tiempo de Construcción**. Dado que el proyecto debe completarse en un plazo ajustado de 6 semanas, se priorizó la adopción de una arquitectura conocida y de bajo riesgo en su implementación, utilizando tecnologías como **Django** para el backend y **React** para el frontend. Esto asegura una curva de aprendizaje mínima para el equipo y un desarrollo eficiente.

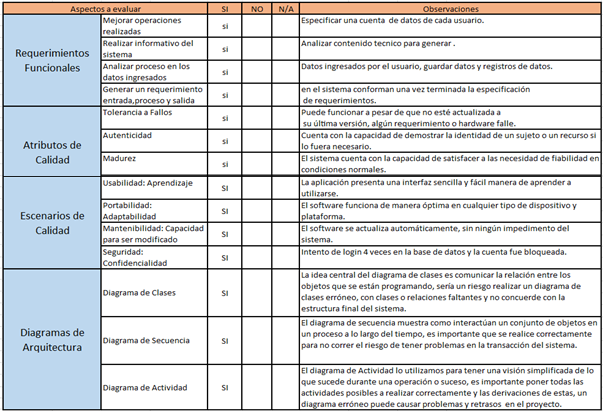
La arquitectura se diseñó de manera modular, con el objetivo principal de separar las responsabilidades. Esta modularización no solo facilita la paralelización del trabajo, permitiendo que los distintos componentes puedan desarrollarse de manera independiente, sino que también habilita el testeo unitario de cada módulo. Al garantizar que cada parte del sistema sea testeable de manera individual, reducimos la tasa de fallos y mejoramos la calidad global del sistema.

Además, la arquitectura se diseñó considerando la **Infraestructura** disponible y las necesidades de **Tolerancia a Fallos**. El sistema utilizará un servicio de hosting que no ofrece replicación ni balanceo de carga en esta fase, lo que nos llevó a estructurar la aplicación de manera que pueda ser escalada en el futuro con servicios adicionales si es necesario.

El escenario de calidad relacionado con la **Mantenibilidad** nos guio hacia un diseño que prioriza la separación de responsabilidades, siguiendo patrones de diseño como el **patrón provider**. Esto permite que *BarberConnect* delegue la obtención de información a sistemas externos a través de piezas de software desacopladas, facilitando su extensibilidad en el futuro. Este enfoque asegura que, en caso de que se necesiten integrar nuevos servicios o modificar los actuales, el sistema pueda adaptarse sin grandes reestructuraciones.

Por último, se eligió una estrategia simple y efectiva para manejar las notificaciones de agendamiento, mediante el envío automático de correos electrónicos cuando se agenda una nueva cita. Esto asegura que tanto los clientes como los barberos reciban notificaciones de manera oportuna sin necesidad de intervención manual. Esta estrategia garantiza que el sistema funcione de manera fluida y minimiza el riesgo de fallos en la comunicación, asegurando una experiencia eficiente para todos los usuarios.

# Identificando Riesgos

**

En los análisis retrospectivos de los proyectos evaluados se identificaron falencias en la gestión de riesgos y esto para los proyectos tecnológicos, específicamente para el desarrollo de software, puede significar problemas y deficiencias en las etapas de planeación y ejecución, así como para dar respuesta a hechos inesperados, que son inherentes a la naturaleza de estos proyectos porque están en entornos de alta incertidumbre.

En línea con esto, según un estudio realizado por el Project Management Instituto (PMI), la mala administración del riesgo es un factor determinante que obstaculiza el logro de los objetivos de un proyecto. Por eso, la gestión de riesgos en proyectos de software cada vez cobra mayor relevancia como mecanismo de anticipación y aprovisionamiento para tener una gestión de proyectos más proactiva que reactiva y de esta forma, poder incrementar la tasa de éxito en todas sus ejecuciones.

# Patrones de Diseño

PATRON DECORATOR:

**Decorator** es un patrón de diseño estructural que te permite añadir funcionalidades a objetos colocando estos objetos dentro de objetos encapsuladores especiales que contienen estas funcionalidades.

¿Y en la práctica cómo funciona?

Existen cuentas premium la cual se dividen en plus o Gold; Cuando una persona se registra, la plataforma web le pregunta si desea ser usuario premium o no, entonces como ese escenario es opcional, como esto no se va a dar para todos los usuarios, algunos aceptan y otros no, podemos definir una clase normal y una clase decorada con esa característica.

# ¿Y cómo lo podemos aplicar a nuestro código?

PASO 1:

Creamos la clase usuario con algunos atributos, nombre y créditos, creamos 2 objetos de esa clase.

Texto

Descripción generada automáticamente

PASO2:

Creamos el decorador UsuarioPremiums, el cual agregara una característica al usuario normal, sin necesidad de modificar la clase usuario, se crea la clase usuario plus y usuario Gold, que heredaran del usuario premium.

Texto

Descripción generada automáticamente

PASO 3:

Aquí creamos 2 objetos:

Texto

Descripción generada automáticamente

Y veamos el resultado:

Texto

Descripción generada automáticamente

En el resultado vemos que José es un usuario normal, pero maría es un usuario pluspremium implementadose correctamente el patrón DECORATOR

Patron Builder:

**Builder** es un patrón de diseño creacional que nos permite construir objetos complejos paso a paso. El patrón nos permite producir distintos tipos y representaciones de un objeto empleando el mismo código de construcción.

