


Este proyecto se centra en el desarrollo de un software para la gestión de una peluquería. A través de un diseño modular, optimizamos la programación de citas y seguimiento de clientes. En conjunto, ofrecemos una solución integral que impulsa el crecimiento del negocio.

Me cortaron mal el pelo

Proyecto fase 2

Kaled Hernández - Alejandra Zamora – Oscar Fernández – Juan Domenech



Índice

Resumen entrega anterior.....	2
Requisitos de alto nivel	3
Funcionales:	3
No funcionales:	3
Casos de Uso.....	4
Caso de uso 1: Agregando horarios disponibles	5
Caso de uso 2: Editando horarios disponibles.....	5
Caso de uso 3: Agendando reserva.....	6
Caso de uso 4: Recordando Reservas	7
Caso de uso 5: Creando clientela fiel.....	8
Caso de uso 6: Gestionando Reservas	8
Caso de Uso 7: Revisando Horarios	9
Caso de Uso 8: Administrando Horarios.....	10
Diagrama de Clases.....	12
Patrones GRASP y GoF	12
Diagrama de despliegue	14
Diagrama de componentes.....	15
Bocetos	16
Conclusión.....	20
Ilustraciones.....	21
Bibliografía	21

Resumen entrega anterior

Previo a este trabajo, se llevó a cabo un análisis detallado de los procesos operativos de una peluquería, identificando áreas clave como la gestión de citas y seguimiento de clientes. Posteriormente, se diseñó un software modular para mejorar la eficiencia en la programación de citas y proporcionando personalización de servicios. Este enfoque integral buscó optimizar las operaciones y contribuir al crecimiento del negocio.

Para lograr esto describimos y entrevistamos a nuestro cliente para lograr conocer mejor la demográfica de sus clientes, su forma de trabajo, los sistemas que usa para su negocio y como planea expandir su negocio (Para consultar la entrevista revisar la primera entrega).

Optamos por concentrarnos en la forma en que nuestro peluquero se conecta con sus clientes, trabajando en el diseño de un Bot utilizando la tecnología y servicios proporcionados por Meta para WhatsApp (Meta, 2023) utilizando el sistema Flow. Incluyendo el desarrollando diagramas de actividades y haciendo análisis Canvas sobre el impacto de este proyecto sobre el negocio de nuestro cliente.

Determinamos ciertas metas a cumplir:

- 1) Simplificar los procesos del sistema
- 2) Facilitar el manejo y la ejecución del negocio
- 3) Automatizar el proceso de reserva
- 4) Minimizar el trabajo manual del usuario
- 5) Permitir la expansión del negocio

Requisitos de alto nivel

Funcionales:

1. El Bot debe poder enviar mensajes
2. El Bot debe poder recibir y guardar respuestas
3. El Bot debe poder conectarse con una interfaz para enviar los datos guardados y generar la agenda
4. El Peluquero debe poder ingresar y gestionar sus horarios disponibles
5. El Peluquero debe poder revisar los horarios reservados y los datos asociados a la reserva
6. El Peluquero debe poder gestionar las reservas
7. El Cliente debe poder reservar un horario disponible
8. El Cliente debe poder gestionar sus reservas

No funcionales:

1. El Bot debe poder funcionar en ambiente móvil
2. Debe tener una base de seguridad con el manejo de datos en las conversaciones
3. El Bot debe actualizar los datos guardados en la base de datos para todos los usuarios en menos de 5 segundos
4. El Bot y la Interfaz deben mostrar mensajes de error que indiquen al usuario el error cometido y lo orienten para que no vuelvan a cometerlo

