



Universidad Tecnológica Ecotec

Facultad de Ingenierías, Arquitectura y Ciencias de la Naturaleza

Título del Proyecto:

Desarrollo de un Asistente Virtual para la asignación y gestión de terapias personalizadas
en el consultorio estético Stella Spa

Línea de investigación:

Tecnologías de la información y la comunicación

Carrera:

Ingeniería en Software

Autores:

Marcell Andre Moreno Salcedo

Tommy Elian Ramirez Ochoa

Agradecimiento**Marcell Andre Moreno Salcedo**

Deseo expresar mi más profundo agradecimiento a mis padres, por su apoyo incondicional y por haberme guiado durante todo el proceso. A mi hermano, por ser una fuente de motivación. Finalmente, a la cosmetóloga Estela Sarango por abrirnos las puertas de su negocio, Stella Spa. Su colaboración fue esencial para la aplicación práctica de este proyecto y para validar los resultados de mi investigación.

Tommy Elian Ramirez Ochoa

Quiero expresar mi más sincero agradecimiento a mi madre por su amor y apoyo incondicional, a mi hermano por su presencia y palabras de aliento, y a mi enamorada por su paciencia y confianza en mí; cada uno de ellos ha sido fundamental en mi camino y me ha brindado la fortaleza y motivación necesarias para seguir adelante.



ANEXO No. 9

**PROCESO DE TITULACIÓN
CERTIFICADO DE APROBACIÓN DEL TUTOR**

Samboyondón, 04 de Agosto de 2025

Magíster

Erika Ascencio Jordán

Facultad de Ingenierías, Arquitectura y Ciencias de la Naturaleza

Universidad Tecnológica ECOTEC

De mis consideraciones:

Por medio de la presente comunico a usted que el trabajo de titulación TITULADO: **Desarrollo de un asistente virtual para la asignación y gestión de terapias personalizadas en el consultorio estético Stella Spa**, fue revisado, siendo su contenido original en su totalidad, así como el cumplimiento de los requerimientos establecidos en la guía para su elaboración, por lo que se autoriza a los estudiantes: **MORENO SALCEDO MARCELL ANDRE / RAMIREZ OCHOA TOMMY ELIAN**, para que procedan con la presentación oral del mismo.

ATENTAMENTE,



Firmado electrónicamente por:
MARCOS ANTONIO
ESPINOZA MINA

Marcos Antonio Espinoza Mina, Ph.D.
Tutor



ANEXO No. 10

PROCESO DE TITULACIÓN CERTIFICADO DEL PORCENTAJE DE COINCIDENCIAS DEL TRABAJO DE TITULACIÓN

Habiendo sido revisado el trabajo de titulación TITULADO: **Desarrollo de un asistente virtual para la asignación y gestión de terapias personalizadas en el consultorio estético Stella Spa**, elaborado por MORENO SALCEDO MARCELL ANDRE / RAMIREZ OCHOA TOMMY ELIAN fue remitido al sistema de coincidencias en todo su contenido el mismo que presentó un porcentaje del 4% mismo que cumple con el valor aceptado para su presentación que es inferior o igual al 10% sobre el total de hojas del documento. Adicional se adjunta print de pantalla de dicho resultado.

Informe de originalidad de SafeAssign • Enviado el jue., 31 jul. 2025 20:24

Documento Final del Proyecto Integrador Curricular (WORD)

PROYECTO INTEGRADOR

Puntuación total: 4 % Riesgo bajo

MARCELL ANDRE MORENO SALCEDO

UUID de entrega: 58484017-0a6b-480c-94c4-0c16287b7523

Número total de informes	Coincidencia máxima	Coincidencia promedio	Enviado el	Conteo de palabras promedio
1	4 % Proyecto Integrador Final Moreno - R...	4 %	31/07/25 20:24 GMT-5	16.744 Más alto: Proyecto Integrador Final M...

Contenido adicional 4 %

Conteo de palabras: 16,744
Origen: Proyecto Integrador Final Moreno - Ramirez.docx

ATENTAMENTE,



MARCOS ANTONIO ESPINOZA MINA, Ph.D.
Tutor

Índice

Índice de Tablas.....	viii
Índice de Figuras.....	ix
Índice de Anexos.....	xi
Resumen.....	1
1. Introducción.....	2
1.1 Contexto histórico social del objeto de estudio.....	3
1.2 Antecedentes.....	4
1.3 Planteamiento del problema.....	5
1.4 Objetivos.....	7
Objetivo General.....	7
Objetivos Específicos.....	8
1.5 Justificación.....	8
2. Marco Teórico.....	10
2.1 Asistentes Virtuales.....	10
2.1.1. Aplicación en diversos sectores.....	11
2.2 Implementación en la atención al cliente.....	11
2.2.1 Principales beneficios de los asistentes virtuales (ChatBots) Atención a la cliente ininterrumpida.....	12
2.2.2 Optimización de recursos y reducción de costos.....	12
2.3 Ley de protección de datos.....	13
2.4 Desarrollo del ChatBot.....	14
2.5 Beneficios de emplearlo en el desarrollo.....	14
2.6 Baileys whaspsapp.....	15
2.7 Conceptos clave dentro de la herramienta Flow (Flujos de conversación).....	15
2.8 Base de datos (DataBase).....	15
2.9 MongoDB.....	16
2.10 API'S.....	16
2.10.1 Funcionamiento.....	17
2.10.2 Beneficios de emplear apis's Aumento de la eficiencia.....	17
2.10.3 Mayor flexibilidad.....	18
2.10.4 Reducción de costos.....	18
2.11 Lenguajes de programación web.....	18

2.11.1 HTML y CSS.....	18
2.11.2 Ventajas de usar CSS.....	19
2.11.3 Consistencia en el diseño.....	19
2.11.4 Optimización del rendimiento.....	19
2.11.5 JavaScript.....	20
2.11.6 Desarrollo Back-End.....	20
2.11.7 Node.JS.....	21
2.11.8 Transmisión de datos.....	21
2.11.9 Compatibilidad entre frontend y backend.....	21
2.11.10 AngularJS.....	22
3. Metodología.....	23
3.1 Metodología SCRUM.....	23
Destacando sus beneficios:.....	23
Product Owner.....	24
Roles Claves.....	25
1. Inicio del proyecto – Definición del Product Owner y visión del producto.....	25
2. Recolección de requisitos y análisis del problema – Product Backlog inicial....	25
3. Diseño de la solución – Sprint Planning.....	25
4. Desarrollo e integración – Sprints de trabajo.....	25
3.2 Plan de Trabajo – Cronograma de Actividades.....	27
Tareas:.....	27
Entregables:.....	28
Fase 2: Diseño de la Solución (Semanas 3-4).....	28
Tareas:.....	28
Entregables:.....	29
Fase 3: Desarrollo e Integración (Semanas 5-10) Objetivos relacionados:.....	31
Tareas:.....	32
Entregables:.....	33
Fase 4: Pruebas Piloto y Validación (Semanas 11-13).....	34
Tareas:.....	34
Entregables:.....	34
Fase 5: Integración Final y Capacitación (Semanas 14-15) Tareas:.....	34
Entregables:.....	35

Fase 6: Evaluación y Cierre del Proyecto (Semana 16) Tareas:	35
Entregables:	35
Actividades de Jira:	35
TAREA: Revisar Documentación y curso de la herramienta:	36
TAREA: Conexión a base MongoDB:	36
TAREA: Primeras consultas y respuestas:	38
TAREA: Almacenamiento de los datos en la base:	39
TAREA: Nuevas funciones para la descripción de los servicios:	40
TAREA: MongoDB:	41
TAREA: Funciones para el ambiente web:	41
4. Análisis e interpretación de resultados:	44
5. Conclusiones:	52
6. Recomendaciones:	53
7. Referencias bibliográficas:	54
8. Anexos:	59

Índice de Tablas

Tabla 1. Descripción de la fase de la metodología Scrum	26
Tabla 2. Resumen de resultados	50

Índice de Figuras

Figura 1. Estructura del modelo	30
Figura 2. Pantalla web de inicio	33
Figura 3. Menú de opciones	33
Figura 4. Interacción con el ChatBot	34
Figura 5. Jira Tareas Asignadas	35
Figura 6. Dependencias del proyecto	36
Figura 7. Librerías del front	37
Figura 8. Librerías del back	37
Figura 9. Librerías del Bot	38
Figura 10. Flujos de eventos	38
Figura 11. Interfaz de respuesta	39
Figura 12. Colecciones de la base de datos	39
Figura 13. Flujo de servicios 1	40
Figura 14. Flujo de servicios 2	40
Figura 15. Agenda de consultas	41
Figura 16. Agendar cita	41
Figura 17. Perfiles	42
Figura 18. Información del paciente	42
Figura 19. Alimentación de información para el bot.	43
Figura 20. Pregunta 1. ¿Con qué frecuencia visitas Stella Spa?	97
Figura 21. Pregunta 2. ¿Cómo agendas actualmente tus citas en Stella Spa?	97
Figura 22. Pregunta 3. ¿Has tenido dificultades al momento de agendar tu cita?	98
Figura 23. Pregunta 4. ¿Te gustaría poder agendar citas mediante un asistente virtual disponible 24/7?	98
Figura 24. Pregunta 5. ¿Consideras importante recibir recordatorios automáticos de tus citas?	98
Figura 25. Pregunta 6. ¿Qué tan importante es para ti recibir recomendaciones de	

tratamientos personalizadas basadas en tu historial? 99

Figura 26. Pregunta 7. ¿Crees que el servicio actual en Stella Spa está suficientemente personalizado a tus necesidades? 99

Figura 27. Pregunta 8. ¿Qué canal preferirías para interactuar con el asistente virtual? 100

Figura 28. Pregunta 9. ¿Estarías dispuesto/a a proporcionar tus datos personales si se garantiza su protección bajo la Ley de Protección de Datos? 100

Figura 29. Pregunta 10. ¿Recomendarías a algún familiar o amigo Stella Spa? 101

Índice de Anexos

Anexo A. Página Oficial del Instagram del SPA y su funcionamiento	59
Anexo B. Captura de pantalla de chat Whatsapp con la Dra. Estela Sarango	59
Anexo C. Fotografía de tratamiento	60
Anexo D. Fotografía de tratamiento de limpieza facial	60
Anexo E. Fotografía de tratamiento Desintoxicación IONICA	61
Anexo F. Reuniones	61
Anexo G. Captura de reunión con la terapeuta para brindar las soluciones con respecto al sistema de agendamiento de citas	63
Anexo H. Entrevista para conocer la experiencia actual del cliente con respecto al servicio de Stella Spa	63
Anexo I. Uso de tratamientos de datos, se solicitaron los permisos para el tratamiento de los datos	65
Anexo J. Estructura para el desarrollo del proyecto	66
Anexo K. Tabla comparativa de empresas que emplean IA	67
Anexo L. Cronograma de actividades	68
Anexo M. Reunión sobre avance del proyecto	69
Anexo N. Gestión de autorización de la empresa	70
Anexo O. Captura de pantalla de página web	70
Anexo P. Integración de los servicios de Stella Spa	71
Anexo Q. Código de página web HTML	71
Anexo R. Código de proyecto	72
Anexo S. Código del CSS	72
Anexo T. Código de configuración del ChatBot	73
Anexo U. Flujos de entrada del Chatbot	73
Anexo V. Flujo de menú	74
Anexo W. Flujo de servicios	75
Anexo X. Flujo de cita	76

	12
Anexo Y. Flujo de hora	76
Anexo Z. Flujo de recomendaciones	77
Anexo AA. Función de recordatorios	78
Anexo BB. Main	79
Anexo CC. Modelos empleados para el BackEnd	80
Anexo DD. Apartado de los Endpoints	83
Anexo EE. Front.	83
Anexo FF. App.Routes	86
Anexo GG. Services	87
Anexo HH. Pages	88
Anexo II. DIAGRAMA DE FLUJO DE STELLA SPA	96
Anexo JJ. GRÁFICOS DE ENCUESTAS	97

Resumen

Este proyecto aborda la problemática de la gestión manual y la falta de personalización en el consultorio estético Stella Spa, lo que impacta negativamente la experiencia del cliente. El objetivo principal fue desarrollar un Asistente Virtual inteligente, integrado vía WhatsApp, que automatice la gestión de citas, personalice las recomendaciones terapéuticas y haga un seguimiento eficiente del progreso del cliente.

El sistema fue diseñado utilizando una metodología ágil (SCRUM) e integra tecnologías clave como la API de OpenAI (GPT-3.5) para generar recomendaciones personalizadas, Baileys para la interacción por WhatsApp, y un stack tecnológico compuesto por Node.js, Angular y MongoDB. Este asistente permite a los clientes gestionar sus citas y recibir atención 24/7, mientras que el personal del spa accede a un panel administrativo para monitorear el historial de los clientes.

La implementación de esta solución se basó en evidencias concretas: el 50% de las citas no se confirmaban de inmediato y el 60% de los clientes no recibía atención personalizada. Los resultados obtenidos validan la eficacia del asistente virtual, mostrando que el 97.5% de los usuarios estaría dispuesto a usarlo y que el 85% prefiere WhatsApp como canal de interacción. La solución propuesta demuestra ser una herramienta eficaz para reducir la carga operativa, mejorar la eficiencia administrativa y elevar significativamente la calidad y personalización del servicio, posicionando a Stella Spa a la vanguardia tecnológica en el sector.

1. Introducción

En la actualidad, la transformación digital ha revolucionado la manera en que las empresas gestionan sus servicios y procesos. La automatización se ha convertido en una necesidad fundamental para mejorar la eficiencia operativa y satisfacer las demandas de los clientes. En el sector estético, donde la atención personalizada y la satisfacción del cliente son cruciales, las herramientas tecnológicas basadas en inteligencia artificial (IA) representan una solución innovadora y efectiva.

En el dinámico sector de la estética, la personalización de tratamientos y la eficiencia en la gestión de citas son factores clave para la satisfacción del cliente y el éxito del negocio (Onibokun et al., 2023). En la ciudad de Samborondón, Ecuador, donde la demanda de servicios estéticos de alta calidad está en constante crecimiento, los consultorios se enfrentan al reto de optimizar sus procesos de asignación de terapias para garantizar una experiencia del cliente fluida y efectiva (Jara, 2020). La gestión manual de agendas y la asignación de terapias basadas en la disponibilidad del personal y los recursos pueden resultar en errores, retrasos y una utilización subóptima de los recursos (Ingole, 2024).

Este proyecto propone el diseño de un asistente virtual basado en IA para el consultorio estético Stella Spa, ubicado en la Urbanización Castilla, vía a Samborondón. Actualmente, Stella Spa ofrece servicios de tratamientos faciales, corporales, masajes terapéuticos y asesoría en cuidados personales. No obstante, enfrenta problemas recurrentes, como la gestión manual de citas, el seguimiento ineficiente de terapias y la falta de personalización en la atención al cliente, lo que afecta la experiencia del usuario y la eficiencia operativa.

La relevancia de este tema se enmarca en las tendencias actuales de la tecnología. Según Sanabria y Frade (2023), más del 70% de los clientes valoran las aplicaciones basadas en IA por su capacidad para brindar respuestas rápidas y precisas. Además, se proyecta un incremento del 25% en la inversión en tecnología IA durante los próximos años, lo que evidencia la importancia de adoptar estas herramientas en el

contexto empresarial actual. Por otro lado, López (2020) destaca que los asistentes virtuales son clave para automatizar procesos y garantizar la disponibilidad constante del servicio, lo cual resulta especialmente valioso en consultorios estéticos como Stella Spa.

Este proyecto se inscribe dentro de las tendencias tecnológicas actuales al integrar inteligencia artificial, aprendizaje automático (Machine Learning) y procesamiento de lenguaje natural (PLN) para optimizar la gestión de citas, personalizar recomendaciones terapéuticas y realizar seguimientos automatizados. Esta solución no solo mejorará la eficiencia operativa, sino que también elevará la satisfacción de los clientes, posicionando a Stella Spa como un consultorio moderno e innovador en el sector estético.

1.1 Contexto histórico social del objeto de estudio

Los ideales de belleza continúan evolucionando, enfatizando la individualidad y el bienestar integral. La sociedad moderna valora la autenticidad y la autoexpresión, lo que impulsa la demanda de tratamientos estéticos personalizados que respeten y realcen las características únicas de cada persona. (Villanueva et al., 2024)

La pandemia de COVID-19 aceleró la adopción de tecnologías digitales en todos los sectores, incluyendo la estética. Los consultorios se vieron obligados a implementar soluciones online para mantener el contacto con sus clientes y ofrecer servicios de forma remota (Pujolar et al., 2022). Esta tendencia ha consolidado la importancia de herramientas como los asistentes virtuales para la gestión de citas, la comunicación con los clientes y la personalización de tratamientos.

Si bien el crecimiento económico puede ser variable, el aumento de la clase media en Samborondón genera una mayor demanda de servicios de calidad y experiencias personalizadas (Banco Mundial, 2025). Esta población, con mayor acceso a información y tecnología, busca soluciones eficientes y adaptadas a sus necesidades individuales.

Los consumidores actuales esperan una experiencia de la cliente fluida,

personalizada y omnicanal. Valorán la comodidad de poder acceder a información y servicios a través de múltiples canales, y esperan que las empresas conozcan sus preferencias y necesidades individuales (Accenture, 2024). Los asistentes virtuales pueden desempeñar un papel crucial en la creación de una experiencia del cliente excepcional y personalizada.

Las redes sociales siguen siendo una herramienta poderosa para la promoción de la belleza y el bienestar. El marketing de influencia, en particular, ha ganado relevancia, con influencers que comparten sus experiencias con tratamientos estéticos y recomiendan productos y servicios (Aswathi et al., 2022). Los asistentes virtuales pueden ayudar a los consultorios a gestionar su presencia en redes sociales, a interactuar con sus seguidores y a promocionar sus servicios de forma efectiva.

1.2 Antecedentes

La automatización de procesos mediante asistentes virtuales y herramientas de inteligencia artificial ha sido ampliamente estudiada e implementada en sectores como la salud, la educación y el servicio al cliente. Según Pareja y otros (2024), en 2020 se reportó el uso de 4.2 millones de asistentes virtuales a nivel global, y se proyecta un crecimiento a 8.4 millones para 2024, con un valor de mercado estimado en 11 mil millones de dólares para 2025. Este incremento refleja la importancia de estas tecnologías para optimizar interacciones y procesos empresariales.

En el ámbito de la salud, el asistente virtual SARAH, desarrollado por la OMS durante la pandemia de COVID-19, demostró ser una herramienta eficiente para ofrecer recomendaciones de salud y manejo del estrés las 24 horas del día (OMS, 2024). Aunque SARAH no realizaba diagnósticos médicos, su impacto resalta el potencial de la IA para mejorar la accesibilidad y personalización del servicio.

El consultorio Stella Spa, como los demás negocios pequeños, se enfrenta a desafíos relacionados como la dependencia de su personal para gestionar citas y brindar recomendaciones personalizadas. El modelo manual aumenta la carga operativa y el riesgo de errores, afectando la eficiencia como la experiencia del cliente.

Crear un asistente virtual para este consultorio buscar cerrar esa brecha, brindando una solución que no solo optimice estos procesos, sino también la atención al cliente empleando modelos de aprendizaje automático.

Otros estudios, como los de Apolo (2024), resaltan que los asistentes virtuales pueden operar 24/7, automatizando tareas rutinarias como la gestión de citas, recordatorios y seguimientos. Esto permite reducir errores humanos y optimizar la experiencia del cliente mediante respuestas ágiles y personalizadas.

Desde una perspectiva técnica, el PLN y el aprendizaje automático proporcionan una base sólida para desarrollar un asistente virtual eficiente y adaptable a las necesidades de un consultorio estético. Aunque estas tecnologías se han implementado con éxito en sectores como la salud y la atención al cliente, su aplicación en estética sigue siendo limitada, lo que refuerza la relevancia y el impacto potencial de este proyecto.

1.3 Planteamiento del problema

En el sector estético, la calidad del servicio depende de la rapidez, eficiencia y personalización de la atención al cliente. No obstante, en el consultorio Stella Spa, la gestión manual de citas y el seguimiento de tratamientos generan retrasos significativos y errores operativos. Esta situación depende en gran medida de la disponibilidad de la cosmetóloga y limita la capacidad del consultorio para atender la creciente demanda de servicios personalizados. Las soluciones actuales, basadas en modelos manuales, presentan las siguientes limitaciones:

- Retrasos en la programación y asignación de citas.
- Falta de seguimiento automatizado de terapias.
- Ausencia de personalización en las recomendaciones terapéuticas.

Actualmente, el proceso que se realiza en el consultorio de Stella Spa para la gestión de agendamiento y atención al cliente por medio de WhatsApp, lo cual representa una serie de limitaciones operativas. Según la cosmetóloga, la comunicación directa con los clientes recae completamente sobre ella, lo que genera

cuellos de botella significativos debido a su disponibilidad durante las jornadas laborales.