¿Y en la práctica cómo funciona?

¡¡¡Construye!!!, Es el primero que se ejecuta cuando creamos un nuevo objeto o una nueva instancia de nuestra clase, es el encargado de inicializar nuestros objetos con ciertas características o reglas que nosotros especifiquemos y menos líneas de código.

Patron Singleton:

**Singleton** es un patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia.

¿Y en la práctica cómo funciona?

Tener una única instancia de una clase durante toda aplicación por ende tiene que ser privado ya que si fuera pública podría crearse más instancia de esa clase, imaginemos que existe una lista de comunas que mostrar al registrarse, no tendría sentido devolver a cada usuario que se registre en el sistema la lista de comunas, sería mejor tener un único objeto el cual sea instanciado una única vez y por cada usuario que necesite esta información y esto en consumo de memoria va a ser mucho mejor.

¿Y cómo lo podemos aplicar a nuestro código?

PASO 1:

En este código creamos una clase singleton, la cual parece que creamos 5 objetos, pero en realidad es el mismo, y se imprime 5 veces, pero es una sola instancia, y para poder confirmar estos podemos apoyarnos de la palabra reservada “is”, ya que podemos conocer si un objeto es otro objeto o el mismo.

Texto

Descripción generada automáticamente

PASO 2:

Como se ve el resultado, pareciera que son 5 objetos, pero en realidad es el mismo, cuando hacemos la comparación en: print (a is b is c is d is e) nos da como resultado “True”.

Texto

Descripción generada automáticamente

EVALUACION APLICANDO MODEL ATAM:

# Fase 1:

## 1.PRESENTACION DE ATAM:

En este estudio se establece la evaluación arquitectónica del diseño del software asociado al funcionamiento de nuestra plataforma de agendamiento para barberías, *BarberConnect*. En este paso, se presenta el modelo ATAM a los stakeholders, utilizando este tiempo para explicar el proceso que se seguirá para la evaluación de la arquitectura del sistema. Se fijan las expectativas de las actividades y se contextualiza el modelo con las necesidades del proyecto *BarberConnect*, estableciendo cómo se evaluará la eficiencia, seguridad, y escalabilidad de la arquitectura del sistema.

## 2.PRESENTACION DE LOS OBJETIVOS DE NEGOCIOS:

El líder del proyecto presentó los objetivos de negocio a los stakeholders, tanto desde el punto de vista de los desarrolladores como desde la perspectiva de los usuarios y servicios que se pretende brindar. Para los desarrolladores, el software de *BarberConnect* debe considerar los siguientes conductores del sistema:

* + **Administración de la información** de los usuarios y barberías, asegurando la integridad y disponibilidad de los datos.
  + **Control y auditoría de información** en la base de datos para garantizar la precisión y seguridad.
  + **Depuración de información** en casos de inconsistencias o colisiones en los datos gestionados.
  + **Soporte para los módulos principales del sistema** (reservas, notificaciones, perfil de usuarios/barberos).
  + **Facilidad de creación de nuevos módulos** en función de los requerimientos futuros de los usuarios, como la integración de pagos online.
  + **Manejo de seguridad** a nivel de acceso a datos sensibles de usuarios y barberos, siguiendo los estándares de confidencialidad e integridad.
  + **Dimensionamiento de las capacidades del sistema** para evitar cuellos de botella en el agendamiento masivo de citas.
  + **Capacidad de gestionar múltiples conexiones simultáneas** en el sistema, especialmente durante horas de alta demanda.

Los atributos de calidad de mayor prioridad para la plataforma incluyen:

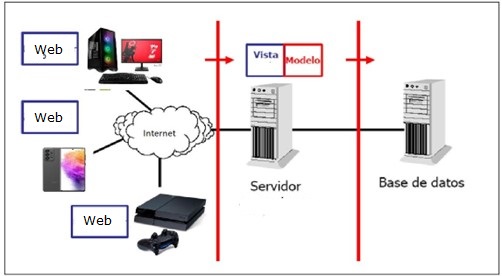
* + **Desempeño**: El sistema debe ofrecer tiempos de respuesta rápidos para la búsqueda de barberías y la confirmación de citas, asegurando una experiencia fluida para los usuarios.
  + **Usabilidad**: La interfaz debe ser fácil de entender y usar por los clientes y barberos, minimizando la curva de aprendizaje y maximizando la eficiencia.
  + **Seguridad**: Dada la importancia de la información personal de los usuarios (como nombres, correos, y preferencias), el sistema debe garantizar la **confidencialidad e integridad** de los datos mediante cifrado y autenticación segura.
  + **Disponibilidad**: El sistema debe estar disponible en todo momento, asegurando que las reservas puedan realizarse sin interrupciones incluso durante horas pico.
  + **Modificabilidad**: La arquitectura de *BarberConnect* debe ser flexible para incorporar nuevos escenarios, entidades o funcionalidades, como la expansión a nuevas regiones o la integración con nuevas pasarelas de pago.
  + **Escalabilidad**: El sistema debe ser lo suficientemente flexible para crecer en capacidad, tanto en términos de usuarios concurrentes como en la posibilidad de añadir nuevas barberías o servicios sin afectar el rendimiento general del sistema.