Casos de Uso

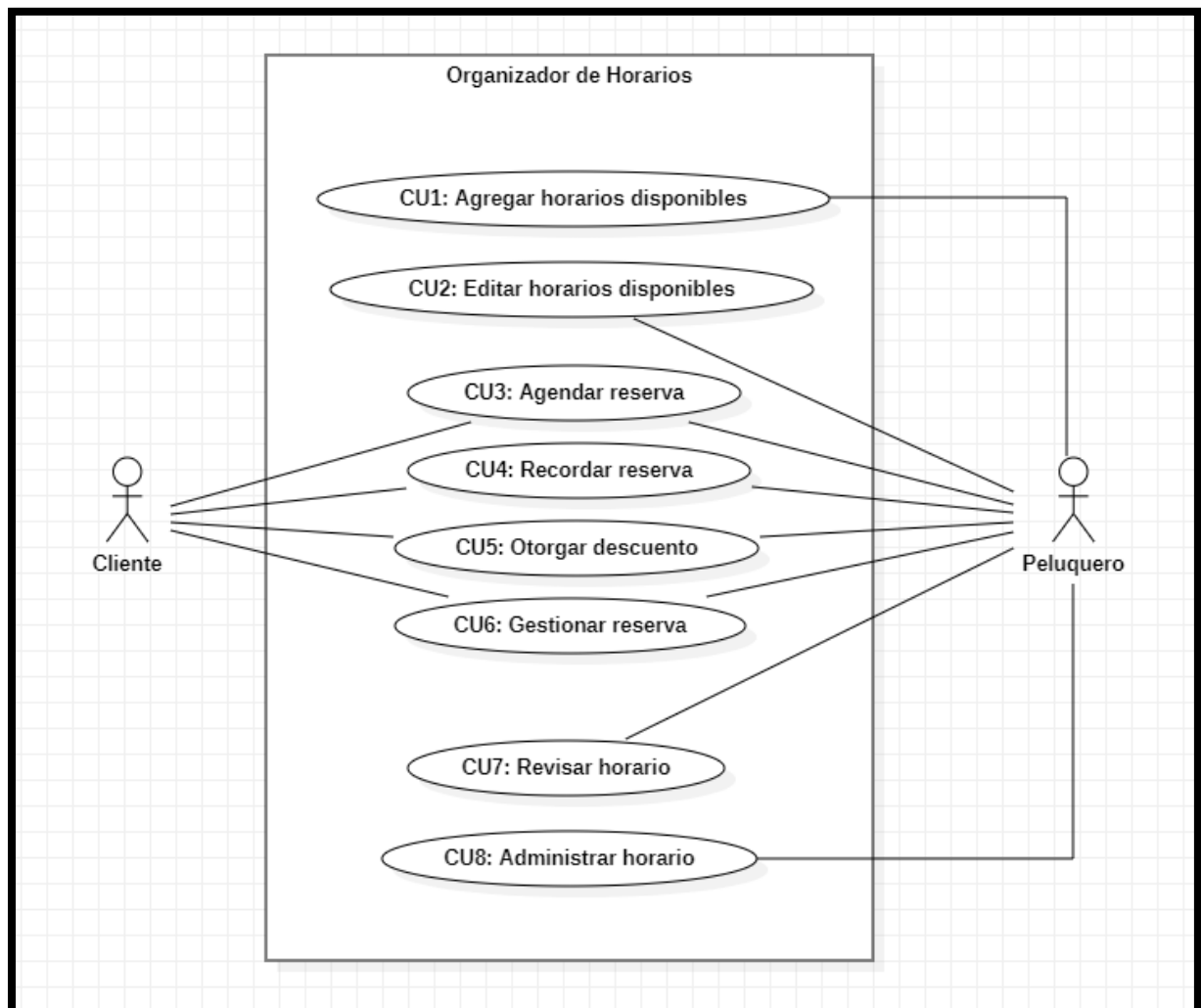


Ilustración 1: Diagrama de Casos de Uso

Caso de uso 1: Agregando horarios disponibles

ID: CU1

Nombre: Agregar horarios disponibles

Actores: Peluquero

Descripción: Permite que el peluquero ingrese sus horarios disponibles (slots).

Precondiciones:

- El peluquero tiene decididas sus horas de trabajo y la duración de cada slot.
- El peluquero ha ingresado a su cuenta de WhatsApp Business y se encuentra en el sector "Flows" en el Administrador de WhatsApp.

Postcondiciones:

- WhatsApp Manager deja guardadas las horas disponibles del peluquero

Flujo Normal:

1. El Peluquero presiona el botón "Crear Flow"
2. El Peluquero rellena los datos pedidos y selecciona "Appointment Booking" en el sector "Template"
3. El Peluquero agrega todos sus horarios disponibles
4. El Peluquero guarda los datos ingresados
5. El Peluquero presiona el botón "Publish" para publicar los horarios ingresados

Flujo Alternativo:

- 3.1. El Peluquero presiona el botón "Publish" para publicar los horarios ingresados

Caso de uso 2: Editando horarios disponibles

ID: CU2

Nombre: Editar horarios disponibles

Actores: Peluquero

Descripción: Permite que el peluquero edite sus horarios disponibles.

Precondiciones:

- El peluquero ya ha creado, al menos, un horario
- El peluquero ha ingresado a su cuenta de WhatsApp Business y se encuentra en el sector "Flows" en el Administrador de WhatsApp.

Postcondiciones:

- El Administrador de WhatsApp deja guardadas las ediciones de las horas disponibles del peluquero.

Flujo Normal:

1. El Peluquero selecciona el “Flow” que desea editar del listado presente
2. El Peluquero edita el/los horario/s que desea editar
3. El Peluquero guarda los cambios
4. El Peluquero presiona el botón “Publish” para publicar los horarios editados

Flujo Alternativo:

- 2.1. En el caso de que ya esté reservado el horario editado, el Sistema envía un mensaje al cliente afectado con los detalles de la modificación

Caso de uso 3: Agendando reserva

ID: CU3

Nombre: Agendar reserva

Actores: Cliente, Peluquero

Descripción: Permite que los clientes del peluquero puedan reservar una hora con él, asegurando su atención.

Precondiciones:

- El Cliente tiene el número del peluquero en WhatsApp

Postcondiciones:

- El horario elegido por el Cliente queda reservado
- El Peluquero es notificado de la reserva de horario

Flujo normal:

1. El Cliente envía un mensaje a través de WhatsApp al número del Peluquero
2. El Sistema envía un mensaje de saludo y cuestionario con la opción de “Reservar hora”, entre otras.
3. El Cliente selecciona “Reservar hora”
4. El Sistema muestra un formulario pidiendo los datos del cliente y los horarios disponibles
5. El Cliente ingresa sus datos y selecciona un horario del listado presentado
6. El Sistema envía un cuestionario pidiendo confirmación de los datos de la reserva (si/no/editar datos)
7. El Cliente confirma sus datos

8. El Sistema envía un mensaje agradeciendo la confirmación
9. El Sistema guarda los datos El Sistema envía una notificación de la reserva al Peluquero

Flujo alternativo:

- 3.1. El Cliente selecciona otra opción
- 4.1.1. Si el Cliente no responde dentro de 2 horas, el Sistema cierra la conversación
- 4.1.2. El Sistema notifica al Peluquero del intento de comunicación
- 7.1. El Cliente rechaza la reserva
- 7.2. Si el Cliente no responde dentro de 2 horas, se rechaza la reserva
- 7.3.1. El Cliente selecciona editar datos
- 7.3.2. El Sistema muestra un formulario pidiendo los datos
- 7.3.3. El Cliente ingresa los datos a modificar
- 7.3.4. El Sistema guarda los datos ingresados
- 7.3.5. El flujo vuelve al paso 6 del flujo normal

Caso de uso 4: Recordando Reservas

ID: CU4

Nombre: Recordar reserva

Actores: Cliente, Peluquero

Descripción: Recuerda al Cliente y Peluquero acerca de la reserva para asegurar asistencia.