La información proporcionada durante la reunión, se identificaron puntos críticos:

- Aproximadamente un 50% de las citas no se confirman de forma inmediata, ya que la atención a los mensajes se ven retrasadas por la carga de trabajo o prácticas de la profesional
- Un 60% de los clientes experimentan una atención poco personalizada, ya que no se cuenta con un historial accesible en tiempo real que permita adaptar las recomendaciones o tratamientos
- El 20% de las recomendaciones post-tratamiento son seguidas adecuadamente, esto a que no existe una herramienta que recuerde al cliente cuidados posteriores a cada sesión
- Se estima que el tiempo para la administración de tareas de agendamiento, coordinación de horario y reprogramaciones toman un 40%.

Indico que la falta de una base de datos digital centralizada impide hacer un seguimiento optimo a los tratamientos de forma eficaz, lo cual afecta la satisfacción general con el servicio por parte del cliente.

Estudios previos han mostrado que la implementación de IA en los servicios puede automatizar tareas repetitivas, personalizar la atención y mejorar la eficiencia operativa. Según Sánchez (2022), los asistentes virtuales basados en IA permiten automatizar tareas, ofrecer respuestas inmediatas y generar recomendaciones personalizadas, mejorando significativamente la experiencia del cliente. Además, la IA ofrece la libertad de librarse de actividades recurrentes, como la elaboración de la bitácora de sesiones, agendamiento, imágenes, patología, prescripción de fármacos y revisión de las interacciones con los medicamentos. Estas tareas se ven apoyadas por los asistentes virtuales, optimizando la eficiencia del servicio médico y disminuyendo el desgaste del profesional.

Con esta implementación, se espera una notable mejora en la eficiencia operativa,

la personalización de tratamientos y el seguimiento continuo de los clientes, optimizando la experiencia del usuario y los resultados del consultorio. Según IBM (2020), los modelos de aprendizaje automático pueden analizar grandes volúmenes de datos para identificar patrones y anticipar necesidades, optimizando así los recursos y mejorando la experiencia del cliente. Este proyecto propone el desarrollo de un asistente virtual basado en inteligencia artificial que automatice estas tareas, permitiendo una atención más ágil, eficiente y adaptada a las necesidades individuales de los clientes.

Para el desarrollo del asistente virtual se empleará inteligencia artificial para interactuar con los clientes de forma natural. El asistente recogerá información clave de los usuarios, analizará sus necesidades y ofrecerá tratamientos personalizados, gestionará las citas y un seguimiento del tratamiento. Donde dichos datos serán almacenados en una plataforma web que será de apoyo para el terapeuta, permitiéndole visualizar los avances de cada terapia según el cliente y gestionar la agenda de citas.

Los resultados esperados a futuro incluyen:

- **Mejora cuantitativa:** Reducción del tiempo de gestión de citas en un 50 %, aumento en la satisfacción del cliente en un 30 %, y una optimización de los recursos del personal en un 40 %.
- **Mejora cualitativa:** Incremento en la personalización del servicio, garantizando una experiencia adaptada a las necesidades de cada cliente.

De no abordar este problema, Stella Spa continuará enfrentando retrasos operativos, insatisfacción de los clientes y una limitación en su capacidad de atender la creciente demanda. Por tanto, esta solución no solo mejorará la eficiencia operativa del consultorio, sino que también permitirá una transformación digital acorde a las necesidades del mercado actual.

1.4 Objetivos

Objetivo General

Integración de una plataforma inteligente basada en inteligencia artificial para optimizar la atención al cliente en el consultorio estético Stella Spa, mediante la automatización de la gestión de citas, la personalización de recomendaciones terapéuticas y el seguimiento efectivo del progreso de los clientes

Objetivos Específicos

- Analizar los procesos actuales de atención al cliente y gestión de citas en Stella Spa para identificar áreas críticas de mejora.
- Implementar una solución con la API de OpenAI para generar recomendaciones terapéuticas personalizadas, aprovechando su capacidad de procesamiento del lenguaje natural para adaptarse a las necesidades individuales de los clientes.
- Construir una plataforma integrada que automatice la gestión de citas, recordatorios y seguimientos de tratamientos en el consultorio
- Validar la funcionalidad y eficiencia de la plataforma inteligente mediante pruebas piloto con usuarios reales del consultorio Stella Spa

1.5 Justificación

El desarrollo e implementación de un asistente virtual basado en inteligencia artificial (IA) en el consultorio estético Stella Spa se fundamenta en la necesidad de resolver problemas recurrentes como la falta de automatización en la gestión de citas y tratamientos, la baja personalización de la atención al cliente y la dependencia del personal para realizar tareas administrativas. Esta situación genera demoras, errores y afecta la calidad del servicio ofrecido, especialmente en un contexto donde los clientes demandan soluciones rápidas, precisas y personalizadas.

La investigación realizada enriquece la comprensión del uso de la inteligencia artificial en servicios personalizados, destacando cómo los modelos de aprendizaje automático pueden analizar datos de clientes para ofrecer soluciones adaptadas a sus necesidades. El uso de IA en sistemas automatizados permite generar recomendaciones más precisas y optimizar procesos. Además, este proyecto contribuirá a explorar la transformación digital en pequeñas empresas y sectores

emergentes, como el estético.

Desde un enfoque metodológico, el desarrollo de este asistente virtual se basa en metodologías ágiles, que permiten una implementación iterativa y ajustable. Estas prácticas no solo optimizan el proceso de desarrollo, sino que también aseguran un producto final que responde a las necesidades del cliente y del usuario final.

En términos prácticos, el asistente virtual mejorará la eficiencia operativa del consultorio estético Stella Spa al automatizar tareas repetitivas como la asignación de citas y el envío de recordatorios, reduciendo la dependencia de la disponibilidad del personal. Por ejemplo, según un estudio de Pionce y otros (2022), los asistentes virtuales pueden disminuir los costos operativos entre un 30 % y un 70%, mientras garantizan disponibilidad las 24 horas del día. En el contexto de Stella Spa, esto se traducirá en un servicio continuo y personalizado, que permita a los clientes gestionar sus citas, recibir recomendaciones basadas en tratamientos previos y obtener información oportuna sobre sus terapias.

El impacto esperado incluye una reducción significativa en los errores administrativos, mejoras en la experiencia del cliente y una optimización del tiempo del personal, permitiendo que este se enfoque en tareas de mayor valor agregado. Además, el uso del asistente virtual fortalecerá la capacidad de Stella Spa para competir en un mercado cada vez más digitalizado y orientado hacia la personalización del servicio.

2. Marco Teórico

En este capítulo se establecen las bases conceptuales y teóricas necesarias para el desarrollo de un asistente virtual para la asignación y gestión de terapias personalizadas en el consultorio estético Stella Spa. Se examinan varios elementos esenciales de evolución, conceptos, herramientas tecnológicas para el desarrollo del asistente virtual, página web, base de datos, lenguajes de programación y metodologías de desarrollo ágiles. Estos principios ofrecen un entendimiento completo de los componentes y métodos que facilitaran el desarrollo de forma eficaz, atractiva y eficiente.

2.1 Asistentes Virtuales

Conocer al respecto sobre los asistentes virtuales resultara esencial para entender el impacto y relevancia en el desarrollo de este proyecto, permitiendo conocer sobre su origen, características básicas, entre otras. Estas definiciones y aplicaciones no solo se destacan por su aplicación en diversos sectores, sino también en mejorar la experiencia del usuario y optimizar tareas de manera más efectiva. Integrar estas tecnologías en mi proyecto requiere tener en cuenta aspectos clave como su evolución, los avances en inteligencia artificial y procesamiento de lenguaje natural, así como sus aplicaciones prácticas en campos como la educación, la salud y el entretenimiento. (Ramires & Valle, 2022)

Además, entender los fundamentos de su arquitectura tecnológica y sus contribuciones a distintos sectores asegura que el desarrollo de mi propuesta no solo aproveche las ventajas de estas herramientas, sino que también ofrezca soluciones innovadoras que potencien su utilidad en contextos específicos.

Un asistente virtual es un software interactivo cuyo propósito es simplificar la vida del usuario de aplicaciones mediante la automatización de las actividades y transformándose en un intermediario entre el ser humano y la máquina. La interacción se lleva a cabo a través del lenguaje natural y te brindan la posibilidad de hablar directamente con el dispositivo. (Contreras, 2023)

Los asistentes virtuales han demostrado ser una solución integral para diversas necesidades diarias. En el ámbito informativo, proporcionan acceso rápido a datos como noticias, actualizaciones de tráfico o resultados deportivos. Como asistentes personales, optimizan la gestión de tareas rutinarias, tales como programar recordatorios, administrar calendarios o realizar pedidos en línea. Además, su integración con tecnologías de hogar inteligente permite controlar dispositivos como luces, termostatos y cámaras de seguridad.

2.1.1. Aplicación en diversos sectores

En 2016, Domino's Pizza revolucionó su atención al cliente al introducir "Dom", un chatbot diseñado para gestionar pedidos y brindar información en tiempo real sobre entregas. Este avance tecnológico no solo facilitó el proceso de compra para los usuarios, sino que también impulsó un aumento del 63% en los pedidos realizados a través de su aplicación.

Por su parte, el banco español BBVA introdujo un asistente virtual llamado "Blue", diseñado para responder preguntas frecuentes sobre servicios financieros. Desde su integración, el banco logró aumentar en un 20% la resolución de consultas sin requerir intervención humana, lo que permitió redirigir los recursos hacia problemas más complejos. (Vorecol, 2024)

Estos casos demuestran que los asistentes virtuales son soluciones efectivas y replicables, incluso en empresas más pequeñas como Stella Spa. Donde actualmente no cuenta con un sistema automatizado que permita gestionar citas, dar seguimiento a tratamientos y personalizar recomendaciones. Generando:

- Retrasos en la atención al cliente debido a la gestión manual de agendas.
- Dificultad para recordar y sugerir tratamientos adecuados basados en el historial del cliente.
- Ineficiencia en el proceso de comunicación y seguimiento post-atención.

2.2 Implementación en la atención al cliente

Según Robio y otros (2022) la implementación de los asistentes virtuales puede

transformar la atención al cliente ofreciendo repuestas rápidas, personalizadas y eficientes. Al implementar un chatbot adaptado a las necesidades de nuestros clientes, podemos optimizar procesos como la gestión de reservas, la promoción de servicios y la resolución de consultas frecuentes. Esto no solo mejorará la experiencia del usuario, sino que también permitirá a Stella Spa aumentar la fidelización y maximizar la eficiencia operativa.

Según Espinoza (2023) el servicio al cliente se basa en estrategias diseñadas para garantizar una experiencia satisfactoria al usuario, proporcionando respuestas rápidas y precisas a sus inquietudes. Para lograrlo, las empresas recurren a múltiples canales de comunicación, como correos electrónicos, aplicaciones móviles, asistentes virtuales y redes sociales, que permiten una interacción eficiente y adaptada a las necesidades del cliente.

Este ámbito ha evolucionado significativamente, priorizando la rapidez, la facilidad de uso, la disponibilidad de múltiples canales, la personalización y, cuando es necesario, la atención humana. Los clientes demandan un servicio que atienda sus necesidades de manera integrada y adaptada a sus preferencias en todos los canales, destacando características como inmediatez, personalización, atención exclusiva y multicanalidad. (AceleraPyme, 2024)

2.2.1 Principales beneficios de los asistentes virtuales (ChatBots) Atención a la cliente ininterrumpida

Para Nicolescu y Tudorache (2022) la capacidad de operar las 24 horas, los 7 días de la semana, convierte a los chatbots en aliados fundamentales para las pymes, al proporcionar atención constante y resolver consultas comunes de forma inmediata. Esta disponibilidad mejora la experiencia del usuario y garantiza que, en caso necesario, las solicitudes puedan ser escaladas a un agente humano para una resolución personalizada.

2.2.2 Optimización de recursos y reducción de costos

La capacidad de interactuar con múltiples usuarios al mismo tiempo convierte a

los chatbots en herramientas económicas y altamente eficientes. No solo optimizan los recursos al reducir la necesidad de personal adicional, sino que también contribuyen al incremento de conversiones y al alcance de nuevos clientes, maximizando el impacto de la inversión inicial.

2.3 Ley de protección de datos

La Ley de Protección de Datos Personales, que entró en vigor en mayo de 2021, establece que todas las organizaciones, independientemente de su tamaño o sector, deben identificar y clasificar los datos personales que manejan, así como los medios en los que se almacenan y procesan. Es crucial que las entidades ajusten sus prácticas de tratamiento de datos a su estructura interna y modelo de negocio, asegurando que se cumpla con los principios de la ley. Este ajuste debe incluir un diagnóstico previo que permita evaluar el estado actual del cumplimiento legal y la seguridad del tratamiento de los datos.

Considerar la Ley de Protección de Datos en mi proyecto es fundamental para garantizar la privacidad y seguridad de los datos personales de los usuarios, especialmente en un entorno donde la información sensible puede ser manejada de manera automatizada. Al implementar asistentes virtuales en Stella Spa, es crucial cumplir con normativas que establece altos estándares de protección y transparencia. Esto no solo asegura el cumplimiento legal, sino que también fomenta la confianza de los clientes al demostrar un compromiso con el manejo ético y seguro de su información, alineándose con las mejores prácticas internacionales.

El artículo 66 numeral 19 de la Constitución de la República establece que las personas tienen derecho a la protección de sus datos personales, incluyendo el acceso y control sobre su información. La recolección, archivo, procesamiento, distribución o difusión de estos datos solo puede realizarse con la autorización del titular o conforme a lo que establezca la ley. (Datos, 2021)

Se establece en el artículo 8 de la Ley “el tratamiento y comunicación de datos personales estará permitido siempre que se obtenga el consentimiento explícito del

titular para hacerlo”. (Art. 8, Ley de Protección de Datos)

De acuerdo con el artículo 12 de la Ley “el titular de los datos personales tiene el derecho a ser informado de manera clara y transparente sobre cómo se procesarán sus datos”. (Art. 12, Ley de Protección de Datos)

Según Ponce (2024) las medidas implementadas por las organizaciones dependerán en gran medida del sector en el que operan y del tipo de datos que manejan. En función de este análisis, las organizaciones pueden determinar las medidas de seguridad más adecuadas para su negocio.

Para *Stella Spa*, esto implica: Implementar controles de acceso y medidas técnicas para proteger la información procesada por el asistente virtual. Cumplir con principios de consentimiento informado y la minimización de datos.

2.4 Desarrollo del ChatBot

Plataforma de desarrolladores de OpenAI

OpenAI es una organización dedicada a la investigación en inteligencia artificial, cuyo objetivo es asegurar que esta tecnología beneficie a toda la humanidad. Entre sus innovaciones clave se encuentra la plataforma de desarrollo, diseñada para permitir a los desarrolladores crear y entrenar modelos de IA de manera ágil y eficaz. (Magazine, 2024)

El modelo GPT-3.5, disponible sin costo para desarrollos básicos, permite integrar capacidades avanzadas de procesamiento de lenguaje natural (NLP). Esto es clave en un chatbot, ya que facilita interacciones más naturales y dinámicas con los clientes de *Stella Spa*.

2.5 Beneficios de emplearlo en el desarrollo

Flexibilidad: OpenAI permite conectarlo con otras herramientas, lo que resulta ideal para el proyecto, donde el asistente virtual no solo debe responder consultas, sino también gestionar citas y recomendar tratamientos. Su capacidad para manejar interacciones en múltiples idiomas, incluido el español, asegura que las respuestas sean claras y relevantes para los usuarios.

Escalabilidad: Capaz de procesar grandes volúmenes de datos y manejar proyectos de cualquier magnitud, lo que la convierte en una opción adecuada para organizaciones de diversos sectores.

2.6 Baileys whatsapp

El desarrollador utiliza @bot-whatsapp, una herramienta diseñada específicamente para facilitar la creación de chatbots orientados a WhatsApp. Esta solución permite construir bots conversacionales de forma rápida y eficiente, sin requerir configuraciones complejas. Además, cuenta con una guía práctica que incluye ejemplos y recursos útiles para acelerar el proceso de desarrollo. (Méndez, 2024)

2.7 Conceptos clave dentro de la herramienta Flow (Flujos de conversación)

Según Henao (2021) un flow representa la estructura de una conversación. Es decir, cómo el bot interactúa con el usuario a partir de palabras clave. Dentro de cada flujo se utilizan dos métodos fundamentales:

- ✓ **AddKeyword:** define las palabras o frases que el bot debe reconocer.
- ✓ **AddAnswer:** establece las respuestas automáticas que el bot debe enviar.

Estos métodos cuentan con múltiples opciones para personalizar el comportamiento del bot y construir diálogos más naturales y efectivos.

- ✓ **Provider (Proveedor de conexión):** El provider es el componente que enlaza tu chatbot con la plataforma de WhatsApp. Esta herramienta soporta diversos proveedores —muchos de ellos gratuitos— y también ofrece integración con servicios oficiales como WhatsApp Business API o Twilio.

2.8 Base de datos (DataBase)

La base de datos se encarga de almacenar el historial de las conversaciones. Según Khosravi y otros (2024) Esto es útil para personalizar la experiencia del usuario, dar seguimiento a interacciones pasadas o incluso segmentar clientes. La herramienta ofrece varios conectores para facilitar la integración con bases de datos comunes como MongoDB, asegurando una fácil adaptación al entorno de desarrollo. Entre ellos tenemos:

Económico y accesible: Al no requerir el uso de la API oficial de WhatsApp, Baileys permite desarrollar el bot sin incurrir en costos adicionales. Esto lo convierte en una excelente opción para negocios emergentes como Stella Spa.

Escalabilidad y flexibilidad: La solución basada en Baileys permite escalar fácilmente el proyecto en el futuro, ya sea integrando nuevos módulos, conectando otras bases de datos o incluso combinándolo con servicios externos como Twilio, sin necesidad de rehacer el sistema desde cero.

2.9 MongoDB

Es un sistema de gestión de bases de datos (DBMS, por sus siglas en inglés) no relacionales y de código abierto, que utiliza documentos flexibles en lugar de tablas y filas para procesar y almacenar varias formas de datos. Al ser una solución de base de datos NoSQL, MongoDB no requiere un sistema de gestión de bases de datos relacionales (RDBMS), por lo que proporciona un modelo de almacenamiento de datos elástico que permite a los usuarios almacenar y consultar fácilmente tipos de datos multivariados. Esto no solo simplifica la gestión de la base de datos para los desarrolladores, sino que también crea un entorno altamente escalable para aplicaciones y servicios multiplataforma. (IBM, 2021)

Los documentos o colecciones de documentos de MongoDB son las unidades básicas de datos. Estos documentos, con formato JSON (Java Script Object Notation) binario, pueden almacenar varios tipos de datos y distribuirse en varios sistemas. El diseño de esquema dinámico de MongoDB brinda a los usuarios una flexibilidad sin igual para crear registros de datos, consultar colecciones de documentos a través de la agregación de MongoDB y analizar grandes cantidades de información.

2.10 API'S

API significa “Interfaz de Programación de Aplicaciones”. Según Lamothe y otros (2021) son mecanismos capaces de permitir que dos componentes de programación puedan comunicarse entre ambos por medio de protocolos. La interfaz se puede considerar un contrato de servicio entre dos aplicaciones, que establece

cómo se comunican a través de solicitudes y respuestas. La documentación de la API proporciona detalles sobre cómo los desarrolladores deben estructurar estas solicitudes y respuestas para que las aplicaciones se comprendan mutuamente.

2.10.1 Funcionamiento

Las API facilitan la comunicación entre productos y servicios sin necesidad de conocer su implementación interna, simplificando así el desarrollo de aplicaciones y ahorrando tiempo y recursos. Proporcionan flexibilidad, permitiendo el diseño, la gestión y el uso de aplicaciones de manera más eficiente y ofreciendo oportunidades de innovación, lo que resulta crucial para crear herramientas y productos nuevos o para mejorar los existentes. (Lamothe y otros, 2021)

Se pueden considerar como contratos, ya que la documentación detallada establece cómo las solicitudes remotas se traducen en respuestas del sistema, estableciendo un acuerdo entre las partes involucradas. Además, facilitan la colaboración entre equipos comerciales y de TI al integrar fácilmente nuevos elementos de las aplicaciones en una arquitectura existente. Para Ehsan y otros (2022) en un entorno digital en constante cambio, las necesidades empresariales evolucionan rápidamente, y las API permiten responder ágilmente con servicios innovadores. El desarrollo de aplicaciones nativas de la nube, que se basa en la conexión de microservicios a través de API, es un método efectivo para acelerar el desarrollo y mantener la competitividad en un mercado dinámico.