## 3.PRESENTACION DE LA ARQUITECTURA:

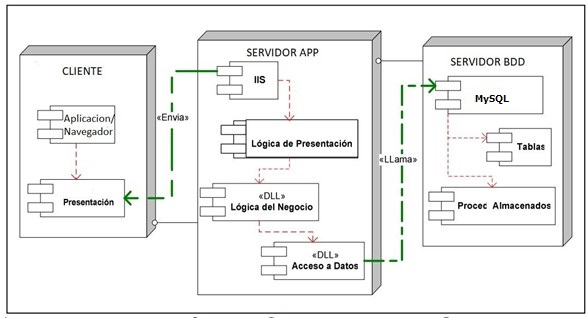
Durante las interacciones que tuvo el equipo de evaluación con los desarrolladores y líderes del proyecto *BarberConnect*, surgieron varias vistas y propuestas arquitectónicas. Los resultados clave se resumen a continuación:

1. **Vistas arquitectónicas:**
   * **Herramientas de desarrollo**: El software está siendo desarrollado utilizando **Django** para el backend, y **HTML**, **CSS** y **React** para el frontend. Esta combinación garantiza un desarrollo ágil, modular y compatible con las necesidades de la plataforma.
   * **Manejo de datos**: Se ha decidido utilizar **MySQL** como motor central de base de datos en el entorno del hosting. MySQL fue elegido por su escalabilidad y robustez para manejar la información estructurada relacionada con usuarios, barberías y citas.
   * **Metodología orientada a objetos**: El sistema sigue un enfoque orientado a objetos, lo que proporciona flexibilidad en la configuración y facilita el mantenimiento del sistema.
   * **Requisitos del navegador**: El sistema es compatible con navegadores modernos como **Google Chrome**, **Firefox**, **Microsoft Edge**, y otros navegadores actuales, asegurando su accesibilidad desde diversos dispositivos y plataformas.
2. **Estilos Arquitectónicos**

* **Arquitectura de 3 capas**: *BarberConnect* sigue una **arquitectura de tres capas**, que se divide en:
  + **Capa de presentación**: Incluye el frontend desarrollado en React, que maneja la interfaz de usuario, y presenta todas las interacciones necesarias para agendar citas y gestionar perfiles de usuarios y barberías.
  + **Capa lógica o de aplicación**: Desarrollada en **Django** (Python), esta capa gestiona la lógica de negocio, como la validación de reservas, la gestión de usuarios, y las notificaciones automáticas.
  + **Capa de datos**: Implementada con **MySQL**, la base de datos almacena toda la información del sistema, como detalles de los usuarios, barberías, citas y servicios ofrecidos.



* **Distribución en nodos**: El sistema se distribuye en tres nodos principales:
* **Cliente (navegador web)**: Los usuarios interactúan con *BarberConnect* a través de un navegador web, enviando solicitudes al servidor web.
* **Servidor Web**: El servidor web, utilizando **Django**, gestiona todas las solicitudes, procesa la lógica de negocio y devuelve respuestas al cliente.
* **Servidor de Base de Datos**: Almacena y gestiona los datos relacionados con los usuarios, barberías y citas. Este servidor es accesado por el servidor web a través de consultas SQL.



* **Estilo de llamada y retorno**: La arquitectura de *BarberConnect* utiliza el estilo de **Llamada y Retorno**, típico en arquitecturas en capas. Cada solicitud del cliente (usuario) genera una llamada al servidor, que procesa la solicitud y devuelve los resultados correspondientes.
* **Procedimientos almacenados**: A nivel de la base de datos, se emplean **procedimientos almacenados** para optimizar las consultas y operaciones comunes como la creación, actualización o eliminación de citas y perfiles. Además, se utilizan **librerías** a nivel de código fuente, que se han depurado a lo largo del desarrollo de la aplicación.
* **Clases y funciones**: Se utilizan clases y funciones dentro del código para invocar los procedimientos almacenados, lo que permite realizar consultas, actualizaciones y eliminaciones de información de manera eficiente a través de las interfaces de usuario.

1. **Requerimientos Arquitectónicos**

Los siguientes **requerimientos arquitectónicos** guían el diseño y desarrollo de *BarberConnect*:

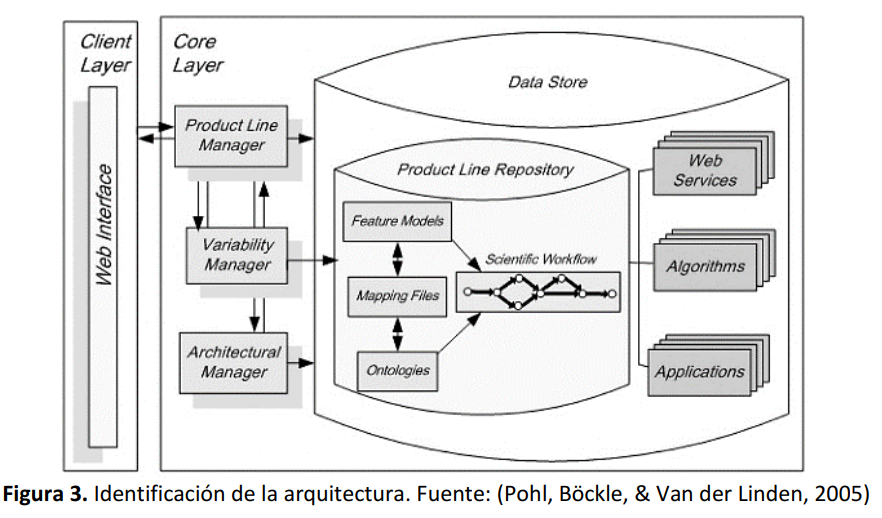
* **Disponibilidad**: El sistema debe ser altamente disponible, asegurando que los usuarios puedan acceder y reservar citas en cualquier momento. En caso de fallos, el sistema debe minimizar el tiempo de inactividad y garantizar que las operaciones críticas, como la gestión de citas, no se vean afectadas.
* **Modificabilidad**: *BarberConnect* debe ser fácilmente modificable para incorporar nuevas funcionalidades y requisitos en el futuro, como la posibilidad de añadir nuevas barberías, servicios adicionales o incluso expandir la plataforma a nivel regional. La estructura orientada a objetos y modular facilita estas modificaciones.
* **Desempeño**: El desempeño es crítico, especialmente durante picos de demanda cuando múltiples usuarios intenten reservar citas al mismo tiempo. El sistema debe manejar estas solicitudes de manera eficiente, proporcionando tiempos de respuesta rápidos para evitar demoras en la confirmación de citas.
* **Seguridad**: La seguridad es un aspecto clave, ya que *BarberConnect* maneja información personal de los usuarios y datos sensibles de las barberías. El sistema debe prevenir accesos no autorizados, garantizando la confidencialidad e integridad de la información a través de medidas de seguridad como la autenticación de usuarios y el cifrado de datos.

# FASE 2:

## 4.IDENTIFICACION DE PATRONES ARQUITECTONICOS:

Después de la presentación de la arquitectura del software de *BarberConnect*, se enumeraron las propuestas arquitectónicas obtenidas a partir de entrevistas con los desarrolladores y líderes del proyecto, así como del análisis de la documentación durante la etapa de preevaluación. Las propuestas arquitectónicas identificadas se enumeran a continuación, aunque no se profundiza en su análisis en esta fase:

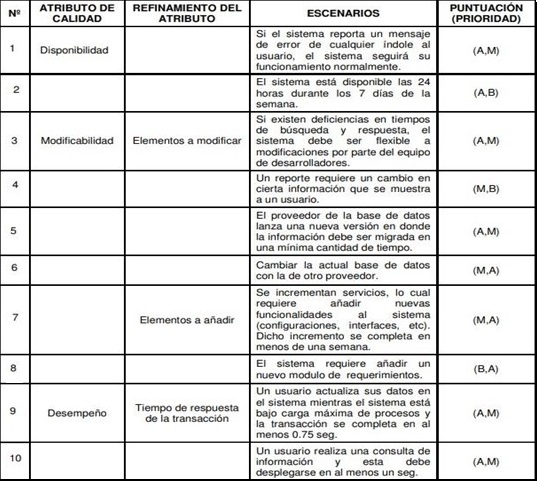
* + **Arquitectura de 3 capas**: El sistema está estructurado en tres capas principales:
* **Capa de presentación**: El frontend desarrollado en **React**, que proporciona la interfaz interactiva para los usuarios.
* **Capa lógica o de negocio**: Gestionada por el framework **Django** (Python), encargada del procesamiento de las reglas de negocio, como la gestión de citas y usuarios.
* **Capa de datos**: Gestionada por una base de datos **MySQL**, donde se almacenan todos los datos relacionados con usuarios, barberos, barberías y citas.
  + **Procesamiento de transacciones en un ambiente cliente/servidor**: El sistema maneja las transacciones de agendamiento de citas y gestión de usuarios a través de un entorno de **cliente/servidor**, donde las solicitudes se envían desde el cliente (usuario) al servidor, que procesa la solicitud y devuelve la respuesta.
  + **Arquitectura orientada a objetos**: *BarberConnect* utiliza un enfoque orientado a objetos en su diseño, lo que permite una fácil extensión y mantenimiento del código, así como una estructura modular que separa claramente las responsabilidades entre los componentes del sistema.
  + **Enlaces HTML entre módulos**: Las diferentes secciones y módulos de la plataforma (como la gestión de perfiles, citas y servicios) están conectadas mediante **enlaces HTML** y rutas bien definidas, lo que facilita la navegación y la interacción del usuario con el sistema.
  + **Arquitectura de datos centralizada**: Todos los datos del sistema están centralizados en una base de datos **MySQL**, que se considera una solución confiable y escalable para gestionar la información del sistema de manera eficiente.
  + **Estilo arquitectónico de llamada y retorno**: *BarberConnect* utiliza un estilo de **llamada y retorno** en su arquitectura, típico de las arquitecturas basadas en capas. Esto significa que las solicitudes de los usuarios generan llamadas al servidor, que procesa la lógica de negocio y devuelve la respuesta adecuada, siguiendo un ciclo claro de procesamiento.