Precondiciones:

- El Cliente debe tener al menos una reserva confirmada

Postcondiciones:

- El Peluquero es notificado de la reserva nuevamente

Flujo normal:

1. El Sistema envía un mensaje y cuestionario al Cliente pidiendo confirmar la reserva 24 horas antes de ésta.
2. El Cliente vuelve a confirmar su reserva.
3. El Sistema envía un mensaje al Cliente agradeciendo su respuesta.
4. El Sistema envía una notificación de la confirmación de hora al Peluquero.

Flujo alternativo:

- 1.1. Si la reserva fue hecha con menos de 24 horas de anticipación, el Sistema no envía el mensaje o cuestionario.
- 2.1.1. El Cliente cancela su reserva.

- 2.1.2. El Sistema elimina la reserva del Cliente.
- 2.1.3. El Sistema notifica al Peluquero de la cancelación.
- 2.2. Si el Cliente no responde dentro de 2 horas, el Sistema notifica al Peluquero para que él pueda tratar de confirmar por su cuenta si lo encuentra necesario.

Caso de uso 5: Creando clientela fiel

ID: CU5

Nombre: Otorgar Descuentos

Actores: Peluquero, Cliente

Descripción: Da una recompensa al Cliente tras su reserva para incentivarlo a volver a reservar una hora con el Peluquero.

Precondiciones:

- El Cliente debe haber sido atendido por el Peluquero al menos una vez.

Postcondiciones:

- El Cliente recibe un código/imagen con un porcentaje de descuento (definido por el Peluquero) para el siguiente servicio solicitado al Peluquero.
- El Peluquero es notificado de la generación del voucher de descuento para evitar abusos.

Flujo Normal:

1. El Sistema envía un mensaje y formulario de reseña del servicio al Cliente 30 minutos - 1 hora después del término de la reserva.
2. El Cliente completa el formulario
3. El Sistema envía un mensaje y cupón de descuento al Cliente agradeciendo su respuesta.
4. El Sistema guarda la reseña para que ésta pueda ser utilizada por el Peluquero si así lo desea.
5. El Sistema envía una notificación del descuento al Peluquero.

Flujo Alternativo:

- 2.1. El Cliente no completa el formulario
- 2.2. El Sistema cierra la posibilidad de obtener el descuento pasado 3 días

Caso de uso 6: Gestionando Reservas

ID: CU6

Nombre: Gestionar Reserva

Actores: Peluquero, Cliente

Descripción: Permite que el Cliente pueda cambiar la fecha/hora o cancelar su reserva si lo desea, dejando el horario original libre para otro posible Cliente.

Precondiciones:

- El Cliente debe tener al menos una reserva confirmada.

Postcondiciones:

- El horario original queda disponible.

Flujo Normal:

1. El Cliente envía un mensaje al WhatsApp del Peluquero
2. El Sistema responde con un mensaje y cuestionario con la opción de “Gestionar reserva existente”, entre otras.
3. El Cliente selecciona “Gestionar reserva existente”
4. El Sistema muestra las opciones de “Cambiar fecha/hora” o “Cancelar su reserva”
5. El Cliente selecciona “Cambiar fecha/hora”
6. El Sistema envía un formulario con los horarios disponibles y opción de cancelar
7. El Cliente selecciona un nuevo horario de la lista presentada
8. El Sistema guarda la reserva del nuevo horario
9. El Sistema envía un cuestionario de confirmación de la modificación
10. El Cliente selecciona “Sí”
11. El Sistema elimina la reserva del horario original
12. El Sistema envía un mensaje al Cliente confirmando los cambios
13. El Sistema envía una notificación de las modificaciones al Peluquero

Flujo Alternativo:

- 3.1. El Cliente selecciona otra opción
 - 5.1.1. El Cliente selecciona “Cancelar su reserva”
 - 5.1.2. El Sistema muestra la(s) reserva(s) existente(s) del Cliente y pide que seleccione cuál cancelará
 - 5.1.3. El Cliente selecciona la reserva que desea cancelar
 - 5.1.4. Continúa con el paso 9 del flujo normal
- 7.1.1. El Cliente selecciona cancelar
- 7.1.2. Vuelve a el paso 4 del flujo normal
- 10.1.1. El Cliente selecciona otra opción
- 10.1.2. El Sistema elimina la reserva del nuevo horario (si existe)
- 10.1.3. Vuelve al paso 4 del flujo normal

Caso de Uso 7: Revisando Horarios

ID: CU7

Nombre: Revisar Horario

Actores: Peluquero

Descripción: Muestra los horarios reservados del día, semana o mes, según se indique.

Precondiciones:

- El Peluquero está ingresado con sus credenciales en el sistema

Postcondiciones:

- El Peluquero sabe cuál es su horario y los cambios que ha tenido

Flujo normal:

1. El Peluquero selecciona <Calendario Diario>, <Calendario Semanal> o <Calendario Mensual>
2. El Sistema muestra en forma de calendario todas horas reservadas en el formato pedido, las cuales están codificadas con colores (verde para las reservas confirmadas, gris para las canceladas y transparente/default para las que no han sido confirmadas aún).

Flujo alternativo:

- 1.1. El Peluquero selecciona otra opción.

Caso de Uso 8: Administrando Horarios

ID: CU8

Nombre: Administrar Horario

Actores: Peluquero

Descripción: Permite que el Peluquero pueda revisar y administrar una reserva en específico

Precondiciones:

- Debe haber al menos una hora reservada
- El Peluquero ha generado un calendario

Postcondiciones:

- El Sistema se actualiza y muestra los cambios hechos en el calendario

Flujo normal:

1. El Peluquero selecciona una reserva de su calendario.
2. El Sistema muestra los detalles de la reserva y del Cliente, dando la opción de contactar al Cliente, cancelar la reserva (si está en espera de o ya ha sido

confirmada) o eliminar la reserva (si ha sido cancelada) y volver al calendario semanal.