Las API no solo conectan infraestructura interna, sino que también posibilitan compartir datos de manera segura con clientes y usuarios externos. Las API públicas son una herramienta clave para expandir las conexiones comerciales, facilitando asociaciones más amplias y rentabilizando datos valiosos, como en el caso de la API de Google Maps.

2.10.2 Beneficios de emplear apis's Aumento de la eficiencia

El uso de APIs con software de comisiones proporciona un aumento significativo en la eficiencia. Estas permiten que los datos se transfieran rápidamente

entre diferentes programas, eliminando la necesidad de ingresar manualmente la información. Como resultado, los cálculos de comisiones pueden realizarse más rápidamente y las declaraciones de comisiones se generan con mayor celeridad. Además, las APIs ayudan a minimizar los errores humanos que pueden surgir durante el ingreso manual de datos, mejorando aún más la eficiencia del proceso. (Lappalainen & Narayanan, 2023)

2.10.3 Mayor flexibilidad

Las APIs también brindan mayor flexibilidad cuando se integran con software de comisiones. Permiten la comunicación fluida entre diferentes programas, sin importar el lenguaje de programación o la plataforma en que se hayan desarrollado. Esto facilita la integración de software de comisiones con una amplia gama de programas, como CRMs, ERPs y software de contabilidad. Para las empresas que utilizan múltiples programas, esto permite una experiencia uniforme para los empleados y una gestión más sencilla de los datos. (De, 2023)

2.10.4 Reducción de costos

El uso de APIs con software de comisiones también ayuda a reducir costos. Al automatizar la transferencia de datos entre programas, se minimiza la necesidad de ingresar manualmente la información, lo que ahorra tiempo y recursos. Además, al reducir los errores a través de la automatización, se disminuyen las disputas y el tiempo necesario para resolverlas, lo que a su vez reduce los costos operativos.

2.11 Lenguajes de programación web

2.11.1 HTML y CSS

Para SW Team (2024) HTML HyperText Markup Language es el lenguaje estándar usado para estructurar contenido en páginas web. Es la columna vertebral del desarrollo web, ya que define la estructura y semántica de los elementos que componen una página, como encabezados, párrafos, imágenes, enlaces, tablas y más. Los navegadores web se encargan de interpretar el código HTML para mostrar el contenido de manera visual y comprensible al usuario. Aunque HTML no es

considerado un lenguaje de programación porque carece de características como variables, ciclos o declaraciones condicionales, es una herramienta indispensable en la creación de sitios web, ya que permite establecer un marco lógico y ordenado para todo el contenido.

Una de las fortalezas de HTML es su capacidad para utilizar etiquetas semánticas que no solo organizan la información, sino que también mejoran la accesibilidad y el SEO (Search Engine Optimization). Estas etiquetas ayudan a los motores de búsqueda a comprender mejor el propósito del contenido y hacen que las páginas sean más fáciles de navegar para usuarios con discapacidades. Además, HTML es compatible con otros lenguajes y tecnologías, lo que lo convierte en una pieza esencial en cualquier proyecto web.

Para complementar y enriquecer las capacidades visuales de HTML, se emplea CSS (Cascading Style Sheets). Este lenguaje de diseño permite separar el contenido de la presentación, lo que facilita la gestión y el mantenimiento de los estilos en un sitio web. CSS otorga a los desarrolladores un control detallado sobre aspectos visuales, como colores, tipografía, espaciado, diseño de cajas, sombras, animaciones y transiciones. A través de CSS, es posible transformar una estructura básica y funcional en una experiencia estética y visualmente impactante para el usuario.

2.11.2 Ventajas de usar CSS

Separación entre contenido y diseño

CSS facilita mantener un código HTML limpio y bien estructurado al separar la presentación visual en un archivo externo. Esto no solo mejora la legibilidad del código, sino que también simplifica el proceso de actualización y mantenimiento, permitiendo realizar cambios de estilo sin alterar la estructura del contenido.

2.11.3 Consistencia en el diseño

Los estilos definidos una vez pueden reutilizarse en múltiples páginas, asegurando una apariencia homogénea y profesional. Además, esta consistencia permite realizar ajustes globales con rapidez, simplificando la gestión del diseño.

2.11.4 Optimización del rendimiento

Al almacenar los estilos en un archivo externo, el navegador puede almacenarlos en caché y reutilizarlos para cargar las diferentes páginas del sitio. Esto reduce significativamente la cantidad de datos que se transfieren entre el servidor y el cliente, mejorando la velocidad de carga y la experiencia del usuario.

2.11.5 JavaScript

JavaScript es un lenguaje orientado a objetos que permite crear experiencias interactivas e inteligentes en la web. Su principal objetivo es lograr que las páginas respondan de manera eficiente y atractiva a las acciones del usuario, proporcionando interactividad y mejorando la usabilidad. Además, con el soporte de frameworks y bibliotecas modernas, como Angular o React, se puede desarrollar un front-end rápido, dinámico y visualmente impactante. (Kodigo, 2024)

Originalmente, JavaScript fue creado para potenciar el desarrollo front-end, permitiendo que las acciones del usuario dentro de una página web sean más dinámicas y responsivas. Esto significa que el código se ejecuta directamente en el navegador del usuario, facilitando la interacción en tiempo real con los elementos del sitio. No obstante, con la evolución de este lenguaje, también se ha adaptado al back-end, donde se ejecuta en el servidor, ampliando así su rango de aplicaciones.

Los desarrolladores tienen a su disposición una amplia gama de herramientas y técnicas para enriquecer la experiencia del usuario. Desde animaciones sofisticadas hasta actualizaciones de contenido sin recargar la página, este lenguaje permite que los sitios web sean más llamativos y funcionales, brindando un entorno interactivo y optimizado.

2.11.6 Desarrollo Back-End

Aunque inicialmente fue diseñado para el desarrollo front-end, JavaScript ha evolucionado hasta consolidarse como un lenguaje apto para el back-end. Esto ha sido posible gracias a Node.js, un entorno de ejecución que permite utilizar JavaScript en el servidor. Con Node.js, es posible desarrollar aplicaciones completas del lado del

servidor, integrando funcionalidades avanzadas y manteniendo un flujo eficiente entre cliente y servidor. (Montalvo & Iñíguez, 2025)

2.11.7 Node.JS

Node.js es un entorno de ejecución de JavaScript diseñado para crear aplicaciones de red escalables y en tiempo real. Basado en el motor V8 de Google Chrome, este software de código abierto permite ejecutar código JavaScript fuera del navegador, aprovechando su eficiencia y rapidez. Al ser una solución multiplataforma y de un solo hilo, Node.js se ha consolidado como una herramienta versátil para el desarrollo web moderno. (Acibeiro, 2025)

A diferencia de otros entornos, Node.js sigue un enfoque único al procesar solicitudes. En lugar de crear un nuevo hilo por cada petición, utiliza un solo hilo principal que administra todas las solicitudes entrantes mediante un bucle de eventos. Este hilo principal distribuye las tareas a hilos trabajadores (worker threads) según sea necesario, sin bloquearse mientras espera respuestas. Este modelo de E/S no bloqueante permite a Node.js gestionar de manera eficiente un gran número de peticiones simultáneas, maximizando el uso de recursos y optimizando el rendimiento.

2.11.8 Transmisión de datos

Las capacidades de Node.js para manejar flujos de datos lo hacen ideal para tareas como la transmisión de archivos o contenidos multimedia. Sus módulos integrados permiten procesar datos en tiempo real mientras se transfieren, reduciendo el tiempo de espera y optimizando recursos. Esto lo convierte en una excelente elección para empresas que ofrecen servicios de streaming o requieren manejar grandes volúmenes de datos sin interrupciones. (Acibeiro, 2025)

2.11.9 Compatibilidad entre frontend y backend

Node.js se distingue por su capacidad de soportar tanto el desarrollo del frontend como del backend. Esto permite reutilizar componentes de código, lo que no solo optimiza el tiempo de desarrollo, sino que también mejora la eficiencia al eliminar la necesidad de cambiar constantemente entre diferentes lenguajes de programación.

(Montalvo & Iñiguez, 2025)

2.11.10 AngularJS

AngularJS es un framework de código abierto diseñado para desarrollar aplicaciones web de una sola página, así como elementos interactivos y animados dentro de páginas HTML. (Deyimar, 2025)

Creado por los ingenieros de Google, Misko Hevery y Adam Abrons, AngularJS fue lanzado oficialmente en 2012 y desde entonces ha sido mantenido y mejorado por Google. Antes de su aparición, existían alternativas para crear páginas dinámicas, pero ninguna ofrecía la misma practicidad y eficiencia. AngularJS utiliza la arquitectura Modelo-Vista-Controlador (MVC), una estructura ampliamente empleada en el desarrollo web: el modelo maneja los datos y la lógica, la vista representa la información visualmente, y el controlador actúa como intermediario entre ambos, respondiendo a las acciones del usuario.

En este framework, AngularJS funciona como el modelo, HTML representa la vista y JavaScript se desempeña como el controlador. Lo que hace destacar a AngularJS es su capacidad para sincronizar automáticamente JavaScript y HTML mediante un mecanismo conocido como data binding, donde los cambios realizados en uno se reflejan de inmediato en el otro. Esta funcionalidad simplifica enormemente el desarrollo al reducir la cantidad de código manual que los programadores deben escribir, facilitando la creación de aplicaciones dinámicas y altamente interactivas.

3. Metodología

La investigación adopta un enfoque mixto, integrando datos cualitativos y cuantitativos para garantizar un análisis integral. El proceso de recolección, análisis y evaluación de datos se desarrollará en el consultorio estético Stella Spa, ubicado en la Urbanización Castilla, vía a Samborondón, durante el periodo comprendido entre noviembre de 2024 y junio de 2025.

Las metodologías ágiles de desarrollo de software tienen como objetivo entregar rápidamente pequeñas partes del software funcional, con el fin de aumentar la satisfacción del cliente. Estas metodologías se caracterizan por su enfoque flexible y el trabajo colaborativo, permitiendo mejoras continuas. Además, la metodología ágil promueve una documentación sencilla del software y está abierta a los cambios que puedan surgir en cualquier etapa del proceso, en lugar de resistirlos. (Red Hat, 2022)

Tienen un enfoque flexible y colaborativo para el desarrollo de software que permite adaptarse rápidamente a los cambios del mercado y las necesidades del cliente, mediante ciclos iterativos, retroalimentación constante y un enfoque en la entrega continua de valor.

3.1 Metodología SCRUM

La metodología ágil Scrum es un marco de trabajo diseñado para el desarrollo de proyectos en entornos cambiantes. Facilita la colaboración entre equipos, garantiza flexibilidad, fomenta la transparencia y permite cumplir con los requisitos del cliente. Según Estrada y otros (2021) SCRUM permite un desarrollo ágil basado en la autoorganización, empleando un enfoque incremental que optimiza la entrega de resultados. No obstante, es esencial distribuir adecuadamente el proyecto y garantizar una conformación efectiva de los equipos.

Destacando sus beneficios:

- ✓ Flexibilidad y Adaptabilidad: Permite responder rápidamente a cambios en los requisitos o condiciones del mercado.

- ✓ **Mejor Gestión del Tiempo y Recursos:** Su estructura incremental facilita un control más preciso de los plazos y los recursos necesarios.
- ✓ **Incremento de la Productividad:** La colaboración y autoorganización del equipo mejora la eficiencia en el trabajo.

Y la que tiene mayor relación con el desarrollo.

- ✓ **Satisfacción del Cliente:** El cliente se involucra activamente en el proceso, obteniendo entregas regulares que le permiten evaluar el progreso y sugerir mejoras.

Por esta razón Scrum es altamente adecuado mi proyecto debido a su capacidad para manejar entornos dinámicos y adaptarse rápidamente a cambios en los requisitos. Al dividir el trabajo en sprints, se garantiza que cada iteración ofrezca resultados funcionales que puedan ser revisados y mejorados en tiempo real. Además, la retroalimentación continua del cliente asegura que las necesidades y expectativas se cumplan en cada etapa. Esto es especialmente relevante con inteligencia artificial, donde los modelos deben ajustarse a nuevos datos y evolucionar constantemente. Los principios de autoorganización, transparencia y ciclos cortos de trabajo de Scrum se alinean perfectamente con los requerimientos de proyectos basados en aprendizaje y adaptación, maximizando la calidad y el valor del resultado final.

Product Owner

El Product Owner es uno de los roles existentes dentro de la metodología Scrum y como acabamos de ver, es el encargado de maximizar el valor del trabajo entregado por el equipo Scrum, además de gestionar el Product Backlog. Es una única persona, no un comité y para que pueda hacer bien su trabajo, es necesario que toda la organización respete sus decisiones. Debe conocer muy bien al cliente y tiene que contar también con una visión del valor que el equipo Scrum le está entregando.

En la guía de la metodología Scrum se identifican las siguientes tareas que se ajustan a nuestro proyecto:

Roles Claves

Product Owner: Podría ser uno de los autores del proyecto o incluso la doctora Estela Sarango (dueña de Stella Spa), ya que tiene contacto directo con las necesidades del cliente y del negocio.

Equipo de Desarrollo: Moreno Salcedo y Ramirez Ochoa

1. Inicio del proyecto – Definición del Product Owner y visión del producto

Se realizaron reuniones con la doctora Estela Sarango (dueña del spa) para conocer las necesidades reales del consultorio. Se definió el objetivo un asistente virtual que automatice citas, recomendaciones y seguimientos.

En Scrum: Se estableció al Product Owner, que conoce el negocio y define las prioridades del producto.

Se formuló una visión del producto clara, requisito clave en Scrum.

2. Recolección de requisitos y análisis del problema – Product Backlog inicial

Análisis del proceso actual de agendamiento.

Identificación de los puntos críticos mediante entrevistas y observación directa.

Detección funcionalidades necesarias.

En Scrum: Se construyó un Product Backlog inicial, es decir, una lista priorizada de funcionalidades necesarias para el asistente virtual.

El Product Owner gestionó este backlog, validando con el cliente.

3. Diseño de la solución – Sprint Planning

Diseño de la arquitectura de la plataforma.

Selección de herramientas: OpenAI, WhatsApp, MongoDB, etc.

Definición de flujos conversacionales y módulos clave.

En Scrum: Esto corresponde a la Sprint Planning, donde el equipo selecciona qué elementos del Product Backlog se desarrollarán en el sprint y cómo lo harán.

4. Desarrollo e integración – Sprints de trabajo

Implementación del chatbot, la conexión con WhatsApp, integración con la API de OpenAI, almacenamiento en MongoDB, etc.

Se desarrollaron de forma incremental los módulos: citas, recomendaciones, historial.

En Scrum: Aquí se ejecutaron múltiples sprints, cada uno con entregables funcionales. Se mantuvo una colaboración constante entre el Product Owner y el equipo de desarrollo.

5. Pruebas piloto – Sprint Review y Retrospective Realización de pruebas con usuarios reales del spa. Recibieron retroalimentación directa del uso del sistema. Ajustes basados en comentarios.

En Scrum: Esto es parte del Sprint Review (evaluar el incremento entregado). Posteriormente, el equipo realiza la Retrospective para reflexionar sobre el proceso y mejorar en el próximo sprint.

6. Capacitación, entrega y cierre – Entrega final Integraron completamente la solución. Capacitaron a la doctora.

Elaboraron el manual y recomendaciones.

En Scrum: Se considera el cierre de un release, una entrega completa y usable del producto que cumple con el “Definition of Done”.

Tabla 1.

Descripción de la fase de la metodología Scrum

Etapas del Proyecto	Práctica Scrum	Resultado Logrado
	Equivalente	
Análisis con la doctora	Product Owner + Visión del producto	Se entendieron las necesidades del cliente
Identificación de funcionalidades	Product Backlog	Lista priorizada de tareas
Diseño técnico y flujos	Sprint Planning	Plan detallado del desarrollo
Desarrollo incremental	Sprints	Implementación funcional por partes

Pruebas con usuarios reales	Sprint Review + Retrospective	Feedback útil y mejoras aplicadas
Capacitación e implementación final	Entrega del incremento	Sistema listo, funcional, con uso real

Elaborado por Moreno y Ramírez, 2025.

3.2 Plan de Trabajo – Cronograma de Actividades

Plan de Trabajo para Plataforma Inteligente en Stella Spa

✓ Fase 1: Análisis y Recolección de Requisitos

Tiempo: Semanas 1-2

Objetivo relacionado: Analizar los procesos actuales de atención al cliente y gestión de citas.

Tareas:

- Reuniones con el personal del consultorio para mapear procesos actuales.

Durante la primera semana, se realizaron reuniones con la Doctora de Stella Spa: Estas sesiones se desarrollaron en un ambiente colaborativo donde se describieron paso a paso los procesos actuales, desde la recepción de un cliente hasta la finalización de su servicio.

En conjunto, con la Dotora ESTELA, se visualizó cómo recibe las llamadas, anota manualmente las citas en una libreta y coordina los horarios disponibles, lo que a menudo genera duplicación de horarios o citas perdidas.

- Identificación de puntos críticos y cuellos de botella.

A partir de las reuniones, surgieron varios puntos críticos. Uno de los más relevantes fue la dificultad para gestionar múltiples citas al mismo tiempo, especialmente cuando hay cambios de último minuto. Se detectaron también inconsistencias en la asignación de turnos y tiempos de servicio, lo que afecta la experiencia del cliente.

En este análisis también se identificó que no existe un sistema automatizado para

recordar citas, lo que provoca ausencias o retrasos frecuentes.

- Recolección de feedback de clientes actuales.

Se realizaron entrevistas breves con varios clientes frecuentes de Stella Spa. Algunos participaron mientras esperaban su turno y otros fueron contactados vía WhatsApp con preguntas simples.

Una clienta mencionó: “Me encanta el trato, pero a veces he llegado y mi cita no estaba agendada. Me gustaría poder confirmar todo por una app.” Otro cliente expresó que valoraría mucho poder agendar fuera del horario de atención, ya que trabaja todo el día.

Este feedback resultó crucial para priorizar funcionalidades como recordatorios automáticos, historial de servicios, y autogestión de citas.

- Documentación de flujos de trabajo actuales.

Se elaboraron diagramas de flujo (BPMN) que representan las etapas actuales de los procesos clave: agendamiento, confirmación, cancelación, atención al cliente, y seguimiento post-servicio. Esta visualización permitió evidenciar los puntos donde se pierde eficiencia o información.

Entregables:

- Documento de requerimientos funcionales y no funcionales: Se detallaron las necesidades del sistema desde el punto de vista del usuario (funcionales) como la creación, modificación y recordatorio de citas, así como los aspectos técnicos (no funcionales) como disponibilidad 24/7, interfaz intuitiva, integración con WhatsApp, y escalabilidad futura.
- Flujo actual

Fase 2: Diseño de la Solución (Semanas 3-4)

Objetivo relacionado: Preparación para la implementación de la plataforma y personalización de la API.

Tareas:

- Selección de tecnologías (OpenAI API, backend, frontend, base de datos).

La elección de las tecnologías como Baileys, la API de OpenAI, MongoDB, los lenguajes de HTML, CSS, JavaScript y Node.js son fundamentales para el desarrollo eficiente y escalable del chatbot para Stella Spa. OpenAI proporciona una potente IA que permite mantener conversaciones naturales gracias a su modelo GPT-3.5, facilitando la atención automatizada en lenguaje natural. Baileys, permite una integración directa y económica con WhatsApp, que será el canal de comunicación con los clientes, y ofrece flexibilidad para personalizar los flujos de conversación y gestionar citas.

MongoDB aporta una base de datos NoSQL, escalable y robusta que almacena el historial de interacciones y agendamientos, mejorando la experiencia del usuario. Los lenguajes como JavaScript y Node.js permiten desarrollar tanto el backend como el frontend del sistema asegurando una respuesta rápida, interacción en tiempo real y fácil integración con APIs externas.

- Diseño de arquitectura de la plataforma.

La arquitectura contará con diferentes ambientes; para el chatbot se tiene la importación de las librerías de `bailayse wsapp`, y su conexión a la base de datos (MongoDB) que almacenará las conversaciones que tenga con el paciente en tiempo real. A su vez, el ambiente web que contará con su frontend con las librerías de Angular, Javascript, npm y PrimeNG como framework, respectivamente conectado con su backend conectado a la base de MongoDB empleando las librerías de nodejs y los servicios de consulta.

- Diagramas de flujo y wireframes de la plataforma.
- Diseño de casos de uso para la generación de recomendaciones personalizadas.

Entregables:

- Gráfica de la arquitectura del proyecto.