## 5.GENERACION DEL ARBOL DE UTILIDAD:

En esta etapa se identifican, priorizan y redefinen las metas de algunos atributos de calidad de la arquitectura de *BarberConnect*. La generación del árbol de utilidad es el resultado de la priorización de los requerimientos específicos relacionados con los atributos de calidad, observados como escenarios clave del sistema.

A continuación, se presenta el árbol de utilidad generado durante la aplicación del modelo ATAM para *BarberConnect*. En este árbol, se identifican los principales atributos de calidad necesarios para la arquitectura del software, cada uno de los cuales se refina con su correspondiente valor de prioridad (Alto, Medio, Bajo).



## 6.ANALIZAR LOS ENFOQUES ARQUITECTONICOS

En este paso, se analizan los escenarios más importantes de acuerdo con el árbol de utilidad generado para *BarberConnect*, estableciendo las decisiones arquitectónicas que se deben tomar para cada escenario, junto con sus puntos de sensibilidad, riesgos, no riesgos, y tradeoffs (cuando la decisión afecta a más de una característica de calidad). Dado que algunos de los escenarios implican cambios, se analiza el impacto de estos cambios en la arquitectura del sistema.

A continuación, se detallan los puntos de sensibilidad, riesgos, no riesgos, y tradeoffs con respecto al análisis de los escenarios más importantes identificados en el árbol de utilidad de *BarberConnect*:

**Escenario 1.**

Si el sistema reporta un mensaje de error de cualquier índole al usuario, el sistema debe seguir funcionando normalmente.

* + S**1**: La **disponibilidad** del sistema es sensible al número y frecuencia de errores generados. Si no se manejan adecuadamente, los errores pueden afectar la experiencia del usuario y la operatividad del sistema. Este es un punto de sensibilidad importante que impacta negativamente la disponibilidad.
  + **R2**: Si el sistema produce errores sin un mecanismo adecuado de captura y manejo de errores, esto representa un riesgo ya que puede dejar de estar disponible para los usuarios. En *BarberConnect*, es esencial implementar un manejo robusto de errores para asegurar que el sistema siga siendo funcional y no se vea afectado por fallos menores.
  + **No riesgo**: Un adecuado manejo de errores con recuperación automática y notificaciones internas evitaría que pequeños errores afecten la **disponibilidad** del sistema.

**Escenario 2.**

Si existen deficiencias en tiempos de búsqueda y respuesta, el sistema debe ser flexible a modificaciones por parte del equipo de desarrolladores.

* **S2**: Optimizar los tiempos de búsqueda es sensible al número de elementos involucrados y la complejidad de los procedimientos. Este es un punto de sensibilidad que afecta la **modificabilidad** del sistema. Dependiendo del número de elementos a modificar, los cambios en los procedimientos pueden mejorar o empeorar la modificabilidad.
* **R3**: El acceso masivo de usuarios en un momento determinado puede generar un riesgo en los tiempos de respuesta del sistema. Si no se mantiene una calidad óptima en los algoritmos de búsqueda y en la gestión de la conectividad, los tiempos de respuesta pueden verse afectados negativamente. En *BarberConnect*, es crucial garantizar que los procedimientos de búsqueda estén optimizados para soportar altos volúmenes de solicitudes.
* **Tradeoff**: Mejorar los algoritmos de búsqueda puede aumentar la **modificabilidad**, pero puede generar un impacto en el **desempeño** y requerir una mayor optimización de los recursos del servidor.

**Escenario 3.**

El proveedor de la base de datos lanza una nueva versión en donde la información debe ser migrada en una mínima cantidad de tiempo.

* + **S3**: Cambiar a una nueva versión del motor de base de datos es sensible a la compatibilidad entre las funciones de la versión actual y las nuevas herramientas de la versión actualizada. Este es un punto de sensibilidad que afecta la **modificabilidad** del sistema, ya que los cambios en el motor de base de datos pueden afectar las consultas existentes y requerir ajustes significativos.
  + **R4**: Migrar a una nueva versión del motor de base de datos sin una planificación adecuada representa un riesgo de pérdida de datos o interrupciones en el servicio. En *BarberConnect*, es fundamental planificar y probar la migración para minimizar el tiempo de inactividad y garantizar que el sistema siga siendo operativo.
  + **Tradeoff**: La migración a una nueva versión del motor de base de datos puede mejorar el **desempeño** y la **escalabilidad**, pero también puede impactar negativamente la **modificabilidad** y aumentar la complejidad del sistema si no se gestionan adecuadamente las diferencias entre versiones.

# FASE3:

## 7.PRIORIZACION DE ESCENARIOS

En este paso, los stakeholders y el equipo de desarrollo participaron en una lluvia de ideas para identificar los escenarios más relevantes para *BarberConnect*. Se discutieron escenarios de casos de uso, escenarios de crecimiento, y escenarios exploratorios. Luego, estos escenarios se priorizaron mediante un proceso de votación en el que a cada participante se le asignó un número de votos equivalente al 30% del número total de escenarios propuestos.