3. El Peluquero selecciona <volver>

Flujo alternativo:

- 3.1.1. El Peluquero selecciona <contactar al Cliente>

- 3.1.2. El Sistema redirige al chat de WhatsApp del Cliente en cuestión

- 3.2.1. El Peluquero selecciona <cancelar reserva>

- 3.2.2. El Sistema envía un mensaje al Cliente informando que el Peluquero ha cancelado su reserva.

- 3.2.3. El Sistema elimina la reserva del calendario

- 3.3.1. El Peluquero selecciona <eliminar reserva>

- 3.3.2. El Sistema elimina la reserva del calendario

Diagrama de Clases

Al inicio del proyecto creamos un diagrama de clases que resultó incompleto e ineficiente para lo que necesitamos, así que trabajamos en incorporar una mayor claridad en la organización de las clases, atributos y métodos, facilitando la comprensión del diseño. Además, se han añadido relaciones más explícitas entre las clases, resaltando las asociaciones y dependencias clave para una visualización más precisa de la arquitectura del sistema, dando como resultado el siguiente diagrama:

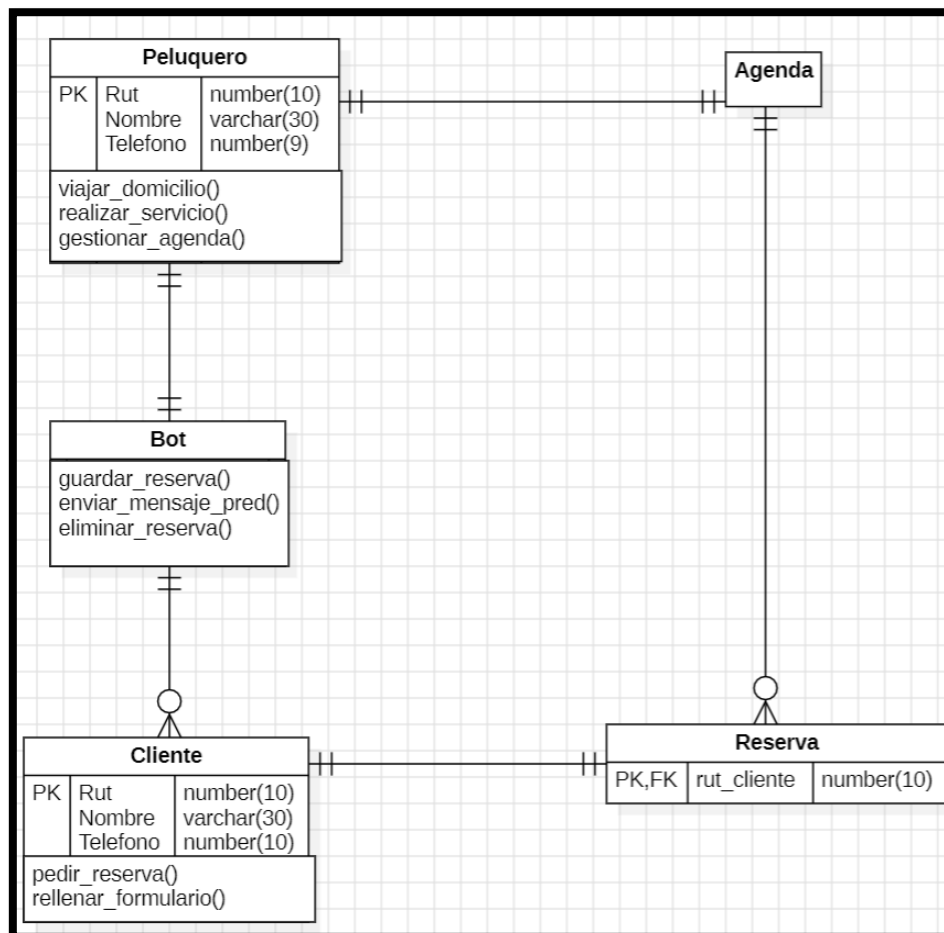


Ilustración 2: Diagrama de clases normalizado

Patrones GRASP y GoF

Es en este diagrama que aplicaremos y reconoceremos los patrones GRASP:

- **Controlador**: nos dimos cuenta que no habíamos implementado un método para generar un calendario, por lo que creamos una interfaz (`VistaCalendario`) que nos sirve como modo de comunicación con los usuarios, con el método `clickGenerarCalendario()` que se comunica con nuestro nuevo controlador

(CalendarioController). Es ésta clase la que se encargará de reunir la información necesaria desde Reserva para poder implementar su método generarCalendario(), el cual crea una nueva instancia de Agenda para que pueda ser vista por el Peluquero.

- **Bajo Acoplamiento:** con los cambios hechos aplicando el patrón anterior vimos que le dejaba mucha responsabilidad a Reserva, por lo que creamos una nueva clase DatosReserva para que sea ésta la encargada de otorgarle los datos al Controlador, disminuyendo así el acoplamiento.

Y los patrones GoF:

- **Mediador:** el Bot funciona como mediador entre el Peluquero y el Cliente, logrando así minimizar el contacto directo entre ellos.
- **Observador:** considerando que nuestra propuesta considera el uso de WhatsApp, podemos poner al Cliente como observador, ya que él puede decidir si quiere o no recibir notificaciones por parte del Bot/Peluquero.
- **Fachada:** utilizamos el patrón de fachada al implementar la interfaz VistaCalendario, ya que ésta muestra una versión simplificada del modelo creado para este sistema.
- **Builder:** con el controlador creado, tenemos un constructor que toma todos los datos de las reservas y genera una agenda/calendario.