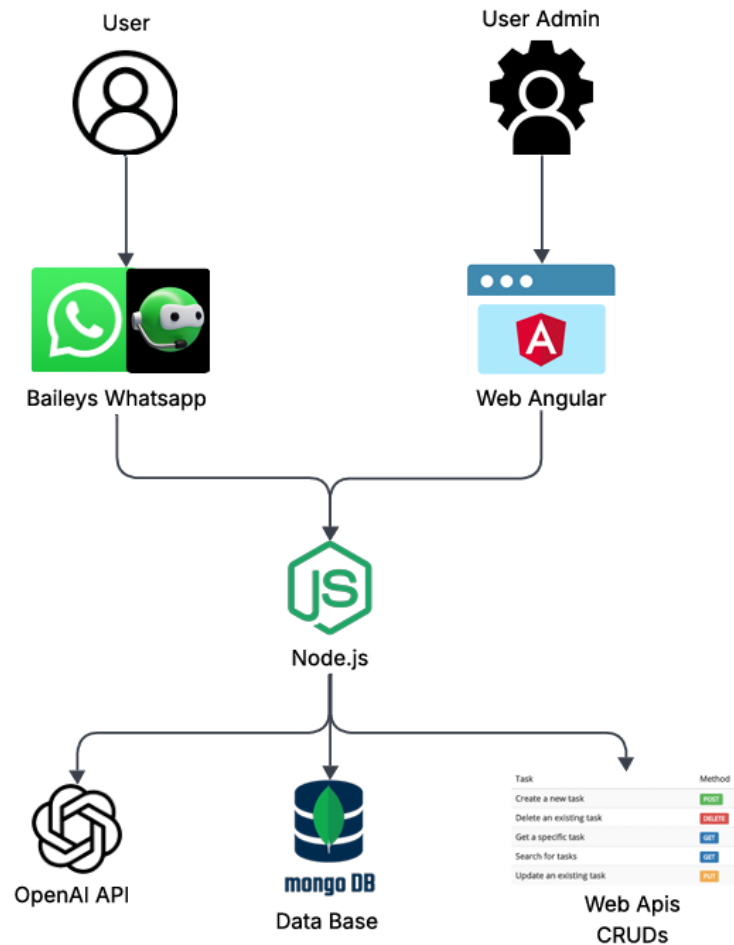


Figura 1. Estructura del modelo

Elaborado por los autores

El proceso empezó con la obtención de las credenciales de autenticación (API key), gestionadas a través de la plataforma para desarrolladoras de OpenAI. Estas credenciales permiten un acceso seguro a los servicios de la API mediante solicitudes HTTP bajo el protocolo de HTTPS.

Empleando los siguientes componentes:

- **INTERFAZ**

WhatsApp:

Usando Baileys para el manejo de los flujos de conversaciones

Web:

Desarrollado en Angular v19 y de framework PrimeNG, donde se consumen los

servicios del backend GET, POST, PUT, DELETE

- **ASISTENTE VIRTUAL**

OpenAI API (GPT):

Provee las respuestas en lenguaje natural con respecto a la información de los servicios que se obtiene desde un apartado de la base que lo alimenta y las recomendaciones que las brindara a partir de un análisis de las terapias ya realizadas al usuario

- **Backend**

Desarrollo en Node.js

Expone una API REST que la consume Angular y Baileys

Conecta con: MongoDB, OpenAI, Baileys y servicios web para operaciones CRUD

- **Base de datos**

MongoDB: Usuarios

Colecciones empleadas

- o Agendas
- o Historialclinfobas
- o Pacientes
- o Servicios
- o Tratamientos
- o Users

La arquitectura del sistema contempla que el backend realice llamadas a la API enviando los mensajes del usuario junto con los parámetros de configuración del modelo, el número máximo de tokens, ajustes del flujo de los mensajes y respuestas. La respuesta generada por el modelo es procesada por el servidor y es entregada al usuario final por medio del canal correspondiente en este caso WhatsApp.

Fase 3: Desarrollo e Integración (Semanas

5-10) Objetivos relacionados:

- Implementar solución con API de OpenAI.

- Construir plataforma automatizada.

Tareas:

- Configuración del entorno de desarrollo.

Se configuró un entorno de desarrollo empleando Node.js como motor principal para el backend y Angular para la interfaz gráfica de la web. Además, se incorporó herramientas como postman para las pruebas de los endpoints y nodemon para facilitar el reinicio automático del servidor durante el desarrollo. El entorno también incluyó la instalación y configuración de MongoDB como base de datos NoSQL, permitiendo el almacenamiento flexible de información de pacientes, citas y tratamientos.

- Desarrollo del backend con lógica de citas y tratamientos.

Se construyó un backend con Express.js, estructurado en rutas, controladores y servicios. La lógica incluye la creación, modificación, consulta y cancelación de citas, así como el registro de tratamientos realizados o programados para cada paciente. Se implementó validación de datos, manejo de errores y control de acceso básico para garantizar integridad y seguridad. Los datos se almacenan y gestionan eficientemente en MongoDB, con modelos definidos mediante Mongoose.

- Integración con la API de OpenAI para recomendaciones personalizadas.

El backend se integró con la API de OpenAI para generar recomendaciones automatizadas de tratamientos según el historial del paciente y sus necesidades. Esta integración se realiza mediante solicitudes HTTP autenticadas con claves API. El sistema recopila información contextual del paciente y la envía a la API para obtener sugerencias relevantes, las cuales son devueltas al usuario a través del canal correspondiente, como parte de la conversación en WhatsApp o desde el panel administrativo.

- Desarrollo de recordatorios automáticos y sistema de seguimiento.
- Desarrollo del frontend (panel administrativo y vista cliente).
- Pruebas unitarias.

Se desarrollaron pruebas unitarias enfocadas en la lógica crítica del backend, especialmente en la creación y modificación de citas, validaciones de datos y comunicación con el servicio externo de OpenAI.

Entregables:

- Plataforma funcional (v1).



Figura 2. Pantalla web de inicio

Elaborado por los autores



Figura 3. Menú de opciones

Elaborado por los autores

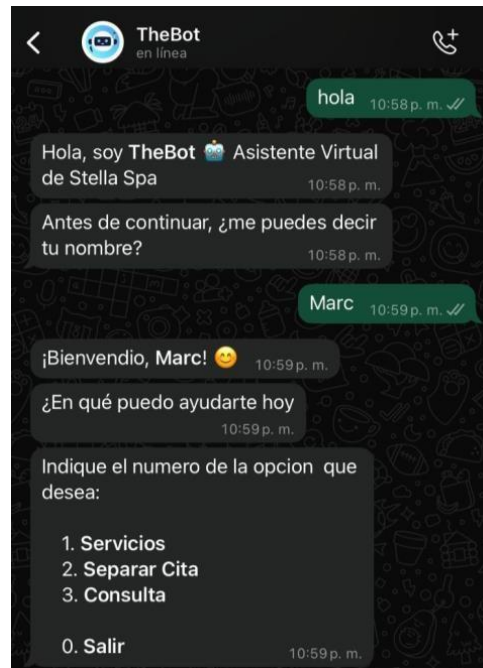


Figura 4. Interacción con el ChatBot

Elaborado por los autores

- Módulo de citas y recomendaciones listo para pruebas internas.

Fase 4: Pruebas Piloto y Validación (Semanas 11-13)

Objetivo relacionado: Validar funcionalidad y eficiencia de la plataforma.

Tareas:

- Selección de grupo piloto de clientes y personal del spa.
- Ejecución de pruebas de usabilidad y funcionalidad.
- Recopilación de retroalimentación.
- Ajustes técnicos y de interfaz según resultados.

Entregables:

- Informe de pruebas piloto.
- Plataforma ajustada con base en feedback real.

Fase 5: Integración Final y Capacitación (Semanas 14-15) Tareas:

- Integración final
- Capacitación al personal de Stella Spa.
- Creación de manual de uso y soporte básico.

Entregables:

- Sistema en producción.
- Manual de usuario.
- Registro de capacitación.

Fase 6: Evaluación y Cierre del Proyecto (Semana 16) Tareas:

- Evaluación del cumplimiento de objetivos.
- Presentación de resultados finales al equipo directivo.
- Recomendaciones para mantenimiento y escalabilidad futura.

Entregables:

- Informe final del proyecto.
- Lista de mejoras sugeridas a largo plazo.

☐	Tipo	Clave	Resumen	Estado	Comentarios	Sprint	Persona asignada
☐	▼ ⚡	SCRUM-1	Servicio Bailyse	EN PROGRESO	➡ Añadir comentario		
☐	☑	SCRUM-2	Instalar dependencias	HECHO	➡ Añadir comentario		TR Tommy Ramirez
☐	⚙	SCRUM-3	Revisar Documentación y curso de la herramienta	HECHO	➡ Añadir comentario		AM Andre Moreno
☐	⚙	SCRUM-4	Conexion a base MONGODB	HECHO	➡ Añadir comentario		AM Andre Moreno
☐	⚙	SCRUM-5	Primeras consultas y respuestas del bot	HECHO	➡ Añadir comentario		TR Tommy Ramirez
☐	⚙	SCRUM-6	Problemas con la interaccion del bot	HECHO	➡ Añadir comentario		TR Tommy Ramirez
☐	📄	SCRUM-7	Almacenamiento de los datos en base	HECHO	➡ Añadir comentario		AM Andre Moreno
☐	☑	SCRUM-14	Nuevas funciones para la descripcion de los servicios	HECHO	➡ Añadir comentario		AM Andre Moreno
☐	▼ ⚡	SCRUM-8	MongoDB	POR REALIZAR	➡ Añadir comentario		
☐	☑	SCRUM-9	Creacion de la base	HECHO	➡ Añadir comentario		TR Tommy Ramirez
☐	☑	SCRUM-10	Creacion de las colecciones y los campos	HECHO	➡ Añadir comentario		
☐	📄	SCRUM-11	Revison de los campos almacenados	HECHO	➡ Añadir comentario		AM Andre Moreno

Figura 5. Jira Tareas Asignadas

Nota: cronograma de actividades, (Moreno & Ramírez 2025).

Actividades de Jira**TAREA: Instalar dependencias:**

Se instalaron las dependencias necesarias que se usaran para el desarrollo.

```

10 package.json
11 },
12 "private": true,
13 "dependencies": {
14   "@angular/animations": "^19.1.0",
15   "@angular/common": "^19.1.0",
16   "@angular/compiler": "^19.1.0",
17   "@angular/core": "^19.1.0",
18   "@angular/forms": "^19.1.0",
19   "@angular/platform-browser": "^19.1.0",
20   "@angular/platform-browser-dynamic": "^19.1.0",
21   "@angular/router": "^19.1.0",
22   "@primeng/themes": "^19.0.9",
23   "body-parser": "^1.20.3",
24   "cors": "^2.8.5",
25   "express": "^4.21.2",
26   "mongoose": "^8.12.1",
27   "primeicons": "^7.0.0",
28   "primeng": "^19.0.9",
29   "rxjs": "~7.8.0",
30   "tslib": "^2.3.0",
31   "zone.js": "~0.15.0"
32 },
33 "devDependencies": {
34   "@angular-devkit/build-angular": "^19.1.6",
35   "@angular/cli": "^19.1.6",
36   "@angular/compiler-cli": "^19.1.0",
37   "@types/jasmine": "~5.1.0",
38   "jasmine-core": "~5.5.0",
39   "karma": "~6.4.0",
40   "karma-chrome-launcher": "~3.2.0",
41   "karma-coverage": "~2.2.0",
42   "karma-jasmine": "~5.1.0",

```

Figura 6. Dependencias del proyecto

Elaborado por los autores

TAREA: Revisar Documentación y curso de la herramienta

Por ser la primera vez en emplear estas herramientas, se procedió a leer la documentación correspondiente para la adecuada instalación y configuración.

- **Angular:** <https://angular.dev/overview>
- **Nodejs:** <https://nodejs.org/es>
- **Mongo:** <https://www.mongodb.com/try/download/community>

TAREA: Conexión a base MongoDB

Se procedió a realizar la respectiva conexión para el ChatBot que almacenara las conversaciones y consultas, también el ambiente web que consta de dos partes el frontend el cual llamara a los servicios del backend.

Front.

```

src > app > services > paciente.service.ts > PacienteService > addTratamiento
1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable } from 'rxjs';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class PacienteService {
9
10     private apiUrl = 'http://localhost:5000/paciente';
11
12     constructor( private http: HttpClient ) {
13
14     }
15
16     //obtener los usuarios
17     getPacientes(): Observable<any[]> {
18         return this.http.get<any[]>(this.apiUrl);
19     }

```

Figura 7. Librerías del front

Elaborado por los autores

```

10  const HistorialClInfoBas = require('./models/historial_cl_info_bas');
11  const Tratamiento = require('./models/tratamientos');
12
13  const app = express();
14  const PORT = 5000;
15
16  // Middleware
17  app.use(bodyParser.json());
18  app.use(cors());
19
20  // MongoDB connection
21  //base
22  mongoose.connect('mongodb://127.0.0.1:27017/usuarios', {
23    useNewUrlParser: true,
24    useUnifiedTopology: true,
25  });
26
27  const db = mongoose.connection;

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

```

PS E:\Marcell\Escritorio\CHATBOT\WEB\w_Stella\backend> node server.js
(node:10568) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: use
ion 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:10568) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option:
r version 4.0.0 and will be removed in the next major version
Server running on http://localhost:5000
Connected to MongoDB

```

Figura 8. Librerías del back

Elaborado por los autores

```
//base
let adapterDB;

// Inicializar la conexión con MongoDB
const conectarMongoDB = async () => {
  adapterDB = new MongoAdapter({
    dbUri: 'mongodb://127.0.0.1:27017/', // Dirección de MongoDB
    dbName: "usuarios"
  });

  console.log("✅ Conexión con MongoDB establecida correctamente.");
};
```

Figura 9. Librerías del Bot

Elaborado por los autores

TAREA: Primeras consultas y respuestas

Con las guías revisadas, se llevó a cabo los flujos que pueden existir en una conversación con el Bot y las acciones que este tomara.

```
.addAnswer("Hola, soy *TheBot* 🤖 Asistente Virtual de Stella Spa")
.addAnswer("Antes de continuar, ¿me puedes decir tu nombre?", { capture: true }, async (ctx,
  nombreUsuario = ctx.body;
  const telefono = ctx.from;

  const pacientesCollection = adapterDB.db.collection('pacientes');
  const pacienteInfo = adapterDB.db.collection('historialclinfobas');

  try {
    // Verifica si ya existe el paciente
    const pacienteExistente = await pacientesCollection.findOne({ telefono });

    if (!pacienteExistente) {
      // Si no existe, lo inserta
      await pacientesCollection.insertOne({
        nombre: nombreUsuario,
        telefono
      });
    }
  }
};
```

Figura 10. Flujos de eventos

Elaborado por los autores



Figura 11. Interfaz de respuesta

Elaborado por los autores

TAREA: Almacenamiento de los datos en la base

Como ya tenemos ingresos de datos, se establecieron las colecciones que va almacenar MongoDB para cada gestión por separado para evitar que este en una sola colección y facilitar el manejo de los datos

localhost:27017 > usuarios		Open MongoDB shell	Create collection	Refresh
Sort by	Collection Name			View
agendas				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	16	119.00 B	1	36.86 kB
historialclinforbas				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	2	421.00 B	1	36.86 kB
history				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
32.77 kB	158	352.00 B	1	36.86 kB
pacientes				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	5	86.00 B	1	36.86 kB
tratamientos				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	2	221.00 B	1	36.86 kB

Figura 12. Colecciones de la base de datos

Elaborado por los autores

Agendas: todas las citas agendadas

Historialclinfobas: información general de los pacientes

History: el historial del chat

Pacientes: pacientes que crearon sus cuentas

Tratamientos: información de las sesiones

TAREA: Nuevas funciones para la descripción de los servicios

Como se mencionó anteriormente se desarrollaron diferentes flujos para las acciones que tomara el Bot en los eventos que se presenten.

```
//servicios llamados para cada una de las areas
const flowServicios = addKeyword(EVENTS.ACTION)
  .addAnswer(areas)
  .addAnswer("Ingrese el numero del servicio para obtener mas detalles 🔍",
    { capture: true },
    async (ctx, { gotoFlow, fallBack, flowDynamic }) => {
      if (!["1","2","3","4","5","6","7","8","0"].includes(ctx.body)) {
        return fallBack(
          "Respuesta no válida, por favor selecciona una de las opciones."
        );
      }
      switch (ctx.body) {
        case "1":
          return gotoFlow(flowFacial);
        case "2":
          return gotoFlow(flowCorporal);
        case "3":
          return gotoFlow(flowEstetica);
        case "4":
          return gotoFlow(flowEspecial);
        case "5":
          return gotoFlow(flowGluteos);
      }
    }
  );
```

Figura 13. Flujo de servicios 1

Elaborado por los autores

```
//entran el flujo para el gpt y responder las consultas
const FlowConsulta = addKeyword([EVENTS.ACTION])
  .addAnswer("Claro, dime tu consulta: ", { capture: true }, async (ctx, ctxFn) => {
    const prompt = resBot;

    const consulta = ctx.body;
    const answer = await chat(prompt, consulta);

    await ctxFn.flowDynamic(answer.content);