El equipo de *BarberConnect*, siendo altamente productivo, generó muchos escenarios a través de este ejercicio, algunos de los cuales ya estaban incluidos en el árbol de utilidad pero no se habían analizado a fondo. Los involucrados expresaron sus opiniones sobre la importancia de los escenarios, lo que permitió que algunos de ellos adquirieran mayor relevancia que en las fases anteriores.

A continuación, se presenta una tabla con una selección de los escenarios más importantes surgidos durante este proceso de priorización. Cabe destacar que algunos escenarios no están completamente estructurados, lo que refleja la espontaneidad del ejercicio de la lluvia de ideas. Si algún escenario necesita una aclaración adicional, se realiza con la ayuda de la persona que lo propuso antes de proceder a la votación.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

.

## 8 ANALISS DE ENFOQUES:

Tras obtener los resultados de la votación de la lluvia de ideas en el paso 7, se ha encontrado que algunos de los escenarios con mayor prioridad coinciden con los descritos previamente en el árbol de utilidad. Sin embargo, también se identificaron nuevos escenarios que deben ser evaluados y considerados en la arquitectura de *BarberConnect*.

De acuerdo con los resultados del proceso de investigación y evaluación del software, se establece que la priorización del diseño debe centrarse en dos elementos clave:

* **Soporte para múltiples usuarios concurrentes**:

La plataforma de *BarberConnect* debe estar diseñada para soportar una gran cantidad de usuarios conectados simultáneamente. A medida que la aplicación crece y se populariza, será crucial que el sistema mantenga su capacidad de gestionar miles de reservas y consultas en tiempo real sin que esto degrade el rendimiento.

* **Conexión estable y rápida**:

Mantener una conexión estable y rápida es fundamental para asegurar una experiencia de usuario positiva. Los tiempos de respuesta, tanto para la búsqueda de barberías cercanas como para la confirmación de citas, deben ser mínimos. Si el sistema no puede ofrecer una experiencia fluida, los usuarios pueden optar por no utilizarlo.

# FASE4:

## 9.PRESENTACION DE RESULTADOS:

Luego de aplicar el modelo ATAM al software de *BarberConnect*, se obtuvieron las siguientes observaciones clave sobre la arquitectura y el proceso de desarrollo:

* + **Estado de la documentación del sistema**:
    - La documentación del sistema está **completa y bien organizada**. Durante la Fase 1 (Presentación) y el análisis de escenarios, se identificó que la documentación incluye detalles cruciales como:
    - Diagramas de casos de uso (vía UML).
    - Descripción detallada de las capas del sistema (capa de presentación, capa lógica y capa de datos).
    - Estructura de la base de datos, donde se especifican los modelos de usuario, barbero, cita y barbería.
* Contar con esta documentación es fundamental para asegurar que el sistema cumpla con los **requerimientos funcionales y de calidad**, así como para llevar a cabo pruebas adecuadas y garantizar un mantenimiento eficiente.
* **Evaluación arquitectónica productiva**:
  + - Gracias a la existencia de una documentación rigurosa, fue posible llevar a cabo una evaluación arquitectónica de *BarberConnect* de manera **más completa y productiva**.
    - Las entrevistas con los líderes del proyecto y el equipo de desarrollo aumentaron la cantidad de documentación generada, lo que resultó en una **mejora de la calidad arquitectónica**. La interacción constante entre los stakeholders permitió identificar áreas clave que requieren atención.
  + **Mejora de la comunicación y comprensión de requerimientos**:
    - Uno de los beneficios más destacados de la evaluación arquitectónica fue la **mejora en la comunicación** entre los diferentes involucrados (stakeholders, desarrolladores, líderes de proyecto, etc.). Esto resultó en una mejor comprensión de los requerimientos funcionales y no funcionales en los que se basa *BarberConnect*.
    - A menudo, **nuevos requerimientos** surgieron como resultado de la evaluación, lo que permitió ajustes y mejoras en la arquitectura del sistema.
  + **Uso de patrones de diseño**:
    - Durante el desarrollo del sistema, cada integrante del equipo de desarrollo se encargó de la implementación de módulos de manera independiente, utilizando **patrones de diseño** en función de sus conocimientos y la naturaleza del módulo.
    - Estos patrones de diseño ayudaron a garantizar que la arquitectura del sistema fuese modular y flexible, facilitando futuras modificaciones y el mantenimiento del sistema.
  + **Definición del diseño a alto nivel**:
    - En la arquitectura se enfatizó la necesidad de definir el diseño a un **nivel alto**, donde se establezcan las **restricciones**, la estructura organizacional del equipo, el razonamiento sobre posibles cambios, y una estimación clara de los **costos** y el **esfuerzo** necesario para el proyecto.
    - Este enfoque ayuda a garantizar que las decisiones arquitectónicas sean escalables y puedan manejar el crecimiento del sistema a medida que se añadan nuevas barberías o funcionalidades.
  + **Importancia del proceso de backups**:
    - Durante la evaluación, se identificó la importancia de manejar un **proceso de backups** adecuado para la base de datos central de *BarberConnect*. Esto incluye definir horarios y medios apropiados para realizar copias de seguridad.
    - Un proceso eficiente de backups garantizará la **seguridad de los datos** y la **continuidad del servicio**, ayudando a evitar la pérdida de información valiosa y contribuyendo al desempeño estable del sistema.