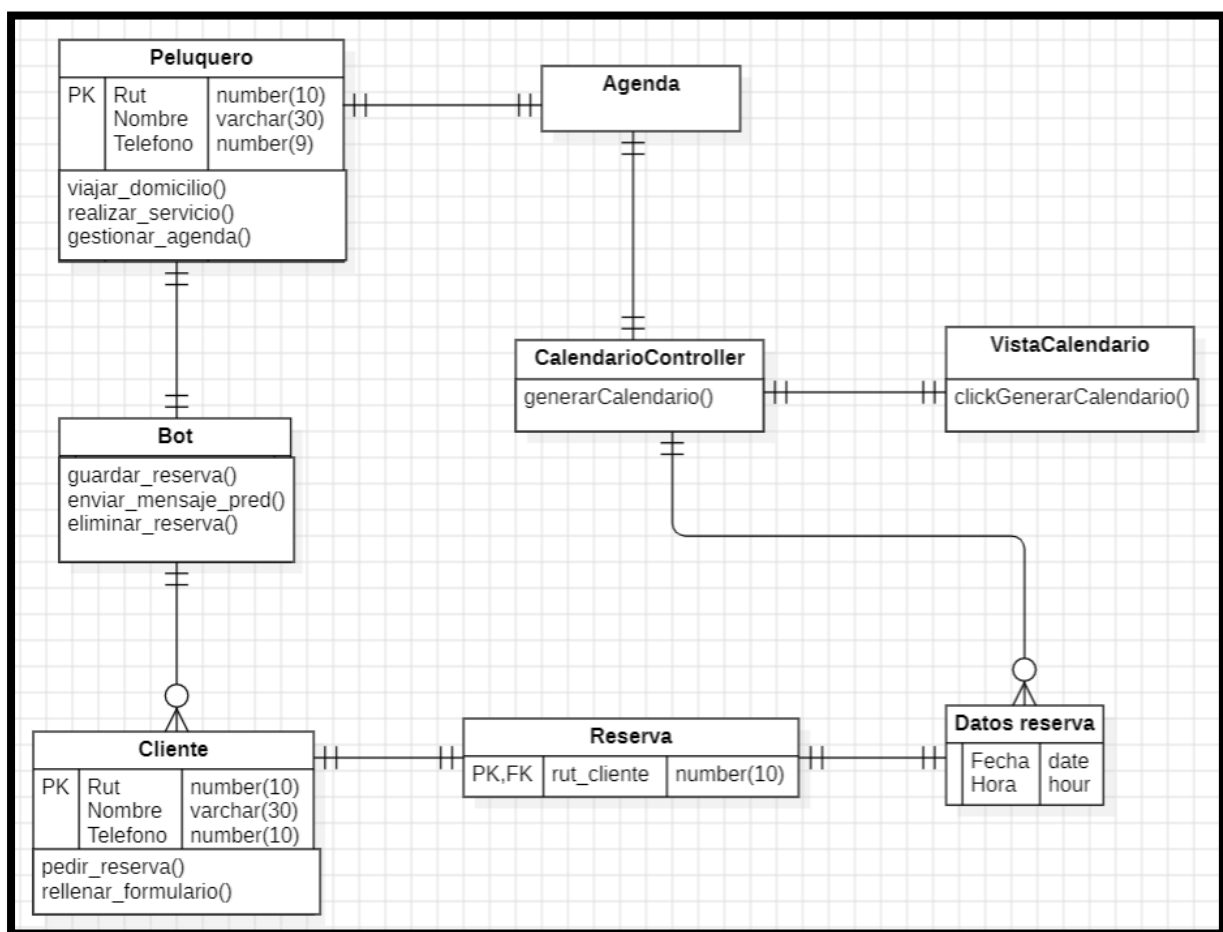


Ilustración 3: Diagrama de clases final

Diagrama de despliegue

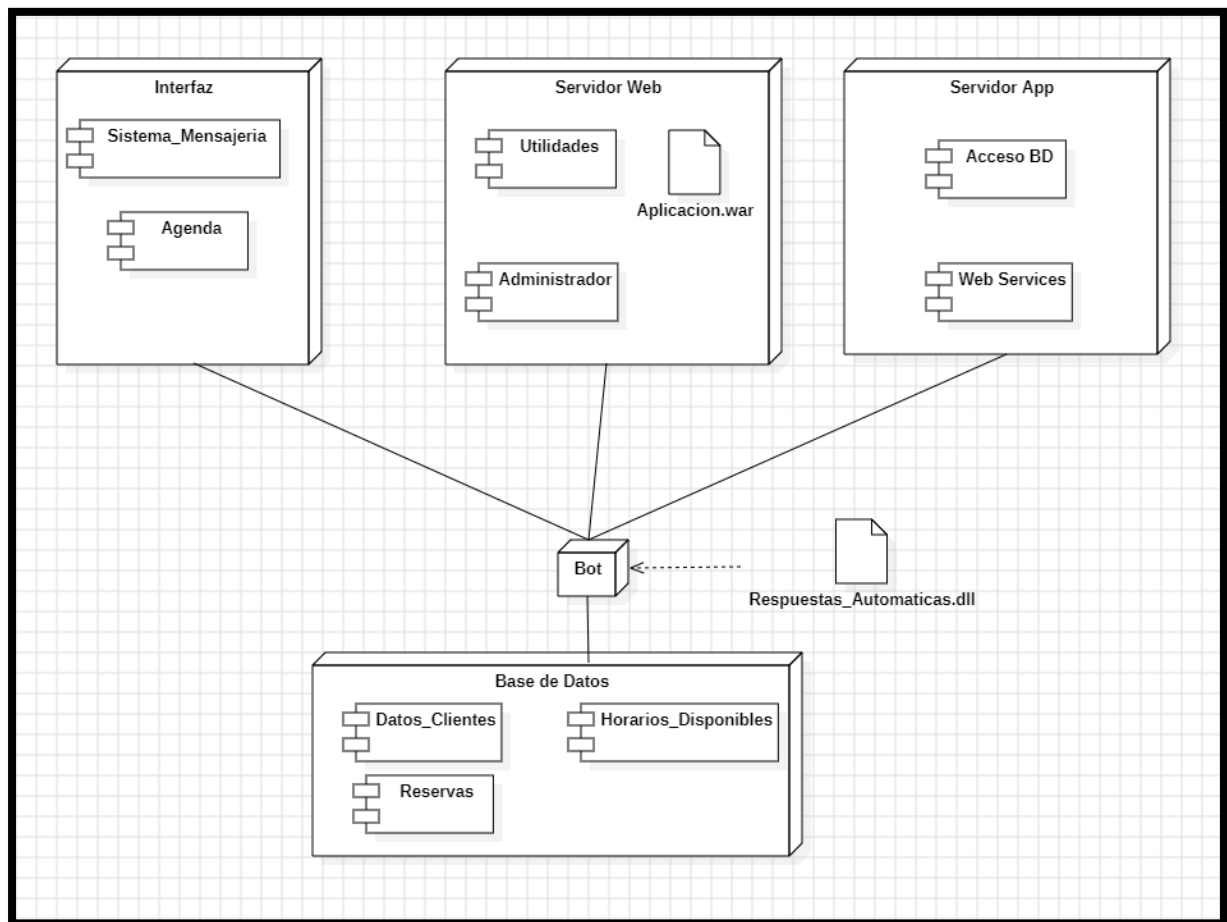


Ilustración 4: Diagrama de Despliegue

Diagrama de componentes

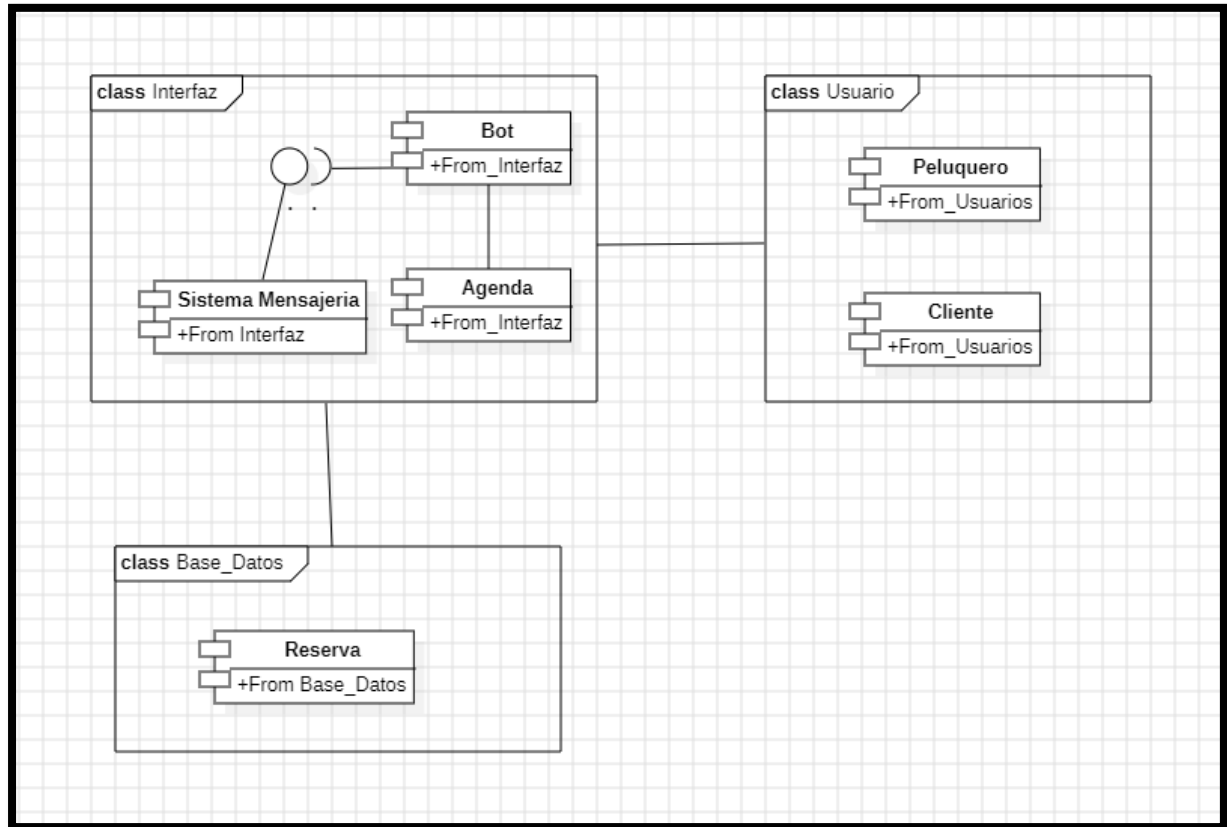


Ilustración 5: Diagrama de Componentes

Bocetos

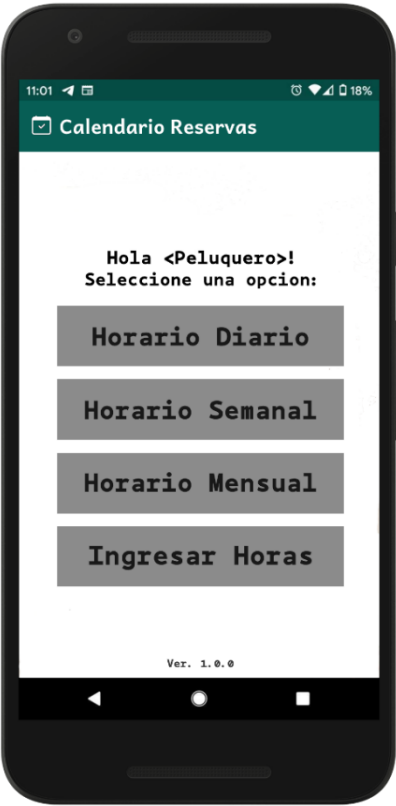


Ilustración 7:Boceto 1

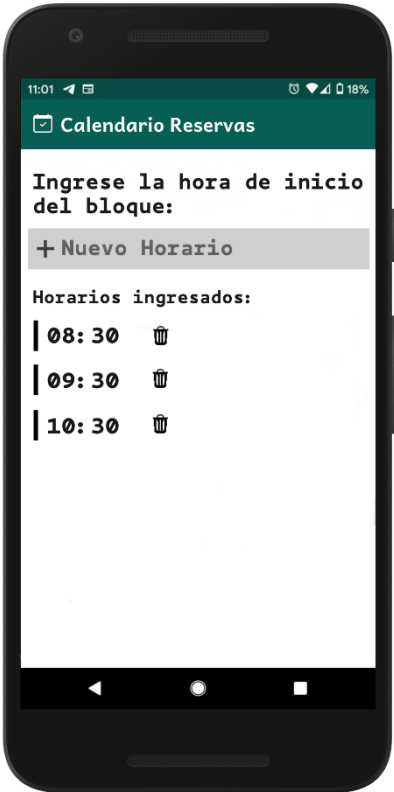


Ilustración 6: Boceto 2



Ilustración 8:Boceto 3



Ilustración 9:Boceto 4



Ilustración 11:Boceto 5

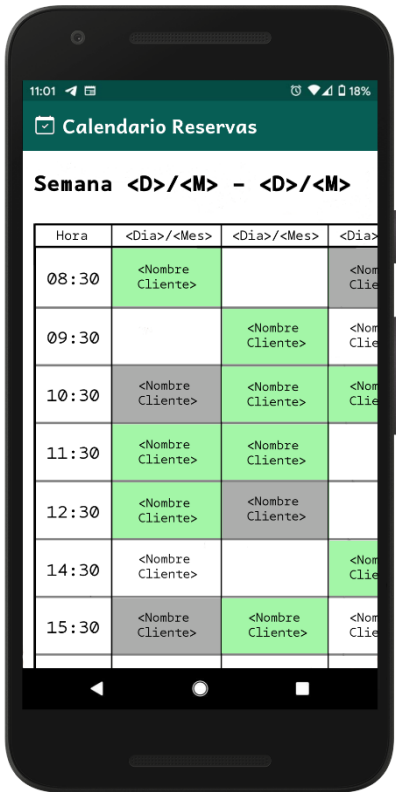


Ilustración 10:Boceto 6

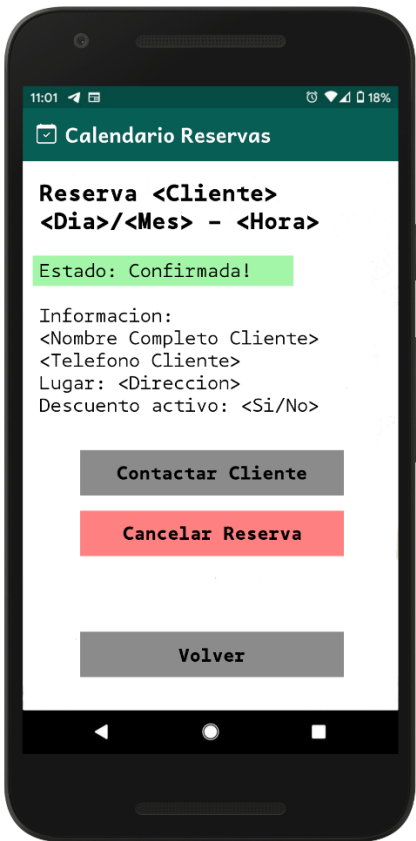


Ilustración 12:Boceto 7

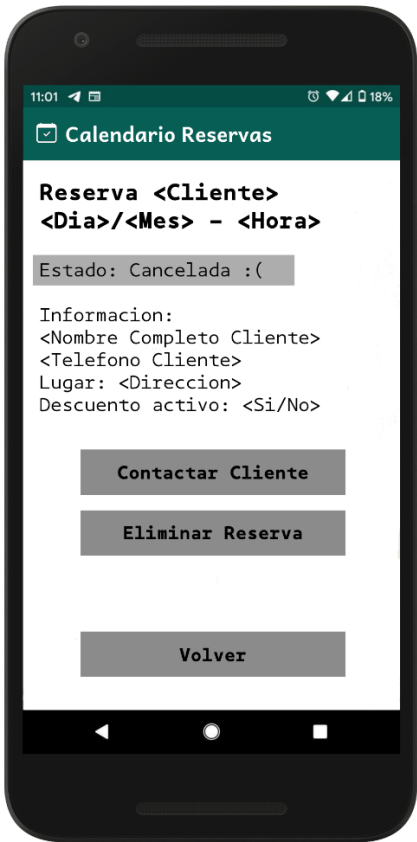


Ilustración 13: Boceto 8

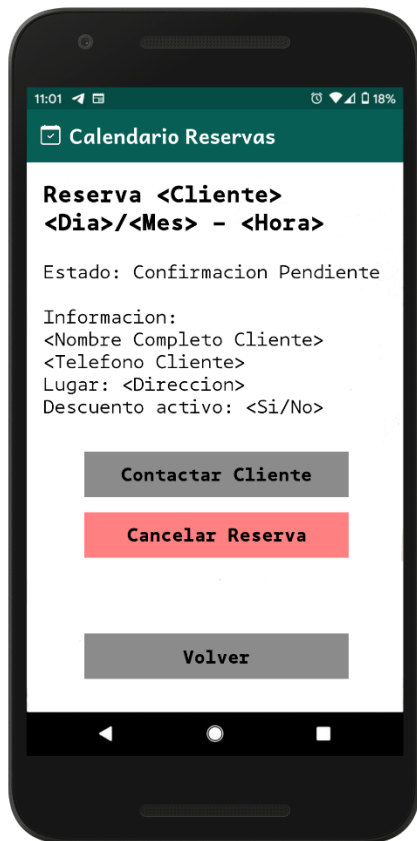


Ilustración 15: Boceto 9



Ilustración 14: Boceto 10



Ilustración 16: Boceto 8



Ilustración 17: Boceto 9



Ilustración 19: Boceto 10

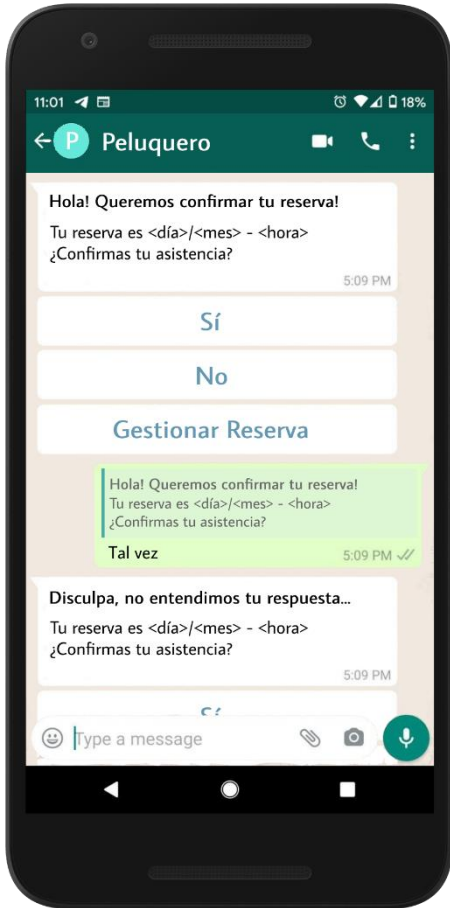


Ilustración 18: Boceto 11

Conclusión