    // // Preguntar si tiene otra duda...
  })
  .addAnswer("Tienes otra duda?")
  .addAnswer("Ingrese si / no", {capture: true}, (ctx, ctxFn) => {...
  })
  .addAnswer("Listo, me alegra haber ayudado, si necesitas algo mas no dudes en preguntar")
```

Figura 14. Flujo de servicios 2

Elaborado por los autores

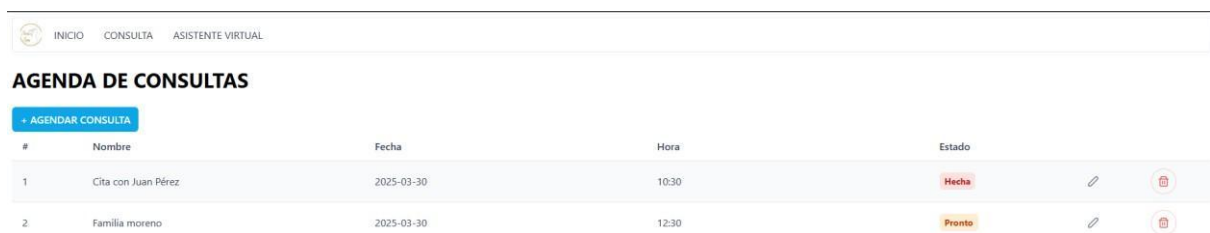
TAREA: MongoDB

Se realizó la junta para analizar que colecciones serán creadas, como se relacionan entre ellas y que información tendrán. Pero como se sabe MongoDB no es una base de datos relacional, se estableció que el número de celular será el ID primario para todas las tablas, al ser un valor que es único para cada paciente.

TAREA: Funciones para el ambiente web

- Agenda de consultas

Apartado para revisar un listado de las citas creadas en el ambiente del Bot, cuenta también con el botón para agendar una cita desde la página web. Se permite editar y eliminar las citas deseadas.



#	Nombre	Fecha	Hora	Estado		
1	Cita con Juan Pérez	2025-03-30	10:30	Hecha		
2	Familia moreno	2025-03-30	12:30	Pronto		

Figura 15. Agenda de consultas

Elaborado por los autores



AGENDAR CITA

Numero de telefono:

Fecha:

Hora:

Agendar

Figura 16. Agendar cita

Elaborado por los autores

- Consulta

Se encuentran los perfiles para cada uno de los pacientes registrados.

< ↺ >

#	Paciente	
1	andre	Perfil
2	Juan Pérez	Perfil
3	Marc	Perfil

Figura 17. Perfiles

Elaborado por los autores

Con su redirección a una pantalla en donde se visualizará información del paciente, con respecto a datos de salud, citas y tratamientos.

Editar

Guardar

Paciente:

Información Básica

Nombre completo:
Marcell Andre Moreno Salcedo

Fecha de nacimiento:
2000-06-14T00:00:00.000Z

Edad:
24

Historial de consultas

Nombre	Fecha	Hora
Oli	1970-01-01	11:00
Marc	2024-10-15	11:00
Marc	2025-04-18	11:11
Marc	2025-04-21	15:55
Marc	2025-06-10	10:00
Marc	1970-01-01	12:00
Marc	2025-06-09	15:00

Historial de procedimientos

+ PROCEDIMIENTO

Fecha	Tipo de tratamiento	Recomendaciones post-sesión	Evolución o seguimiento	Observaciones		
2025-06-01T21:09:26.174Z	acne	jugos verdes, agua, frutas	Mejora notable en la piel, menos brotes de acne	Aplicar crema hidratante por las noches y evitar el sol directo		

Figura 18. Información del paciente

Elaborado por los autores

- Bot

Aparatados para añadir información que alimentara el conocimiento del asistente virtual que ayudara a responder preguntas a los clientes


INICIO AGENDA CONSULTA



Combos

Description

hola

Save



Tratamientos Corporales

Description

Descripción del tratamiento: Explica en qué consiste cada procedimiento de forma breve y fácil de entender.

Beneficios principales: Detalla qué mejoras o resultados puede esperar el cliente tras el tratamiento.

Recomendaciones previas: Indica si hay algo que el cliente deba hacer o evitar antes de realizar el tratamiento.

Recomendaciones posteriores: Brinda consejos para maximizar los resultados después del tratamiento.

Duración del tratamiento: Indica cuánto tiempo dura cada sesión.

Frecuencia recomendada: Sugiere cada cuánto tiempo es ideal.

Save



Tratamientos de Depilación

Description

Información general para tratamientos de estética y depilación

Cada procedimiento incluye detalles clave para brindar una atención personalizada y segura. La información está organizada en las siguientes categorías:

Áreas del cuerpo disponibles: Indica qué zonas se pueden tratar.

Beneficios principales: Ventajas del tratamiento, como suavidad, durabilidad o efectos estéticos visibles.

Save



Tratamientos de extremidades inferiores

Description

Eres un asistente virtual especializado en tratamientos de glúteos y escultura corporal. Brinda respuestas claras y detalladas sobre esta categoría, enfocándose en los beneficios, procedimientos y recomendaciones. Asegure de incluir la siguiente información clave:

Descripción del tratamiento: Explica en qué consiste cada procedimiento de manera sencilla y profesional.

Objetivo principal: Indica cuál es el propósito del tratamiento (tonificación, aumento de volumen, reducción de grasa, etc.).

Beneficios principales: Detalla los resultados esperados, como

Save



Tratamientos Especiales

Description

Eres un asistente virtual especializado en terapias especiales. Tu objetivo es brindar respuestas claras, detalladas y profesionales sobre los tratamientos disponibles en esta categoría. Proporciona información clave sobre los siguientes puntos:

Descripción del tratamiento: Explica de manera breve y comprensible en qué consiste cada terapia especial.

Beneficios principales: Detalla los resultados esperados, como alivio, bienestar o mejoras físicas o emocionales.

Condiciones ideales: Indica para qué tipo de personas o situaciones

Save



Tratamientos Faciales

Description

Eres un asistente virtual experto en tratamientos faciales de un centro de estética. Al cliente, proporciona información clara y breve sobre los tratamientos faciales disponibles. Para cada tratamiento, menciona:

Su función principal.

Los beneficios para la piel.

La duración estimada.

Las recomendaciones según el tipo de piel o necesidades (si aplica).

También se cortas y se educado con el cliente

Save

Figura 19. Alimentación de información para el bot.

Elaborado por los autores

4. Análisis e interpretación de resultados

El desarrollo del asistente virtual para Stella Spa, implementado bajo la metodología ágil Scrum, evidenció resultados positivos en cada fase del proyecto, desde la recolección de requerimientos hasta la entrega final del sistema. El enfoque iterativo permitió una construcción progresiva de la solución, adaptándose a los cambios identificados durante el proceso mediante la participación de la doctora Estela Sarango como Product Owner, lo cual facilitó una comprensión profunda del contexto real del negocio y una priorización efectiva de funcionalidades.

La verificación de problemas como la duplicación de citas, la falta de recordatorios automáticos y la falta de disponibilidad fuera del horario de atención, permitió diseñar una solución integral orientada a la eficiencia operativa y la satisfacción del cliente. La incorporación de tecnologías como OpenAI, Baileys y MongoDB potenció la capacidad del chatbot para gestionar citas, emitir recomendaciones personalizadas y mantener una trazabilidad completa del historial de atención, todo desde un entorno accesible como WhatsApp.

Durante los sprints, se logró una interacción constante entre los roles del equipo, fortaleciendo el trabajo colaborativo y asegurando que cada entrega funcional respondiera a los objetivos definidos. Las pruebas piloto validaron la funcionalidad del sistema y reflejaron una aceptación favorable por parte de los usuarios, quienes destacaron la facilidad de uso, la precisión en los recordatorios y la mejora en la gestión de sus tratamientos.

En cuanto a los resultados de las encuestas se llegó al siguiente análisis para cada una de ellas:

Pregunta 1:

Se encuestó a un total de 40 personas con el objetivo de conocer la frecuencia con la que visitan Stella Spa. Las respuestas obtenidas muestran una distribución diversa, siendo la opción más frecuente "una vez al mes", seleccionada por el 42,5% de los participantes. En segundo lugar, el 27,5% indicó que asiste con una frecuencia menor a

una vez al mes, mientras que el 22,5% señaló acudir cada dos semanas. Finalmente, solo el 7,5% manifestó visitar el spa una vez por semana.

Asimismo, se identifica un segmento de clientes frecuentes (aquellos que acuden semanal o quincenalmente), que representa el 30% de los encuestados. Este grupo podría beneficiarse de programas de fidelización, membresías o servicios personalizados que refuercen su vínculo con el spa. Por otro lado, los clientes ocasionales (27,5%) constituyen un público objetivo relevante para campañas promocionales, incentivos por visitas recurrentes o paquetes especiales que motiven un uso más continuo de los servicios.

Gráfico de la pregunta de la encuesta no.1

[8.34.1 Gráfico 1](#)

Pregunta 2:

Los resultados indican que el 56,4% de los encuestados agenda sus citas a través de WhatsApp, consolidándose como el canal de preferencia entre los clientes. Un 25,6% realiza la gestión de manera presencial, mientras que un 17,9% utiliza la llamada telefónica. Ningún encuestado seleccionó la opción "otro", lo que sugiere que los métodos disponibles actualmente cubren adecuadamente las formas de contacto que utiliza la clientela.

El hecho de que más de la mitad de los clientes prefiera agendar sus citas por WhatsApp evidencia una fuerte inclinación hacia canales digitales, por lo cual es imprescindible mantener una atención ágil, organizada y profesional en dicha plataforma. Asimismo, la presencialidad sigue teniendo un peso significativo (una cuarta parte de los encuestados), lo cual puede estar asociado a factores como hábitos adquiridos, cercanía al establecimiento o limitada familiaridad con herramientas digitales. Este grupo representa una oportunidad para promover la digitalización, facilitando el uso de plataformas tecnológicas a través de incentivos o acompañamiento.

En cuanto a las llamadas telefónicas (17,9%), aunque es el método menos utilizado

entre los tres principales, continúa siendo relevante, posiblemente por clientes que buscan un trato más directo o personalizado.

Gráfico de la pregunta de la encuesta no.2

[8.35.2 Gráfico 2](#)

Pregunta 3:

Los resultados revelan que un 15% de los encuestados indicó tener dificultades frecuentes, lo que equivale a 6 personas que experimentan problemas de manera continua al agendar sus citas. La mayoría, un 35% (14 personas), manifestó que a veces enfrenta dificultades, lo que sugiere la existencia de inconvenientes intermitentes que podrían estar asociados a variables contextuales como horarios, disponibilidad del personal o saturación del sistema.

Un 25% (10 personas) respondió que rara vez ha tenido inconvenientes, lo que implica que, si bien el problema no es frecuente para ellos, el sistema aún presenta fallas esporádicas. Finalmente, otro 25% (10 personas) expresó que nunca ha tenido problemas, lo cual evidencia que el sistema de agendamiento puede funcionar de manera eficiente en ciertos casos.

Los datos muestran que el 60% de los participantes ha experimentado algún grado de dificultad al momento de agendar una cita, lo que representa una alerta sobre la necesidad de mejoras estructurales o tecnológicas en el proceso.

Gráfico de la pregunta de la encuesta no.3

[8.36.3 Gráfico 3](#)

Pregunta 4:

Del total, el 70% (28 personas) manifestó que sí estaría interesado en agendar citas mediante un asistente virtual disponible las 24 horas del día, los 7 días de la semana. Esta alta aceptación refleja una clara disposición por parte de los usuarios a adoptar soluciones tecnológicas que les brinden flexibilidad, autonomía y facilidad de acceso al momento de gestionar sus citas.

Un 27,5% (11 personas) eligió la opción “Tal vez”, lo cual sugiere una postura

neutral o una necesidad de contar con más información antes de tomar una decisión.

Solo una persona (2,5%) respondió negativamente, indicando un nivel de resistencia mínimo, lo cual no representa una barrera significativa para la implementación de esta herramienta tecnológica.

Los resultados muestran que un 97,5% de los encuestados están abiertos a utilizar un asistente virtual, ya sea de manera afirmativa o con predisposición a considerarlo. Esta tendencia sugiere que la implementación de un sistema de agendamiento automatizado puede no solo resolver las dificultades actuales en el proceso de citas, sino también mejorar la experiencia del usuario, reducir la carga operativa del personal y garantizar disponibilidad continua.

Gráfico de la pregunta de la encuesta no.4

[8.37.4 Gráfico 4](#)

Pregunta 5:

El 82,1% de los encuestados indicó que considera muy útil recibir recordatorios automáticos de sus citas, lo cual evidencia una marcada preferencia por sistemas que faciliten la gestión del tiempo y ayuden a evitar olvidos. Por otro lado, un 15,4% manifestó que no considera necesario este tipo de recordatorios, mientras que solo el 2,5% declaró que le es indiferente.

Estos resultados reflejan una alta demanda por parte de los usuarios hacia la implementación de recordatorios automáticos, lo cual respalda su incorporación como una funcionalidad clave en herramientas tecnológicas aplicadas a la gestión de citas.

Gráfico de la pregunta de la encuesta no.5

[8.38.5 Gráfico 5](#)

Pregunta 6:

El 45% de los encuestados considera muy importante recibir recomendaciones personalizadas de tratamientos basadas en su historial, mientras que un 32,5% lo calificó como algo importante. Esto indica que más de tres cuartas partes de los participantes (77,5%) valoran significativamente la personalización del servicio en

función de su historial clínico o estético.

Por otro lado, un 22,5% considera esta funcionalidad poco importante, y ningún encuestado marcó la opción de nada importante, lo cual refuerza la percepción general positiva hacia la personalización.

Este resultado evidencia una tendencia clara hacia la preferencia por servicios individualizados, lo cual representa una oportunidad para fortalecer la relación con los clientes a través de la tecnología.

Gráfico de la pregunta de la encuesta no.6

[8.39.6 Gráfico 6](#)

Pregunta 7:

El 82,5% de los encuestados respondió que sí considera que el servicio actual en Stella Spa está suficientemente personalizado a sus necesidades. Un 17,5% indicó que podría mejorar, mientras que ningún participante seleccionó la opción "No".

Estos resultados reflejan una alta percepción positiva respecto a la personalización del servicio ofrecido por Stella Spa. Sin embargo, aunque la mayoría se muestra satisfecha, el hecho de que casi una quinta parte de los clientes sugiera que “podría mejorar” indica que existen oportunidades de optimización.

Gráfico de la pregunta de la encuesta no.7

[8.40.7 Gráfico 7](#)

Pregunta 8:

Los resultados obtenidos evidencian una clara preferencia por el uso de WhatsApp como medio de interacción, con un 85% de las respuestas. Esta abrumadora mayoría indica una inclinación significativa hacia este canal, lo que puede explicarse por su amplia penetración en el mercado, facilidad de uso, y la familiaridad de los usuarios con la plataforma.

En contraste, la página web y la aplicación móvil obtuvieron porcentajes considerablemente menores, representando aproximadamente el 7,5% cada una. Estas cifras sugieren que, aunque estos canales pueden resultar funcionales, no son

percibidos como los más cómodos o accesibles por parte de los usuarios encuestados. Finalmente, la opción "Otro" no presentó una representación significativa, lo que refuerza la idea de que los canales alternativos no son actualmente una prioridad para los usuarios.

Gráfico de la pregunta de la encuesta no.8

[8.41.8 Gráfico 8](#)

Pregunta 9:

Un 85% de los encuestados manifestó estar dispuesto a proporcionar su información personal si se asegura el cumplimiento de las disposiciones legales sobre protección de datos. Este dato refleja un alto nivel de confianza hacia los mecanismos legales vigentes y pone en evidencia que la garantía jurídica constituye un factor decisivo en la disposición de los usuarios a compartir información sensible.

Por otro lado, un 15% indicó la opción "Tal vez", lo cual sugiere la existencia de ciertos niveles de incertidumbre o escepticismo frente al cumplimiento real de la normativa o a la eficacia de las medidas de protección. Notablemente, ninguna respuesta correspondió a la opción "No", lo que refuerza aún más la percepción favorable de los encuestados hacia el tratamiento responsable de los datos personales, siempre que exista respaldo legal.

Gráfico de la pregunta de la encuesta no.9

[8.42.9 Gráfico 9](#)

Pregunta 10:

La totalidad de los 40 encuestados respondió afirmativamente, lo que representa un 100% de aprobación y disposición a recomendar el servicio.

Este resultado constituye un indicador clave de satisfacción del cliente y de fidelización. En términos de marketing relacional, la recomendación espontánea por parte de los usuarios representa uno de los niveles más altos de aprobación, pues no solo implica una experiencia positiva, sino también la voluntad de extender dicha experiencia a personas cercanas.

Gráfico de la pregunta de la encuesta no.10

[8.43.10 Gráfico 10](#)

Realizando una comparación con los datos recolectados al inicio del proceso, se realizó una comparación para evaluar el proceso anterior vs proceso con el asistente virtual.

Tabla 2.

Resumen de resultados

Problema identificado	Resultado de Encuesta	Sistema Implementado	Interpretación
50% de las citas no se confirman de forma inmediata por carga laboral de la profesional	- El 35% ha tenido dificultades al agendar - El 70% desea agendar con un asistente 24/7	Chatbot agenda citas automáticamente vía WhatsApp, disponible 24/7 Se conecta a una base de datos para disponibilidad actualizada	Mejora la eficiencia operativa, elimina tiempos muertos y evita duplicidad o pérdida de citas.
El 60% de los clientes percibe poca atención personalizada	32.5% señala la importancia de recibir recomendaciones personalizadas	El ChatBot brinda recomendaciones basadas en el historial clínico y de las terapias	Se fortalece la realcion del cliente y personaliza su experiencia
Solo el 20% sigue	82.1% considera	El chatbot envía	Mejora el

recomendaciones	recibir recordatorios	recordatorios	seguimiento de
post-tratamiento, por	automáticos	automáticos antes de	los cuidados post
falta de recordatorios		la cita	tratamiento
40% del tiempo se	El 85% prefiere usar	Gestión de citas	Optimiza tiempo
dedica a agendar,	WhatsApp como	desde WhatsApp	de atención al
coordinar y canal	para		cliente
reprogramar citas	interactuar con un		
	asistente virtua		
No se contaba con	El 82.5% cree que el	Recomendaciones	Mejora la
trazabilidad o control	servicio actual es	brindadas por el bot	personalización
automatizado de	personalizado, pero	basados en su	basada en sus
historial clínico	puede mejorar	historial	datos
No se respondían	El 70% desea	El chatbot responde	Mejora la
preguntas frecuentes	atención 24/7	preguntas generales y	disponibilidad al
o específicas fuera		específicas 24/7	cliente
del horario laboral			

Elaborado por los autores

Finalmente, con la capacitación al personal y la implementación de la plataforma en un entorno real, se consolidó una herramienta innovadora que no solo resuelve problemáticas existentes, sino que también establece las bases para futuras mejoras y escalabilidad del servicio, contribuyendo significativamente a la transformación digital del consultorio estético.

5. Conclusiones

A través de las herramientas de recolección de datos se determinaron los procesos actuales sobre la atención al cliente y gestión de citas en Stella Spa, donde se identificó falencias respecto a llevar registros manuales y poco eficientes que derivan en problemas de gestión de información.

Por lo que, se implementó la API de OpenAI a fin recopilar recomendaciones personalizadas, y usar este recurso para procesar información y ofrecer un tratamiento específico para cada cliente, lo que mejora significativamente el servicio al cliente y genera un mejor posicionamiento en el mercado.

Con esta información, se construyó una plataforma integrada a fin de automatizar la gestión de las citas de los clientes, recordatorios y seguimientos de los tratamientos en el consultorio.

Y para validarlo, se realizaron pruebas pilotos con el personal del spa y con clientes a fin de obtener resultados de eficiencia y buena ejecución de la plataforma.