# Metodología de Modelado

Análisis de Requerimientos

Se realizó un análisis detallado de los requerimientos funcionales del sistema para identificar las entidades clave y sus relaciones. Esto incluyó:

* Definir las necesidades del sistema en términos de datos a almacenar y su relación.
* Determinar las funcionalidades específicas requeridas, como agendar citas, gestionar barberos y enviar notificaciones.

# Diseño del Modelo Entidad-Relación

A partir del análisis, se construyó un modelo Entidad-Relación (ER) que incluyó las principales entidades:

1. **Clientes**: Usuarios del sistema.
2. **Barberos**: Profesionales que prestan servicios en las barberías.
3. **Servicios**: Ofertas específicas como cortes de cabello y perfilado de barba.
4. **Barberías**: Locales que brindan los servicios.
5. **Agendamiento**: Registros de citas entre clientes y barberos.
6. **Notificaciones**: Mensajes enviados a los clientes para recordar citas u ofrecer promociones.

Cada entidad se describió con sus atributos principales, como se detalla a continuación:

**Ejemplo de Entidad Cliente**

* idClientes (clave primaria): Identificador único.
* nombre: Nombre completo del cliente.
* correo: Correo electrónico del cliente.
* telefono: Número de contacto.

**Relaciones Definidas**

* Cada barbería puede tener múltiples barberos asociados.
* Un cliente puede realizar múltiples agendamientos.
* Cada agendamiento está vinculado a un cliente, un barbero, un servicio y una barbería.

# Implementación en MySQL

El diseño ER se tradujo a un modelo relacional implementado en MySQL. A continuación, se detalla el proceso seguido:

**Creación de Tablas**

Se diseñaron tablas para cada entidad identificada. Un ejemplo es la tabla Clientes:

sql

Copiar código

CREATE TABLE barberiasdb.Clientes (

idClientes INT NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(100) NOT NULL,

correo VARCHAR(100) NOT NULL UNIQUE,

telefono VARCHAR(15) NULL,

PRIMARY KEY (idClientes)

) ENGINE = InnoDB;

# Definición de Relaciones

Las claves foráneas se establecieron para mantener la integridad referencial. Un ejemplo es la tabla Agendamiento, que relaciona múltiples entidades:

sql

Copiar código

CREATE TABLE barberiasdb.Agendamiento (

idAgendamiento INT NOT NULL AUTO\_INCREMENT,

idcliente INT NOT NULL,

idbarbero INT NOT NULL,

idservicio INT NOT NULL,

idbarberia INT NOT NULL,

fecha\_cita DATETIME NOT NULL,

PRIMARY KEY (idAgendamiento),

FOREIGN KEY (idcliente) REFERENCES barberiasdb.Clientes (idClientes),

FOREIGN KEY (idbarbero) REFERENCES barberiasdb.Barberos (idBarberos),

FOREIGN KEY (idservicio) REFERENCES barberiasdb.Servicios (idServicio),

FOREIGN KEY (idbarberia) REFERENCES barberiasdb.Barberias (idBarberia)

) ENGINE = InnoDB;

# Índices y Optimización

Para mejorar el rendimiento, se añadieron índices en campos utilizados frecuentemente en las consultas, como:

sql

Copiar código

CREATE INDEX idcliente\_idx ON barberiasdb.Agendamiento (idcliente);

# Resultados y Validación

Inserción de Datos

Se realizaron pruebas de inserción de datos para validar la estructura y restricciones. Ejemplo de inserción en la tabla Clientes:

sql

Copiar código

INSERT INTO barberiasdb.Clientes (nombre, correo, telefono)

VALUES ('Juan Pérez', 'juan.perez@gmail.com', '987654321');

**Consultas de Verificación**

Se ejecutaron consultas que involucraron múltiples tablas para comprobar la correcta integración de los datos. Ejemplo:

**Consulta:** Mostrar las citas agendadas con detalles:

sql

Copiar código

SELECT c.nombre AS Cliente, b.nombre AS Barbero, s.nombre AS Servicio, a.fecha\_cita

FROM barberiasdb.Agendamiento a

JOIN barberiasdb.Clientes c ON a.idcliente = c.idClientes

JOIN barberiasdb.Barberos b ON a.idbarbero = b.idBarberos

JOIN barberiasdb.Servicios s ON a.idservicio = s.idServicio;

**Validación de Integridad**

Se probaron restricciones de integridad referencial al intentar insertar registros inválidos, lo que confirmó que el modelo previene inconsistencias.

**Conclusión**

El proceso de modelado de la base de datos cumplió con los objetivos establecidos. Las entidades y relaciones fueron correctamente definidas, permitiendo la implementación de un sistema que es robusto y escalable.

**Lecciones Aprendidas**

1. La planificación detallada en la etapa de modelado es esencial para evitar errores durante la implementación.
2. Nombrar de manera única las restricciones y claves evita conflictos durante la creación de tablas.

Este modelo de base de datos constituye una base sólida para el desarrollo del sistema de gestión de barberías, asegurando una gestión eficiente de los datos y facilitando futuras ampliaciones.