El proceso de análisis y diseño de software para la peluquería ha sido integral y orientado a la mejora significativa de las operaciones del negocio. A través de la comprensión profunda de los procesos existentes, se logró desarrollar una solución que optimiza la gestión de citas y el seguimiento de clientes, incluyendo un sistema de fidelización de estos. En última instancia, este trabajo no solo busca automatizar tareas, sino también potenciar la experiencia del cliente y contribuir al crecimiento sostenible de la peluquería en un entorno competitivo.

A lo largo del trabajo hemos logrado mejorar nuestro uso de patrones GRASP y GoF para lograr resultados estandarizados y de mayor calidad, mejorando la legibilidad de nuestros diagramas presentados permitiendo que sea más sencilla la creación del software a partir del prototipo, todo esto con el objetivo de optimizar el flujo de datos y de procesos (Todo esto es posible visualizarlo gracias a los casos de uso).

Analizar los requisitos del proyecto y separarlos según su tipo (Funcionales o no funcionales) también ha permitido obtener resultados más coherentes y fáciles de implementar a la hora de poner en práctica el diseño del software.

Las dificultades de este proyecto estuvieron principalmente en lograr observar el trabajo desde la perspectiva del peluquero junto en su totalidad, ya que en la primera entrega no se incluyeron bocetos que mostraran la experiencia del peluquero con el resultado final, la organización de tiempo también fue una parte importante en las dificultades.

Esto se manejó oportunamente separando el trabajo en partes, dando principal atención a la perspectiva del peluquero, además de reuniones de grupo para distribuir el tiempo de manera ordenada.

Mirando hacia el futuro, se vislumbran diversas mejoras potenciales para el software de gestión de la peluquería. Entre ellas se incluyen la implementación de tecnologías emergentes, como la inteligencia artificial para recomendaciones personalizadas de cortes de pelo, así como la expansión de la plataforma para abordar nuevas necesidades comerciales.