6. Recomendaciones

- ✓ Se sugiere implementar en el asistente virtual, que a través del scaneo corporal realice un diagnóstico inicial del paciente basados en los datos proporcionales que indiquen dichos pacientes.
- ✓ Se recomienda impulsar una interacción de comando de voz, con la finalidad de brindar una mejor comunicación efectiva y experiencia los usuarios mediante “notas de voz”, en el asistente virtual.
- ✓ Se recomienda evaluar periódicamente la experiencia del usuario con el chatbot mediante encuestas automáticas.
- ✓ Se sugiere utilizar los datos recolectados por el chatbot para tomar decisiones estratégicas y de marketing.

7. Referencias bibliográficas

- Accenture. (2024). *Life Trends 2023: Creating experiences for a world in motion*. Accenture.
<https://www.accenture.com/content/dam/accenture/final/capabilities/song/marketing-transformation/document/Accenture-Life-Trends-2023-Full-Report.pdf>
- AceleraPyme. (2024). *La atención total a cliente y las ventajas de los asistentes virtuales / chatbots*. España.
- Acibeiro, M. (10 de Enero de 2025). *lucushost*. Lucushost:
<https://www.lucushost.com/blog/que-es-node-js/>
- Apolo. (19 de septiembre de 2024). *espanol.apolo.app*. Ninsaude:
<https://espanol.apolo.app/asistente-virtual-para-medicos-que-es-y-sus-ventajas/>
- Aswathi, K., Kalapurackal, & Jacob. (2022). Marketing de influencers y comportamiento del consumidor: una revisión sistemática de la literatura. *Visión Revista de Perspectiva Empresarial*, 28(2), 1-20. <https://doi.org/10.1177/09722629221114607>
- Banco Mundial. (14 de Abril de 2025). *Ecuador: panorama general*. Grupo Banco Mundial:
<https://www.bancomundial.org/es/country/ecuador/overview>
- Cabezas Garzón, Jenny Paola, & Castañeda Herrera, Nicolas. (2023). *IRIS : chatbot con inteligencia artificial para la gestión de procesos empresariales*.
- Contreras, R. (12 de Abril de 2023). *computing*. Computing:
<https://www.computing.es/noticias/el-nuevo-amanecer-de-los-asistentes-virtuales/>
- Datos, L. d. (26 de Mayo de 2021). *finanzaspopulares*.
https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf
- De, B. (2023). *API Management*. Apress.
https://doi.org/https://doi.org/10.1007/979-8-8688-0054-2_2
- Deyimar, A. (07 de Marzo de 2025). *hostinger*. Hostinger:
<https://www.hostinger.com/es/tutoriales/que-es-angular>
- Ehsan, A., Abuhaliqa, M., Catal, C., & Mishra, D. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences*,

- 12(9), 4369. <https://doi.org/https://doi.org/10.3390/app12094369>
- Espinoza, R. (2023). *Estrategias de Marketing Digital para impulsar los servicios de una estética y atraer clientes potenciales*. Universidad Peruana de Ciencias Aplicadas (UPC). <https://doi.org/http://hdl.handle.net/10757/671875>
- Estrada Velasco, M. V., Núñez-Villacis, Jenny Alexandra, Saltos-Chávez, Pedro Rubén , & Cunuhay-Cuchipec, Wilmer Clemente. (2021). *Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software*.
- Henao, F. (2021). *APOYO, INTEGRACION Y DESAROLLO DE BOT PARA ALGAR TECH. TECNOLÓGICO DE ANTIOQUIA- INSTITUCIÓN UNIVERSITARIA MEDELLIN*. <https://doi.org/https://dspace.tdea.edu.co/server/api/core/bitstreams/677686b8-c164-4355-bc59-40b186a8643c/content>
- IBM. (2020). *ibm*. IBM: <https://www.ibm.com/mx-es/topics/artificial-intelligence-medicine>
- IBM. (14 de Octubre de 2021). *IBM*. IBM: <https://www.ibm.com/es-es/think/topics/mongodb>
- Ingole, P. (2024). *Optimización de la asignación de recursos en hospitales mediante análisis predictivos y sistemas de información*. Revista de Ingeniería y Gestión de Sistemas de Información. <https://doi.org/10.52783/jisem.v10i1s.224>
- J. Ignacio Criado, Miquel Salvador, Edgar Alejandro Ruvalcaba Gómez, & Rodrigo Sandoval Almazan. (2021). *Inteligencia Artificial y Ética en la Gestión Pública*.
- Jara, G. (2020). *Tendencia del Mercado Cosmético Ecuatoriano*. Dirección de Inteligencia Comercial Nacional, MPCEIP. http://www.inteligenciaproductiva.gob.ec/archivos/oportunidades_de_negocio/pichincha/pichincha_01.pdf
- Khosravi, H., Reza, M., Hajiabadi, M., Shoyeb, A., & Ahmed, I. (2024). Chatbots and ChatGPT: a bibliometric analysis and systematic review of publications in Web of Science and Scopus databases. *International Journal of Data Mining, Modelling and Management*, 16(2). <https://doi.org/https://doi.org/10.1504/IJDMMM.2024.138824>
- Kodigo. (2024). *Kodigo*. Kodigo: <https://kodigo.org/para-que-se-utiliza-javascript/>
- Lamothe, M., Guéhéneuc, Y.-G., & Shang, W. (2021). A Systematic Review of API Evolution

Literature. *ACM Computing Surveys (CSUR)*, 54(8), 1-36.

<https://doi.org/https://doi.org/10.1145/3470133>

Lappalainen, Y., & Narayanan, N. (2023). Aisha: A Custom AI Library Chatbot Using the ChatGPT API. *Journal of Web Librarianship*, 17(3), 37-58.

<https://doi.org/https://doi.org/10.1080/19322909.2023.2221477>

López López, J. (2020). *Estudio de rendimiento de asistentes virtuales de voz en condiciones ruidosas*.

Magazine, I. (11 de Junio de 2024). *icrono*. Icrono:

<https://icrono.com/tendenciasmarketing/todo-lo-que-necesitas-saber-sobre-la-plataforma-de-desarrollo-de-openai/#%3A~%3Atext%3Ddesarrollo%20de%20OpenAI.-%2C%2BFQu%C3%A9%20es%20la%20plataforma%20de%20desarrollo%20de%20OpenAI%3F%2Cde%20modelos%20de%20intel>

Méndez, L. (2024). *bot-whatsapp.netlify.app*. Chatbot: <https://bot-whatsapp.netlify.app/docs/>

Montalvo, E., & Iñiguez, C. (2025). *Desarrollo del back-end y front-end en la reingeniería del Sistema de Pre-planificación de asignaturas del PAO para la FIS : desarrollo front-end*. Escuela Politécnica Nacional.

<https://doi.org/https://bibdigital.epn.edu.ec/handle/15000/26488>

Nicolescu, L., & Tudorache, M. (2022). Human-Computer Interaction in Customer Service: The Experience with AI Chatbots—A Systematic Literature Review. *Electronics*, 11(10), 1579. <https://doi.org/https://doi.org/10.3390/electronics11101579>

OMS. (2024). OMS. OMS: <https://www.who.int/es/campaigns/s-a-r-a-h>

Onibokun, T., Ejibenam, A., Chukwuemeka, P., & Damilola, K. (2023). The impact of Personalization on Customer Satisfaction. *Journal of Frontiers in Multidisciplinary Research*, 4(1), 333-341. <https://doi.org/10.54660/JFMR.2023.4.1.333-341>

Pareja Arredondo, F. C., Carrasco Taipe, J. I., & Aquino Cruz, M. (2024). *Revisión sistemática de la evolución de los asistentes virtuales: desde los primeros Chatbots hasta la IA conversacional*.

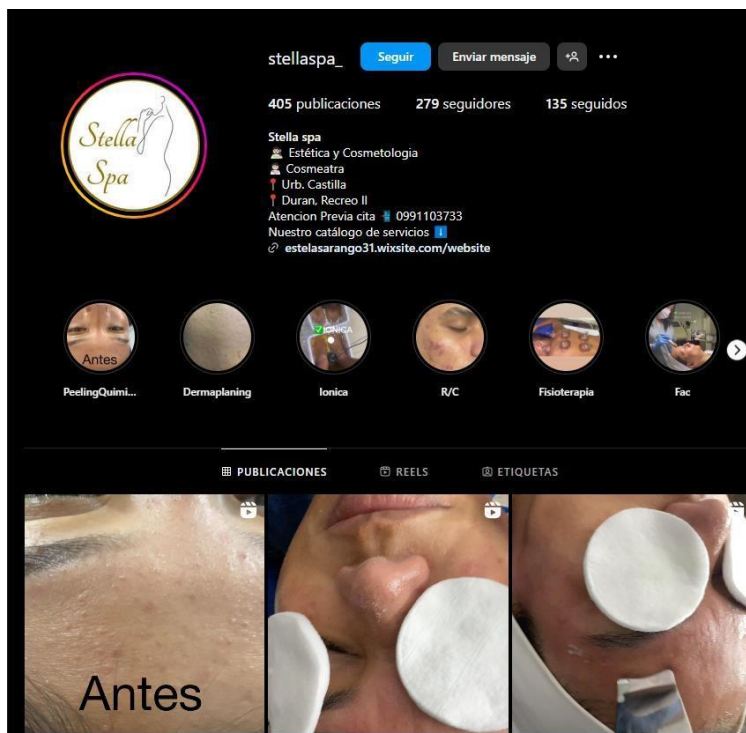
Pionce Arteaga, M. A., Caicedo Plúa, C. R., Delgado Lucas, H. B., & Murillo Quimiz, R. L.

- (2022). *Chatbots para ventas y atención al cliente*. Manabi.
- Ponce, L. (2024). *Todo lo que debes conocer sobre la Protección de Datos Personales*.
- Pujolar, G., Oliver, A., Vargas, I., & Velásquez, M. L. (2022). Cambios en el acceso a los servicios de salud durante la pandemia de COVID-19: una revisión del alcance. *International Journal of Environmental Research and Public Health*, 19(3), 1749. <https://doi.org/10.3390/ijerph19031749>
- Ramires Hernandez, P., & Valle Cruz, D. (2022). *Los Asistentes virtuales basados en Inteligencia Artificial*.
- Ramires, P., & Valle, D. (2022). Los Asistentes virtuales basados en Inteligencia Artificial. *ReCIBE, Revista electrónica De Computación, Informática, Biomédica Y Electrónica*, 11(2), C1-11. <https://doi.org/https://doi.org/10.32870/recibe.v11i2.251>
- Red Hat. (19 de Julio de 2022). *redhat*. Redhat: <https://www.redhat.com/es/topics/devops/what-is-agile-methodology>
- Rubio, J., Neira, T., Molina, D., & Vidal, C. (2022). Proyecto UBOT: asistente virtual para entornos virtuales de aprendizaje. *Información tecnológica*, 33(4), 85-92. <https://doi.org/http://dx.doi.org/10.4067/S0718-07642022000400085>
- Sanabria Rodríguez, E. E., & Frade Barreto, W. J. (2023). *APORTES DE LA INTELIGENCIA ARTIFICIAL AL ÁREA DE SERVICIO AL CLIENTE*.
- Sanchez, C. (2022). *botmaker*. Botmaker: <https://botmaker.com/es/posteos/inteligencia-artificial-atencion-cliente>
- SW Team. (29 de Mayo de 2024). *swhosting*. Swhosting: <https://www.swhosting.com/es/blog/principales-lenguajes-de-programacion-web>
- Villanueva, C., Tornero, J., & Rubio, A. C. (2024). Influencias culturales en las decisiones de Medicina Estética: el rol de la psicología. *Medicina Estética*, 10(4), 1-24. <https://doi.org/10.57662/am.v10i4.15534>
- Vorecol, E. d. (2024). *blogs-es.vorecol.com*. Vorecol: <https://vorecol.com/es/articulos/articulo-la-integracion-de-chatbots-y-asistentes-virtuales-en-plataformas-de-engagement-beneficios-y-desafios-172611#6.-casos-de-%C3>

%A9xito-en-la-implementaci%C3%B3n-de-chatbots-en-empresas

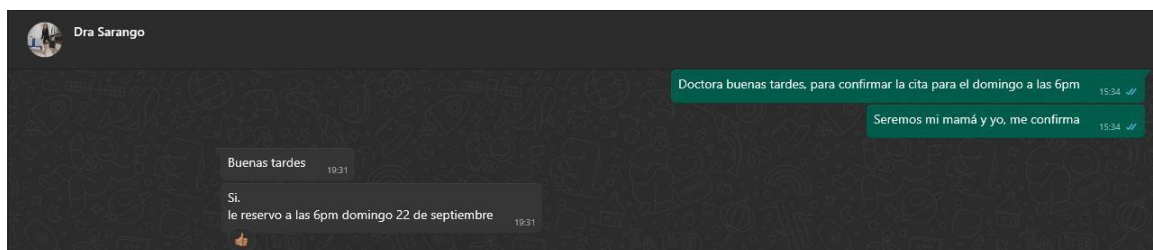
8. Anexos

Anexo A. Página Oficial del Instagram del SPA y su funcionamiento



Nota, El consultorio cosmético Stella Spa muestra las terapias en su plataforma de Instagram, donde comparte diferentes etapas de las sesiones y como se realizan. Además, ofrece un numero de contacto de WhatsApp para obtener para información sobre algún tratamiento

Anexo B. Captura de pantalla de chat Whatsapp con la Dra. Estela Sarango



Nota, Se procedió con el contacto de la cosmetóloga para separar una cita y observar cómo es el servicio al cliente.

Anexo C. Fotografía de tratamiento

Nota, todas las sesiones se llevan a cabo en 1 hora o 2 horas según el tratamiento de acné.

Anexo D. Fotografía de tratamiento de limpieza facial

Nota, todas las sesiones se llevan a cabo en 1 hora o 2 horas según el tratamiento facial

Anexo E. Fotografía de tratamiento Desintoxicación IONICA

Nota, todas las sesiones se llevan a cabo en 1 hora o 2 horas según el tratamiento de desintoxicación iónica

Como cliente se evidencio la demora por parte de la asignación de turno, el diagnostico brindado fue completo, pero para empezar el tratamiento se perdió algo de tiempo en decidir como empezar provocando acortar el periodo de la sesión, que según el tratamiento puede ser negativo. Posteriormente, se brindaron recomendaciones y se estableció en los tiempos para las siguientes sesiones del tratamiento.

Anexo F. Reuniones

El día 14 de noviembre del presente año se realizó una reunión para abordar temas con respecto a un asistente virtual por medio de una videollamada.

Participantes de la reunión

Beneficiaria: Doctora Estela Sarango. Dueña de Stella Spa

Entrevistador: Marcell Moreno Salcedo, Tommy Ramírez. Estudiante de la carrera de Ingeniería

Durante la primera reunión se trataron los siguientes temas:

- Proceso actual del contacto con el cliente
- Agendamiento de las citas
- Seguimiento de los tratamientos

En la primera sesión, se habló de cómo se llevaba a cabo el inicio del servicio desde el contacto con el cliente, el diagnóstico, las sesiones de tratamiento, las sugerencias y recomendaciones. Posteriormente de programar la nueva sesión para las próximas semanas o días, según cual sea la recomendación.

La cosmetóloga destacó que por cuestiones de tiempo no siempre puede agendar las citas de forma inmediata para los pacientes, tampoco darle recomendaciones más seguidas, en ocasiones debe preparar los instrumentos después de haber hablado con el paciente de lo que se quiere realizar provocando que la sesión dure menos del tiempo requerido por conflictos de horarios, los pacientes no cumplen con el tiempo programado para las siguientes terapias.

Menciono que le parece mejor un sistema en donde el cliente realice por su propia cuenta la reservación de la cita. Se tomaron apuntes de esos requerimientos para brindar las soluciones respectivas y dispuestas a ser discutidas en la próxima sesión.

Esto evidencia la dependencia en la terapeuta para gestionar cada solicitud, lo cual resulta en una carga adicional y aumenta el riesgo de errores en la programación. Este método subraya la necesidad de un sistema automatizado que facilite el proceso y reduzca la carga operativa para la terapeuta.

El día 17 de noviembre se llevó a cabo la segunda reunión

Participantes de la reunión

Beneficiaria: Doctora Estela Sarango. Dueña de Stella Spa

Entrevistador: Marcell Moreno Salcedo, Tommy Ramírez. Estudiante de la carrera de Ingeniería

Donde se trataron los siguientes temas:

- Optimización del agendamiento de citas
- Asistente virtual (chatbot)

Se realizó la reunión con la terapeuta para brindar las soluciones con respecto al sistema de agendamiento de citas. En esta sesión, se abordó con mayor profundidad como es el flujo actual del agendamiento, se identificaron los problemas de disponibilidad y la gestión manual de citas. Por medio de un sistema de agendamiento que lo realiza directamente el cliente. Y de una plataforma web en donde podrá observar las citas registradas, junto con información del paciente, terapias hechas y su historial.

Anexo G. *Captura de reunión con la terapeuta para brindar las soluciones con respecto al sistema de agendamiento de citas*



Nota, se discutió la implementación de un Asistente Virtual que permitiría optimizar el proceso de agendamiento y ofrecer un seguimiento automatizado de las terapias.

Anexo H. *Entrevista para conocer la experiencia actual del cliente con respecto al servicio de Stella Spa*



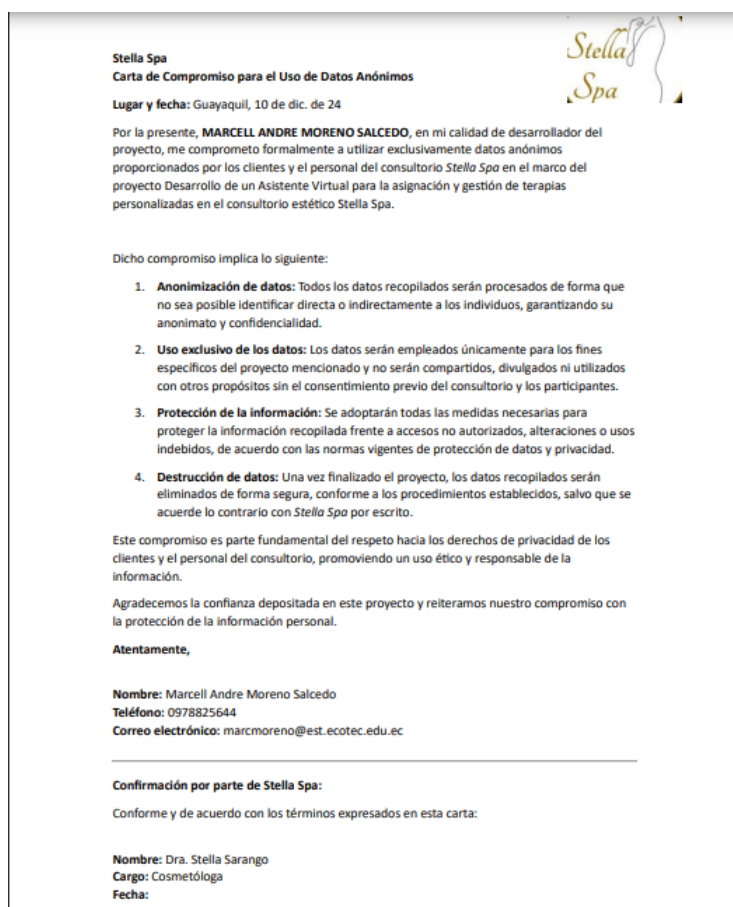
Nota, se aplicaron las siguientes preguntas.

- ✓ ¿Cómo describiría su experiencia general como cliente al momento de contactar al consultorio para agendar una cita?
- ✓ ¿Le resulta claro y fácil obtener información sobre las sesiones disponibles y las recomendaciones que podrían ajustarse a sus necesidades?
- ✓ ¿Cree que las recomendaciones de tratamientos están adaptadas a sus necesidades, o siente que falta personalización en el servicio?
- ✓ Si pudiera sugerir alguna mejora en el servicio al cliente del consultorio. ¿Qué recomendarías?

Entrevista: <https://youtu.be/GoL8ekUPuAk>

En donde se obtuvieron las siguientes conclusiones. El análisis de las respuestas revela que los principales puntos de mejora en el servicio al cliente de Stella Spa incluyen la reducción del tiempo de espera para agendar citas, la necesidad de una comunicación más clara sobre los servicios disponibles y la falta de personalización en las recomendaciones de tratamientos. La implementación de un asistente virtual automatizado aborda estas deficiencias al ofrecer respuestas inmediatas, gestionar citas de manera eficiente y generar recomendaciones personalizadas basadas en las características y el historial de los clientes. Esto no solo optimiza la experiencia del usuario, sino que también libera a la cosmetóloga de tareas administrativas, permitiéndole enfocarse en los tratamientos. Así, el asistente virtual se presenta como una solución clave para modernizar el consultorio, mejorar la satisfacción del cliente y aumentar la eficiencia operativa.

Anexo I. Uso de tratamientos de datos, se solicitaron los permisos para el tratamiento de los datos



Anexo J. Estructura para el desarrollo del proyecto

El asistente virtual para Stella Spa está diseñado como un sistema modular e integrado que combina las capacidades de OpenAI, para garantizar una interacción fluida con los usuarios y una gestión eficiente de las tareas. Su arquitectura está compuesta por varios componentes clave que trabajan en conjunto:

✓ Interfaz de Usuario: Conexión a través de WhatsApp

El punto de entrada para los usuarios es WhatsApp, una plataforma ampliamente utilizada y accesible para clientes de todas las edades. Esta elección asegura que los usuarios puedan interactuar con el asistente virtual sin necesidad de descargar aplicaciones adicionales.

El cliente envía un mensaje a través de WhatsApp y la información se transmite automáticamente a, que actúa como intermediario entre el cliente y los sistemas subyacente

✓ Automatización de Procesos: Manejo de flujos con Baileys WhatsApp

Una vez que el cliente decide realizar una acción, como agendar una cita o pedir información sobre los servicios, entrara en los flujos respectivos para su interacción con el bot.

✓ Reservación de Citas:

Por medio de la interfaz de WhatsApp este accede al flujo de cita, en donde insertara los datos en la base para realizar la reservación, validación y retorno de confirmación de la reserva. Esta información será visualizada en por parte del cliente y el usuario admin, en le medio de comunicación empleado.

Benchmarking

Con los ejemplos investigados, se evaluaron ciertos puntos para caso desde el NLP, automatización, personalización, accesibilidad, cumplimiento legal y la facilidad de eso, en comparación con el asistente que se desarrollara en Stella Spa.

Anexo K. Tabla comparativa de empresas que emplean IA

Criterio	SARAH (OMS)	Blue (BBVA)	Dom (Domino's)	Asistente de Stella Spa
NLP	Avanzado	Básico	Limitado	Intermedio-avanzado (OpenAI)
Automatización	Limitada	Alta (consultas)	Alta (pedidos)	Alta (citas, seguimientos)
Personalización	Básica	Nula	Aprendizaje simple	Alta (recomendaciones y terapias)
Accesibilidad	Web, móvil	Web, móvil	Web, móvil	WhatsApp, web
Cumplimiento	GDPR	GDPR	N/A	Ley de Protección de Datos (Ecuador)
Facilidad de uso	Alta (usuario)	Media	Alta	Alta (intuitivo y flexible)

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

El asistente virtual de Stella Spa combina lo mejor de los casos estudiados, destacándose por su automatización integral al gestionar citas, responder preguntas y realizar seguimientos automatizados, así como por su personalización avanzada gracias a OpenAI, que adapta recomendaciones y servicios según el historial del

cliente.

Además, integra WhatsApp como canal principal, garantizando accesibilidad y simplicidad, mientras alinea su desarrollo con las normativas locales de protección de datos para ofrecer seguridad y confianza. Esta solución no solo se posiciona como innovadora en el sector estético, sino que también demuestra un enfoque estratégico al aprovechar lo mejor de la tecnología existente, adaptándola a un contexto específico y con amplias oportunidades de escalabilidad futura.

Anexo L. Cronograma de actividades

☐	Tipo	Clave	Resumen	Estado	Comentarios	Sprint	Persona asignada
☐	▼ 	SCRUM-1	Servicio Bailyse	EN PROGRESO	 Añadir comentario		
☐		SCRUM-2	Instalar dependencias	HECHO	 Añadir comentario		 Tommy Ramirez
☐		SCRUM-3	Revisar Documentacion y curso de la herramienta	HECHO	 Añadir comentario		 Andre Moreno
☐		SCRUM-4	Conexion a base MONGODB	HECHO	 Añadir comentario		 Andre Moreno
☐		SCRUM-5	Primeras consultas y respuestas del bot	HECHO	 Añadir comentario		 Tommy Ramirez
☐		SCRUM-6	Problemas con la interaccion del bot	HECHO	 Añadir comentario		 Tommy Ramirez
☐		SCRUM-7	Almacenamiento de los datos en base	HECHO	 Añadir comentario		 Andre Moreno
☐		SCRUM-14	Nuevas funciones para la descripcion de los servicios	HECHO	 Añadir comentario		 Andre Moreno
☐	▼ 	SCRUM-8	MongoDB	POR REALIZAR	 Añadir comentario		
☐		SCRUM-9	Creacion de la base	HECHO	 Añadir comentario		 Tommy Ramirez
☐		SCRUM-10	Creacion de las colecciones y los campos	HECHO	 Añadir comentario		
☐		SCRUM-11	Revison de los campos almacenados	HECHO	 Añadir comentario		 Andre Moreno

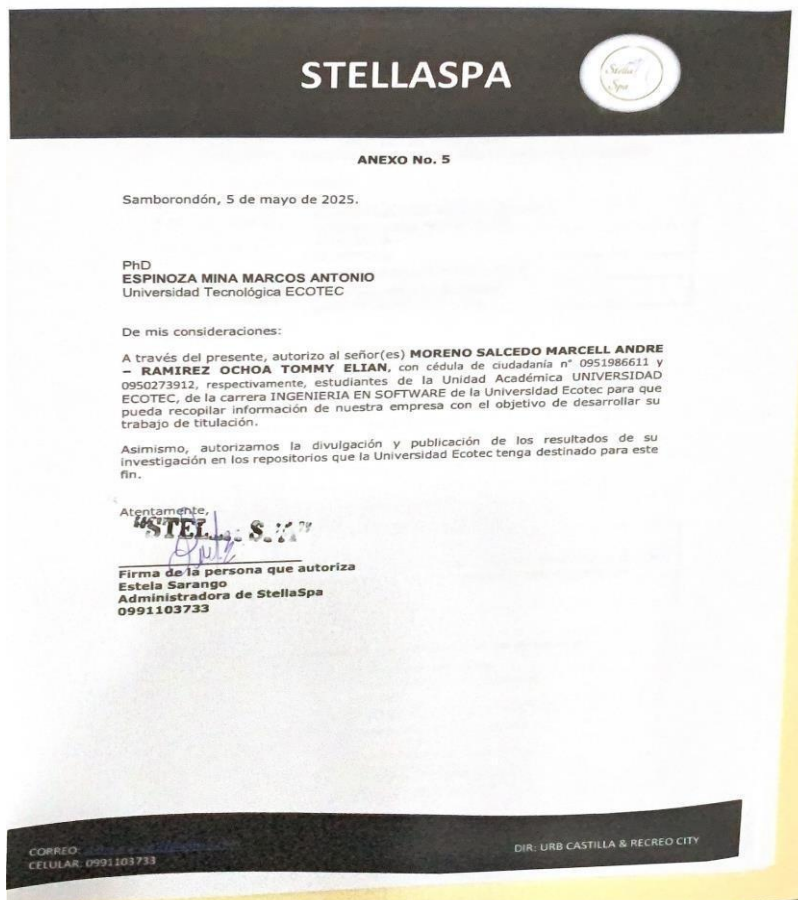
Nota, Elaboración propia, (Moreno & Ramírez, 2025).

Anexo M. Reunión sobre avance del proyecto



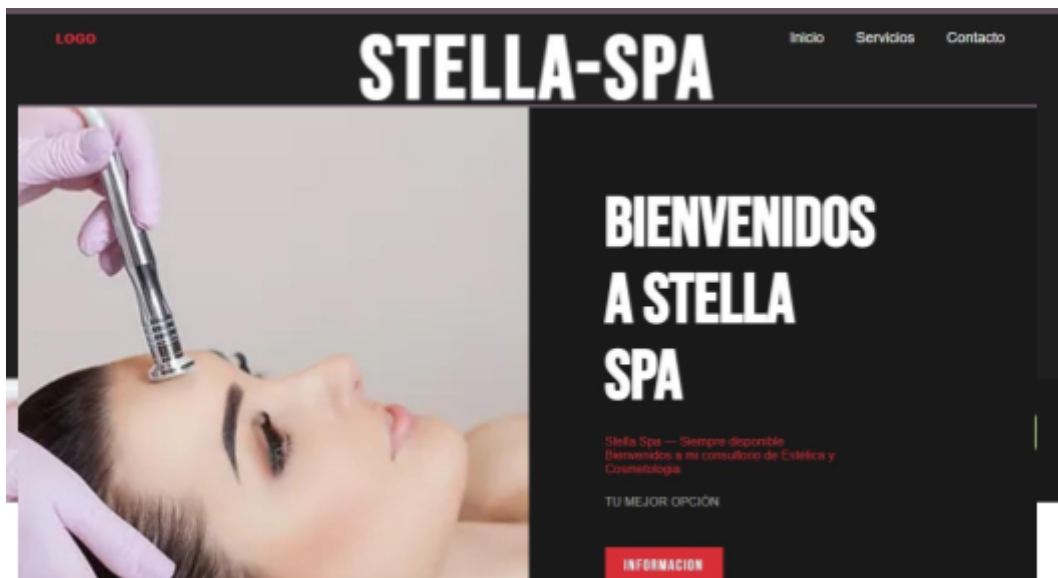
Nota, Elaboración propia, (Moreno & Ramírez, 2025).

Anexo N. Gestión de autorización de la empresa



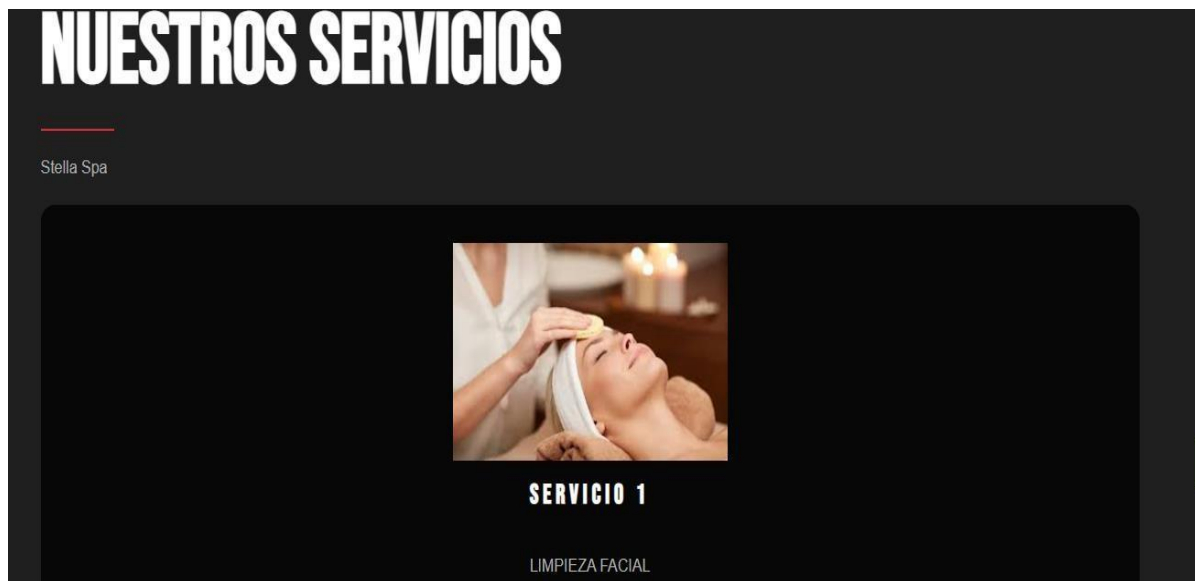
Nota, Se realizó la gestión para tener los permisos necesarios

Anexo O. Captura de pantalla de página web



Nota, demostración de página web

Anexo P. Integración de los servicios de Stella Spa



Nota, servicios de Stella Spa

Anexo Q. Código de página web HTML