# Diagrama ER

Diagrama

Descripción generada automáticamente

# Código SQL base de datos

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema barberiasdb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `barberiasdb` DEFAULT CHARACTER SET utf8mb4 ;

USE `barberiasdb`;

-- -----------------------------------------------------

-- Table `Clientes`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Clientes`;

CREATE TABLE IF NOT EXISTS `Clientes` (

`idClientes` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(100) NOT NULL,

`correo` VARCHAR(100) NOT NULL,

`telefono` VARCHAR(15) NULL,

`direccion` VARCHAR(255) NULL,

PRIMARY KEY (`idClientes`),

UNIQUE INDEX `correo\_UNIQUE` (`correo` ASC)

)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Barberias`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Barberias`;

CREATE TABLE IF NOT EXISTS `Barberias` (

`idBarberia` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(255) NULL,

`telefono` VARCHAR(15) NULL,

PRIMARY KEY (`idBarberia`)

)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Barberos`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Barberos`;

CREATE TABLE IF NOT EXISTS `Barberos` (

`idBarberos` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(100) NOT NULL,

`especialidad` VARCHAR(50) NULL,

`Barberias\_idBarberia` INT NOT NULL,

PRIMARY KEY (`idBarberos`, `Barberias\_idBarberia`),

INDEX `fk\_Barberos\_Barberias1\_idx` (`Barberias\_idBarberia` ASC),

CONSTRAINT `fk\_Barberos\_Barberias1`

FOREIGN KEY (`Barberias\_idBarberia`)

REFERENCES `Barberias` (`idBarberia`)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Servicios`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Servicios`;

CREATE TABLE IF NOT EXISTS `Servicios` (

`idServicio` INT NOT NULL AUTO\_INCREMENT,

`nombre\_servicio` VARCHAR(100) NOT NULL,

`precio` FLOAT NULL,

PRIMARY KEY (`idServicio`)

)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Agendamiento`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Agendamiento`;

CREATE TABLE IF NOT EXISTS `Agendamiento` (

`idAgendamiento` INT NOT NULL AUTO\_INCREMENT,

`idcliente` INT NOT NULL,

`idbarbero` INT NOT NULL,

`idservicio` INT NOT NULL,

`idbarberia` INT NOT NULL,

`fecha\_cita` DATETIME NOT NULL,

PRIMARY KEY (`idAgendamiento`, `idcliente`, `idbarbero`, `idbarberia`),

INDEX `id\_cliente\_idx` (`idcliente` ASC),

INDEX `id\_barbero\_idx` (`idbarbero` ASC),

INDEX `idservicio\_idx` (`idservicio` ASC),

INDEX `idbarberia\_idx` (`idbarberia` ASC),

CONSTRAINT `idcliente`

FOREIGN KEY (`idcliente`)

REFERENCES `Clientes` (`idClientes`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `idbarbero`

FOREIGN KEY (`idbarbero`)

REFERENCES `Barberos` (`idBarberos`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `idservicio`

FOREIGN KEY (`idservicio`)

REFERENCES `Servicios` (`idServicio`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `idbarberia`

FOREIGN KEY (`idbarberia`)

REFERENCES `Barberias` (`idBarberia`)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Notificaciones`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Notificaciones`;

CREATE TABLE IF NOT EXISTS `Notificaciones` (

`idNotificaciones` INT NOT NULL AUTO\_INCREMENT,

`idcliente` INT NOT NULL,

`mensaje` TEXT NOT NULL,

`fecha\_envio` DATETIME NOT NULL,

PRIMARY KEY (`idNotificaciones`),

INDEX `idcliente\_idx` (`idcliente` ASC),

CONSTRAINT `fk\_notificaciones\_cliente`

FOREIGN KEY (`idcliente`)

REFERENCES `Clientes` (`idClientes`)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Perfilcliente`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Perfilcliente`;

CREATE TABLE IF NOT EXISTS `Perfilcliente` (

`idPerfilcliente` INT NOT NULL AUTO\_INCREMENT,

`idcliente` INT NOT NULL,

`fecha\_nacimiento` DATE NULL,

`genero` VARCHAR(45) NULL,

`Perfilclientecol` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idPerfilcliente`),

INDEX `idcliente\_idx` (`idcliente` ASC),

CONSTRAINT `fk\_perfilcliente\_cliente`

FOREIGN KEY (`idcliente`)

R

# Reflexión

The development of this project allowed us to face real challenges that required both technical skills and communication and teamwork abilities. Throughout this process, we learned to manage time efficiently, prioritize tasks, and coordinate efforts to achieve the objectives set.

One of the most valuable aspects was our ability to adapt to changes and unforeseen events, strengthening our problem-solving skills and decision-making under pressure. We also experienced the importance of maintaining open and effective communication within the team, which enabled us to address issues collaboratively and find comprehensive solutions.

In terms of learning, this project reaffirmed our understanding of the importance of planning and structured design in developing technological systems. Additionally, it gave us a clearer perspective on how the solutions we design can positively impact end users and our clients' business processes.

In conclusion, this project not only solidified our technical knowledge but also helped us grow into more well-rounded professionals, ready to tackle the challenges of the professional world.