Ilustraciones

<i>Ilustración 1: Diagrama de Casos de Uso</i>	4
<i>Ilustración 2: Diagrama de clases normalizado</i>	12
<i>Ilustración 3: Diagrama de clases final</i>	13
<i>Ilustración 4: Diagrama de Despliegue</i>	14
<i>Ilustración 5: Diagrama de Componentes</i>	15
<i>Ilustración 6: Boceto 2</i>	16
<i>Ilustración 7: Boceto 1</i>	16
<i>Ilustración 8: Boceto 3</i>	16
<i>Ilustración 9: Boceto 4</i>	16
<i>Ilustración 10: Boceto 6</i>	17
<i>Ilustración 11: Boceto 5</i>	17
<i>Ilustración 12: Boceto 7</i>	17
<i>Ilustración 13: Boceto 8</i>	17
<i>Ilustración 14: Boceto 10</i>	18
<i>Ilustración 15: Boceto 7</i>	18
<i>Ilustración 16: Boceto 8</i>	18
<i>Ilustración 17: Boceto 9</i>	18
<i>Ilustración 18: Boceto 11</i>	19
<i>Ilustración 19: Boceto 10</i>	19

Bibliografía

Meta. (m.f de m.f de 2023). *Meta for Developers*. Obtenido de <https://developers.facebook.com/docs/whatsapp/flows/introduction/whatarewaflows>