```

EXPLORER
v OPEN EDITORS
  x index.html C:\Users\ramir\OneDrive\...
  # style.css C:\Users\ramir\OneDrive\...
v NO FOLDER OPENED
v OUTLINE
  v html
    v head
      meta
      meta
      title
      link
    v body
      v header.header
        v div.menu.container
          a.logo
          input#menu
          label
            img.menu-icono
          nav.navspa
            ul
              li
                a
              li
            v div.header-content.container
              div.header-txt
                h1
  v TIMELINE

index.html x # style.css
C:\Users\ramir\OneDrive\Documentos\STELLA SPA > index.html > html > head
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8"> <!-- se define el conjunto de caracteres como UTF-8 -->
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Permite que la página sea respon
6   <title>Stella Spa</title> <!-- Título de la pestaña del navegador -->
7
8   <link rel="stylesheet" href="style.css"> <!-- Enlace al archivo de estilos CSS -->
9 </head>
10 <body>
11
12 <!-- ENCABEZADO PRINCIPAL -->
13 <header class="header">
14
15   <div class="menu container"> <!-- Contenedor del menú de navegación -->
16
17     <a href="#" class="logo">Logo</a> <!-- Logo o nombre del sitio -->
18     <input type="checkbox" id="menu"> <!-- Checkbox oculto para activar el menú en móviles -->
19     <label for="menu"> <!-- Etiqueta que activa el menú cuando se hace clic -->
20        <!-- Icono del menú -->
21     </label>
22
23     <!-- Navegación del sitio -->
24     <nav class="navspa">
25       <ul>
26         <li> <a href="#">Inicio</a></li> <!-- Enlace a inicio -->
27         <li> <a href="#">Servicios</a></li> <!-- Enlace a servicios -->
28         <li> <a href="#">Contacto</a></li> <!-- Enlace a contacto -->
29       </ul>
30     </nav>
31   </div>
32

```

Nota, Código HTML responsivo para la página principal de "Stella Spa" con menú de navegación adaptable y estilo externo

Anexo R. Código de proyecto

```

1  <html lang="es">
2  <body>
3  <header class="header">
4
5      <!-- Contenido del encabezado (frase y datos) -->
6      <div class="header-content container">
7
8          <div class="header-txt">
9              <h1>STELLA-SPA</h1> <!-- Título principal -->
10             <p> Bienvenidos a mi consultorio de Estética y Cosmetología </p> <!-- Texto introductorio -->
11             <a href="#" class="btn-1">Información</a> <!-- Botón -->
12          </div>
13
14          <!-- Información rápida (dirección, teléfono y horario) -->
15          <div class="header-dir">
16              <div class="dir">
17                  <h3>Dirección</h3>
18                  <p>Urb. Castilla & Recreo city</p>
19              </div>
20
21              <div class="dir">
22                  <h3>Teléfono</h3>
23                  <p>+593978793148</p>
24              </div>
25
26              <div class="dir">
27                  <h3>Horarios</h3>
28                  <p>08:30 - 19:00</p>
29              </div>
30          </div>
31      </div>
32  </header>
33
34  <!-- SECCIÓN DE BIENVENIDA -->
  
```

Nota, Este segmento HTML del encabezado de "Stella Spa" presenta el título, mensaje de bienvenida, botón informativo y datos clave como dirección, contacto y horario.

Anexo S. Código del CSS

```

1  /* Importación de fuentes desde Google Fonts */
2  @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700;800&display=swap');
3  @import url('https://fonts.googleapis.com/css2?family=Bebas+Neue&display=swap');
4
5  /* Fuente alternativa de título (comentada para referencia) */
6  /* font-family: 'Bebas Neue', cursive; */
7
8  /* Reset básico: elimina márgenes, paddings y estilos por defecto */
9  * {
10     margin: 0;
11     padding: 0;
12     box-sizing: border-box;
13     text-decoration: none;
14     list-style: none;
15 }
16
17 /* Estilo general del cuerpo */
18 body {
19     font-family: 'Poppins', sans-serif; /* ¡OJO! Tiene un error: debería ser 'Poppins' */
20     background-color: #1f1f1f; /* Color de fondo oscuro */
21 }
22
23 /* Contenedor con ancho máximo centrado */
24 .container {
25     max-width: 1200px;
26     margin: 0 auto;
27 }
  
```

Nota, Este CSS define estilos base del sitio, importa fuentes de Google y aplica un reinicio

general para asegurar coherencia visual entre navegadores.

Anexo T. Código de configuración del ChatBot

```
base-baileys-memory > app.js > flowCita
1 // Importar las dependencias necesarias
2 const { createBot, createProvider, createFlow, addKeyword, EVENTS } = require('@bot-whatsapp/bot');
3 require('dotenv').config;
4 const QRPortalWeb = require('@bot-whatsapp/portal');
5 const BaileysProvider = require('@bot-whatsapp/provider/baileys');
6 const chrono = require('chrono-node');
7 //base
8 const MongoAdapter = require('@bot-whatsapp/database/mongo');
9 //para lectura
10 const path = require('path');
11 const fs = require('fs');
12 //openAI
13 const chat = require('./chatGPT');
14 const { set } = require('mongoose');
15
16 //ARCHIVOS DE TEXTO PARA LOS MENUS
17 const menuPath = path.join(__dirname, 'mensajes', 'menu.txt');
18 const menu = fs.readFileSync(menuPath, 'utf-8');
19 //RESPUESTAS DEL BOT
20 const consultaPath = path.join(__dirname, 'mensajes', 'promptConsultas.txt');
21 const resBot = fs.readFileSync(consultaPath, 'utf8');
22 //base
23 let adapterDB;
24 // Inicializar la conexión con MongoDB
25 const conectarMongoDB = async () => {
26   adapterDB = new MongoAdapter({
27     dbUri: 'mongodb://127.0.0.1:27017/',
28     dbName: 'usuarios'
29   });
30   console.log('✅ Conexión con MongoDB establecida correctamente.');
```

Nota, Este fragmento de código importa las librerías necesarias y declara constantes para configurar la conexión con Baileys, MongoDB, prompts y la API de OpenAI.

Anexo U. Flujos de entrada del Chatbot

```
const flowBienvenida = addKeyword(EVENTS.WELCOME)
  .addAnswer("Hola, soy *TheBot* 🤖 Asistente Virtual de Stella Spa")
  .addAnswer("Antes de continuar, ¿me indicas tu nombre y apellido?", { capture: true }, async (ctx, { gotoFlow, flowDynamic }) => {
    const nombreUsuario = ctx.body.trim();
    const regexNombreValido = /^[A-Za-zÁÉÍÓÑáéíóñ\s-]{2,}$/;

    if (!regexNombreValido.test(nombreUsuario)) {
      await flowDynamic("❌ Nombre inválido. Usa solo letras y espacios. Ej: Juan Pérez.");
      // Repetimos la misma pregunta SIN salir del flujo
      return;
    }

    const telefono = ctx.from;
    const pacientesCollection = adapterDB.db.collection('pacientes');
    const pacienteInfo = adapterDB.db.collection('historialclinforbas');

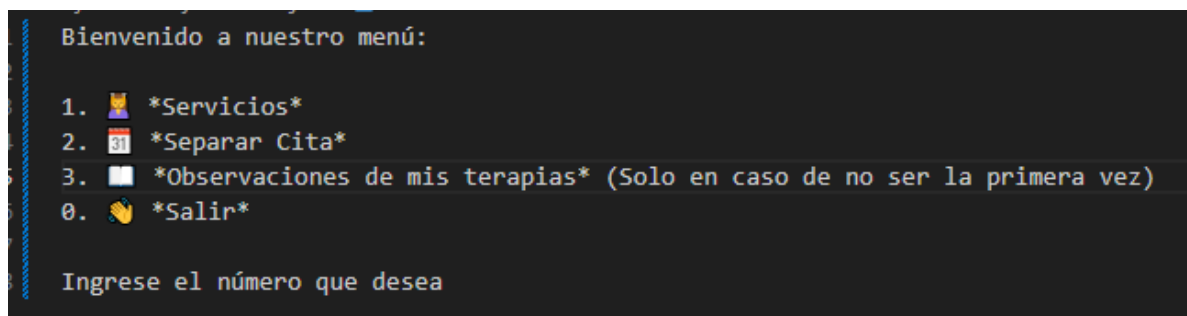
    try {
      const pacienteExistente = await pacientesCollection.findOne({ telefono });
      if (!pacienteExistente) {
        // Si no existe, lo inserta
        await pacientesCollection.insertOne({
          nombre: nombreUsuario, telefono
        });
        await pacienteInfo.insertOne({ ...
        });
      } else {
        console.log('👤 Paciente ya registrado');
      }
    } catch (error) {
      console.error('❌ Error al guardar paciente:', error);
    }

    await flowDynamic(`¡Bienvenido, *${nombreUsuario}*! 😊`);
    return gotoFlow(menuFlow);
  });
```



Nota, El flujo de bienvenida se activa con el primer mensaje del usuario, solicita su nombre y, si es nuevo, lo registra en la colección PACIENTES, creando además su perfil en HISTORIALCLINFOBAS; al finalizar, lo redirige al flujo menuFlow.

Anexo V. Flujo de menú

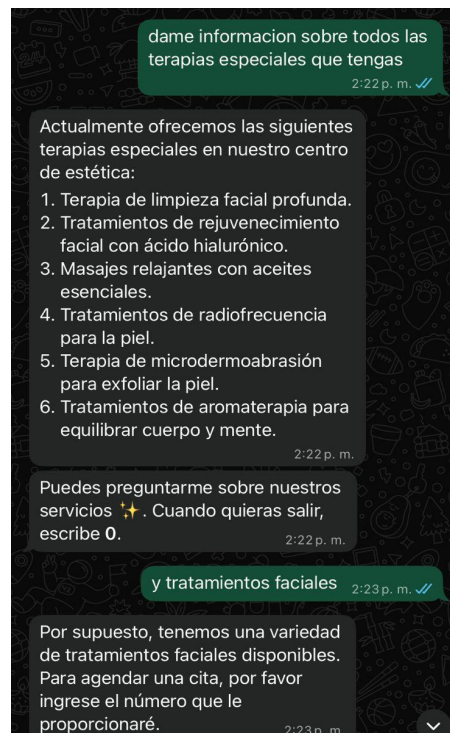


Nota, se presentará las opciones con las que cuenta el asistente virtual. Esta lista se

obtiene de un prompt que es llenado en un archivo .txt. El cual es cargado y mostrado al usuario en la interfaz de WhatsApp, el usuario debe ingresar el número correspondiente a la opción a la que quiere ingresar para ser dirigido al flujo correspondiente.

Anexo W. Flujo de servicios

```
const flowServicios = addKeyword(EVENTS.ACTION)
.addAnswer("Puedes preguntarme sobre nuestros servicios ✨. Cuando quieras salir, escribe *0*.", {
  capture: true
}),
async (ctx, ctxFn) => {
  const consulta = ctx.body.trim();
  if (consulta === "0") {
    await ctxFn.flowDynamic("¡Perfecto! Gracias por tu tiempo. Si necesitas algo más, no dudes en escribirme 😊");
    return ctxFn.gotoFlow(menuFlow);
  }
  const palabrasCita = ["agendar", "cita", "separar", "reservar", "Agendar"];
  const quiereCita = palabrasCita.some(p => consulta.includes(p));
  if (quiereCita) {
    await ctxFn.flowDynamic("¡Perfecto! Te ayudaré a agendar tu cita 📅");
    return ctxFn.gotoFlow(flowCita);
  }
  const serviciosCollection = adapterDB.db.collection('servicios');
  const servicios = await serviciosCollection.find().toArray();
  const promptServicios = servicios
    .map((s, i) => `${i + 1}. ${s.nombre}: ${s.contenido}`)
    .join("\n");
  const prompt = `
  Eres un asistente de estética.
  Usa palabras para ambos generos femenino/masculino
  le dices que ingrese la palabra agendar/cita/reservar si quiere una cita
  No pidas metodos de pagos
  No le pidas numero para agendar cita
  Usa solo la siguiente información para responder preguntas de clientes de forma clara y breve:
  ${promptServicios}
  Consulta del usuario: "${consulta}"
  `;
  const respuesta = await chat(prompt, consulta);
  await ctxFn.flowDynamic(respuesta.content);
  return ctxFn.gotoFlow(flowServicios);
});
```



Nota, El flujo de servicios permite al usuario interactuar con el ChatBot mediante la API de OpenAI, que responde en lenguaje natural usando datos de la colección SERVICIOS; además, reconoce palabras clave como "agendar" o "cita" para redirigir automáticamente al

flujo de agendamiento sin volver al menú principal

Anexo X. Flujo de cita

```
// Variables globales para almacenar los datos
let globalHora = "";
let fecha = "";

// flujo para agendar citas
const flowCita = addKeyword(EVENTS.ACTION)
.addAnswer('*Importante*: \n 📌 - Para agendar tu cita, utilice su propio dispositivo móvil.')
.addAnswer('*¿Cuál es la fecha para tu cita? 📅 \n (Formatos: ej. - DD/MM, - 01-05-2025)', { capture: true }, { async (ctx, ctxFn) => {
const entradaFecha = ctx.body;
const globalFecha = chrono.es.parseDate(entradaFecha);

if (!globalFecha) {
await ctxFn.flowDynamic("❌ No entendí la fecha. Por favor ingresa una fecha válida como 'hoy', 'mañana', '28 de este mes' o '2025-07-01'.");
return await ctxFn.gotoFlow(flowCita);
}

const zonaOffsetHoras = -5;
const fechaLocal = new Date(globalFecha.getTime() + zonaOffsetHoras * 60 * 60 * 1000);
const fechaHoy = new Date();
fechaHoy.setHours(0, 0, 0, 0);
fechaLocal.setHours(0, 0, 0, 0);

if (fechaLocal < fechaHoy || fechaLocal.getFullYear() !== fechaHoy.getFullYear()) {
await ctxFn.flowDynamic("🚫 Solo puedes agendar una cita desde hoy en adelante y dentro del año actual. Intentalo nuevamente.");
return await ctxFn.gotoFlow(flowCita);
}else{
// Guardar en variable global si es válida
fecha = entradaFecha;
await ctxFn.gotoFlow(flowHora);
}
})
})
```

Nota, Este flujo permite al usuario agendar una cita solicitando primero la fecha en formato válido (DD/MM o DD-MM-AAAA), validando que no sea una fecha pasada ni de otro año; además, se declaran las variables globales de hora y fecha para usarlas en flujos posteriores

Anexo Y. Flujo de hora

```
//flujo de la hora
const flowHora = addKeyword(EVENTS.ACTION)
.addAnswer('*¿A qué hora te gustaría agendarla? \n Formato: ej. - 4pm o 16:00', { capture: true }, { async (ctx, ctxFn) => {
globalHora = ctx.body;
const telefonoUsuario = ctx.from;

// fecha almacenada del flujo anterior
const globalFecha = chrono.es.parseDate(fecha);
const zonaOffsetHoras = -5;
const fechaLocal = new Date(globalFecha.getTime() + zonaOffsetHoras * 60 * 60 * 1000);
fecha_separada = fechaLocal.toISOString().split('T')[0];

const fechaDetectada = chrono.es.parseDate(globalHora);
if (!fechaDetectada) {
await ctxFn.flowDynamic("❌ No entendí la hora. Usa un formato válido como '14:00', '4pm', '8:30am'.");
return await ctxFn.gotoFlow(flowCita);
}

const horas = fechaDetectada.getHours().toString().padStart(2, '0');
const minutos = fechaDetectada.getMinutes().toString().padStart(2, '0');
const horaFormateada = `${(horas):${(minutos):}`;
globalHora = horaFormateada;

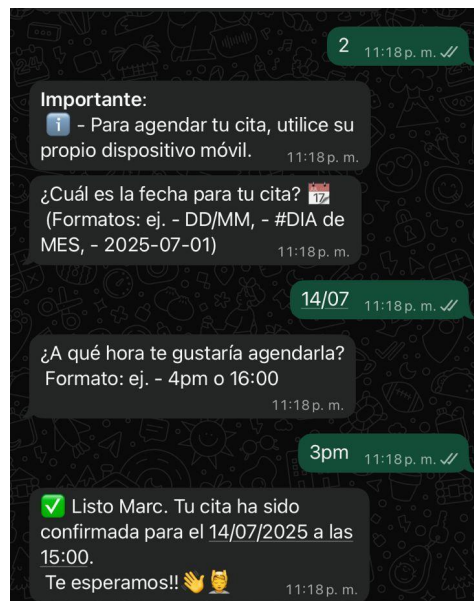
if (fechaDetectada.getHours() < 8 || fechaDetectada.getHours() >= 19) {
await ctxFn.flowDynamic("🕒 Solo se permiten citas entre las 08:00 y las 19:00. Intenta otra hora.");
return await ctxFn.gotoFlow(flowCita);
}

//colecciones
const pacientesCollection = adapterDB.db.collection("pacientes");
const agendasCollection = adapterDB.db.collection("agendas");

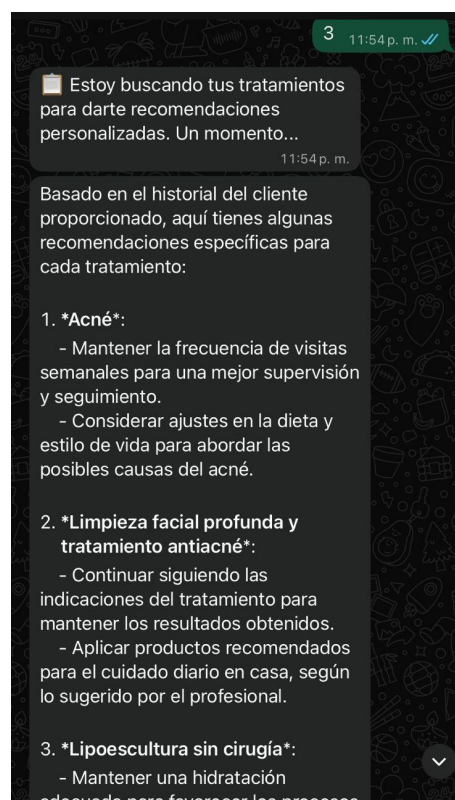
try {
const paciente = await pacientesCollection.findOne({ telefono: telefonoUsuario });
if (!paciente) {
await ctxFn.flowDynamic("❌ No se encontró tu registro. Por favor vuelve al inicio y proporciona tu nombre.");
return;
}
const citaExistente = await agendasCollection.findOne({
fecha: fecha_separada,
hora: horaFormateada
});

if (citaExistente) {
await ctxFn.flowDynamic("🕒 Ya hay una cita registrada para el ${fecha_separada} a las ${globalHora}. Por favor elige otra hora o cambia la fecha.");
return ctxFn.gotoFlow(flowCita);
}

const yaTieneCita = await agendasCollection.findOne({
telefono: telefonoUsuario,
fecha: fecha_separada,
hora: horaFormateada
});
});
```



Anexo Z. Flujo de recomendaciones



```
// Flujo de recomendaciones
const flowRecomendaciones = addKeyword(EVENTS.ACTION)
.addAnswer("👋 Estoy buscando tus tratamientos para darte recomendaciones personalizadas. Un momento...", {}, async (ctx, ctxFn) => {
  const telefono = ctx.from;
  const tratamientoCollection = adapterDB.db.collection('tratamientos');
  const tratamientosUsuario = await tratamientoCollection.find({ telefono }).toArray();
  if (!tratamientosUsuario.length) {
    await ctxFn.flowDynamic("No encontré tratamientos registrados para tu número (${telefono}). Si crees que es un error, por favor contáctanos directamente 📞.");
    return ctxFn.gotoFlow(menuFlow);
  }
  // Limitar a los 3 primeros tratamientos
  const tratamientosLimitados = tratamientosUsuario
    .map((t, i) => { i + 1, tratamiento: t.tratamiento })
    .filter((t, i) => i < 3)
    .map(t => ({
      Observaciones: t.Observaciones,
      Evaluación: t.Evaluación
    }));
  const prompt = `
  Actúa como un experto que analiza texto y da recomendaciones puntuales.
  - Lee el siguiente texto.
  - No lo resumas completo.
  - Devuelve solo entre 2 y 5 recomendaciones útiles, claras y accionables.
  - Sé directo, evita repetir lo que ya dice el texto.
  - Las recomendaciones deben ser prácticas y fáciles de entender.
  - Relacionadas con los tratamientos del cliente.
  - Como adicional recomienda futuras terapias que debería realizar el cliente.
  Historial del cliente (teléfono: ${telefono}):
  ${tratamientosLimitados}
  `;
  const respuesta = await chat(prompt, telefono);
  if (!respuesta?.content) {
    await ctxFn.flowDynamic("No pude generar recomendaciones en este momento. Por favor intenta más tarde 🙏.");
    return ctxFn.gotoFlow(menuFlow);
  }
  await ctxFn.flowDynamic(respuesta.content);
  return ctxFn.gotoFlow(menuFlow);
});
```

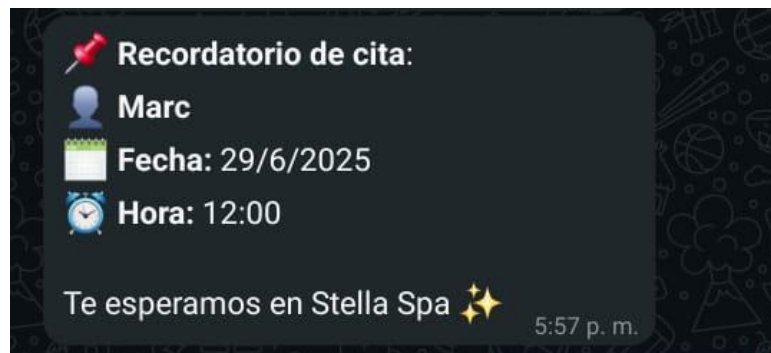
Nota, Este flujo analiza automáticamente los datos de la colección TRATAMIENTO mediante IA, genera recomendaciones personalizadas usando un prompt, y si el usuario es nuevo o no tiene sesiones registradas, informa que no hay recomendaciones; luego envía la respuesta y retorna al menú.

Anexo AA. Función de recordatorios

```
//recordatorio
const enviarRecordatorio = async () => {
  if (!sock) {
    console.warn("⚠️ No se puede enviar recordatorios: sock no está listo.");
    return;
  }
  const agendasCollection = adapterDB.db.collection("agendas");
  const pacientesCollection = adapterDB.db.collection("pacientes");
  const hoy = new Date();
  hoy.setHours(0, 0, 0, 0);
  const manana = new Date(hoy);
  manana.setDate(manana.getDate() + 1);

  const citas = await agendasCollection.find({
    fecha: {
      $gte: hoy,
      $lt: manana
    },
    status: "pendiente"
  }).toArray();

  for (const cita of citas) {
    const paciente = await pacientesCollection.findOne({ telefono: cita.telefono });
    if (!paciente) continue;
    const mensaje = `🌟 *Recordatorio de cita:*
    📞 *${paciente.nombre}*
    📅 *Fecha:* ${cita.fecha.toLocaleDateString("es-EC", { timeZone: "UTC" })}
    🕒 *Hora:* ${cita.hora}
    Te esperamos en Stella Spa 🌟`;
    const numeroWhatsApp = `${cita.telefono.replace(/^0/, "593")}@c.us`;
    try {
      await sock.sendMessage(numeroWhatsApp, { text: mensaje });
      console.log("✅ Recordatorio enviado a", paciente.nombre);
    } catch (err) {
      console.error("❌ Error al enviar recordatorio a", paciente.nombre, err.message);
    }
  }
}
```



Notas, La función sock permite comunicarse con un numero en específico para el envío del recordatorio al usuario por medio de un mensaje a WhatsApp. La función accede a la colección de AGENDAS Y PACIENTES, que revisara las citas agendadas para el día siguiente al día actual, en caso de encontrar uno que cumpla con la fecha del día siguiente extrae el número de teléfono de la cita y busca al paciente para realizar el envío del recordatorio un día antes.

Anexo BB. Main

```
const main = async () => {
  await conectarMongoDB();

  const adapterFlow = createFlow([
    flowBienvenida,
    menuFlow,
    flowServicios,
    flowCita,
    flowHora,
    flowRecomendaciones,
    flowDespuesDeCita
  ])

  const adapterProvider = createProvider(BaileysProvider);
  createBot({
    flow: adapterFlow,
    provider: adapterProvider,
    database: adapterDB,
  })

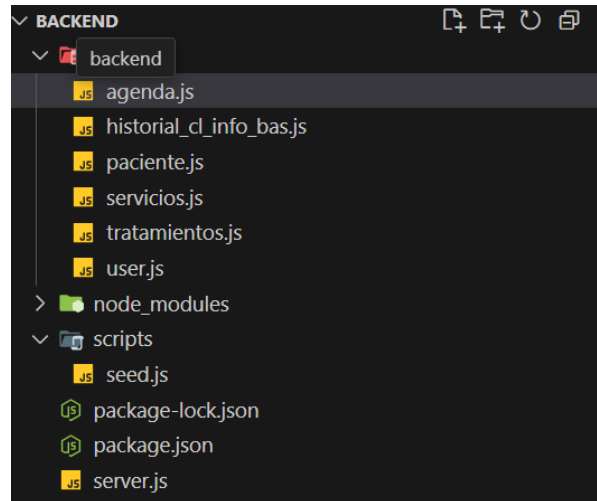
  setTimeout(async () => {
    try {
      const instance = await adapterProvider.getInstance();
      sock = instance;
      console.log("✅ Socket listo para enviar mensajes");
    } catch (err) {
      console.error("❌ No se pudo obtener el socket:", err);
    }
  }, 2000);
  setTimeout(enviarrecordatorio, 24 * 60 * 60 * 1000); // Ejecutar cada 24 horas
  QRPortalWeb();
}

main()
```

Nota, Función en donde se realizará todas las funciones definidas anteriormente, los flujos,

la conexión a base, a los servicios de Baileys y la sincronización para el envío de los registros

Anexo CC. Modelos empleados para el BackEnd



Nota, Para el back se procedio a crear los modelos que serán empleados en el proyecto, como se verán a continuación:

Donde cada archivo .js tiene su propia configuración con los campos que lo construyen y la colección a la que pertenecen dentro de la base de MongoDB

```
const mongoose = require('mongoose');

const agendaSchema = new mongoose.Schema({
  nombre: String,
  telefono: {type: String, required: true},
  fecha: Date,
  hora: String,
  status: {type: String, default: "Activo"}
});

module.exports = mongoose.model('Agenda', agendaSchema);
```

```
const mongoose = require('mongoose');

const pacienteSchema = new mongoose.Schema({
  nombre: String,
  telefono: String
});

module.exports = mongoose.model('Paciente', pacienteSchema);
```

```
const mongoose = require('mongoose');

const tratamientoSchema = new mongoose.Schema({
  telefono: String,
  fecha: Date,
  tratamiento: String,
  recomendaciones: String,
  evaluacion: String,
  observaciones: String,
});

module.exports = mongoose.model('Tratamiento', tratamientoSchema);
```

```
const mongoose = require('mongoose');

const serviciosSchema = new mongoose.Schema({
  fecha: Date,
  servicio: String,
  descripcion: String,
  precio: Number
});

module.exports = mongoose.model('Servicios', serviciosSchema);
```

```
const mongoose = require('mongoose');

const historialClInfoBasSchema = new mongoose.Schema({
  telefono: String,
  nombre_completo: String,
  fecha_nacimiento: Date,
  edad: Number,
  genero: String,
  direccion: String,
  correo: String,
  fecha_ingreso: Date,

  sangre: String,
  peso: Number,
  altura: Number,
  alergias: String,
  enfermedades: String,

  contacto_emergencia: String,
  relacion: String,
  telefono_emergencia: String,
});

module.exports = mongoose.model('HistorialClInfoBas', historialClInfoBasSchema);
```

```
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
  name: {type: String, required: true, unique: true},
  price: {type: Number, required: true}
});

module.exports = mongoose.model('User', userSchema);
```

Anexo DD. Apartado de los Endpoints

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');

//modelos
const User = require('./models/user');
const Agenda = require('./models/agenda');
const Paciente = require('./models/paciente');
const HistorialClInfoBas = require('./models/historial_cl_info_bas');
const Tratamiento = require('./models/tratamientos');
const Servicios = require('./models/servicios');

const app = express();
const PORT = 5000;

// Middleware
app.use(bodyParser.json());
app.use(cors());

// MongoDB connection
mongoose.connect('mongodb://127.0.0.1:27017/usuarios', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

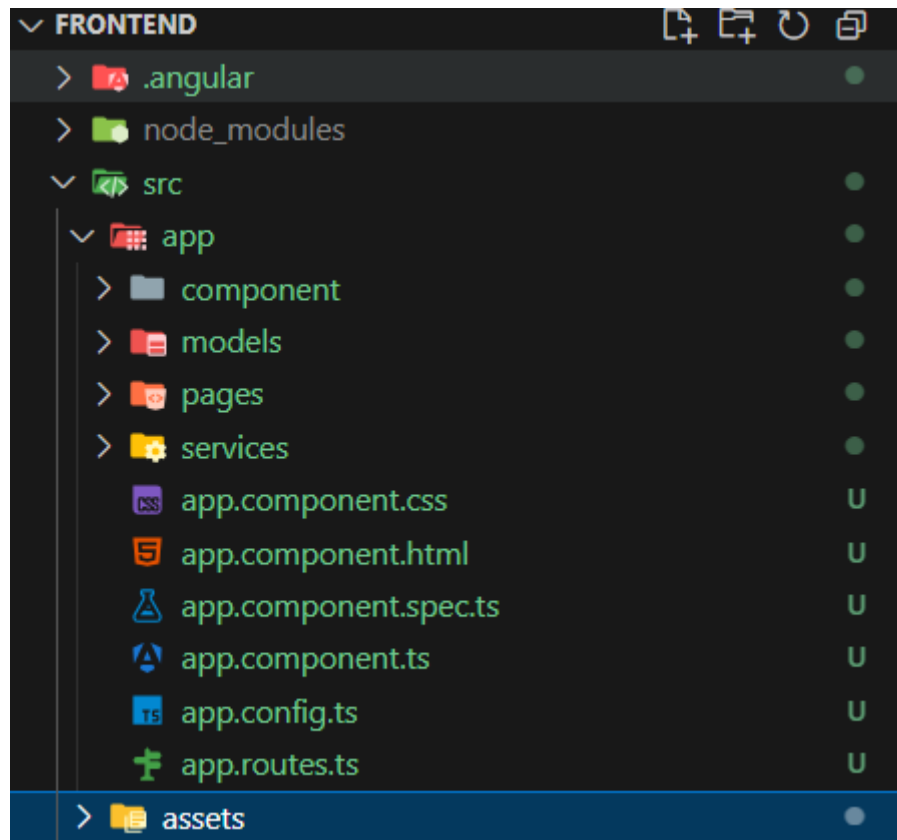
const db = mongoose.connection;
db.on('error', console.error.bind(console, 'MongoDB connection error:'));
db.once('open', () => {
  console.log('Connected to MongoDB');
});

// Start server
app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});
```

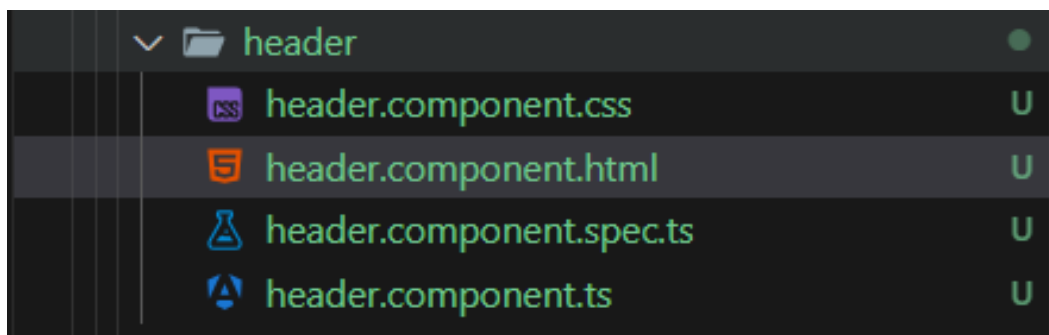
Nota, se configuran las conexiones a la base de datos y modelos para ejecutar las operaciones CRUD de cada servicio.

Anexo EE. Front.

Las carpetas mostradas a continuación, son las esenciales para el desarrollo del proyecto:



Nota, Dejando a un lado las carpetas predeterminadas que contienen toda la librería de los componentes que son Angular y node_modules. Las que nosotros crearemos a medida que se avanzó con el proyecto fueron las de component que almacenara el header empleado en todas las pantallas del entorno web.



Nota, En donde, se encontrará los archivos respectivos, donde editar la apariencia en el HTML y en el component las funciones respectivas que serán llamadas

```
src > app > component > header > 📄 header.component.html > ...
```

```
Go to component
1  ✓ <header class="header">
2  ✓   <div class="header-content">
3      
4      <span class="logo-name">Stella Spa</span>
5    </div>
6  </header>
7
```

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import { HeaderComponent } from './header.component';

describe('HeaderComponent', () => {
  let component: HeaderComponent;
  let fixture: ComponentFixture<HeaderComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [HeaderComponent]
    })
    .compileComponents();

    fixture = TestBed.createComponent(HeaderComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Anexo FF. App.Routes

```
export const routes: Routes = [
  {
    path: '',
    redirectTo: 'Principal',
    pathMatch: 'full'
  },
  {
    path: 'Principal',
    component: InicioComponent,
  },
  {
    path: 'login',
    component: LoginComponent
  },
  {
    path: '',
    component: LayoutComponent,
    children: [
      {path: 'home', component: HomeComponent},
      {path: 'home/agenda', component: AgendaComponent},
      {path: 'home/consulta', component: ConsultaComponent},
      {path: 'home/consulta/paciente/:telefono', component: PacienteDetailComponent},
      {path: 'home/asistentevirtual', component: AvirtualComponent}
    ]
  }
];
```

Nota, Permite la navegación entre las distintas pantallas de la plataforma, como home, login, agenda, consulta, detalle de paciente y asistente virtual.

Anexo GG. Services

```
private apiUrl = 'http://localhost:5000/agenda';

constructor( private http: HttpClient) {

}

//obtener la agenda
getAgenda(): Observable<any[]> {
  return this.http.get<any[]>(this.apiUrl);
}

//ingresar
addAgenda(telefono: string, fecha: Date, hora: string): Observable<any>{
  return this.http.post(`${this.apiUrl}/create`,{telefono,fecha,hora});
}

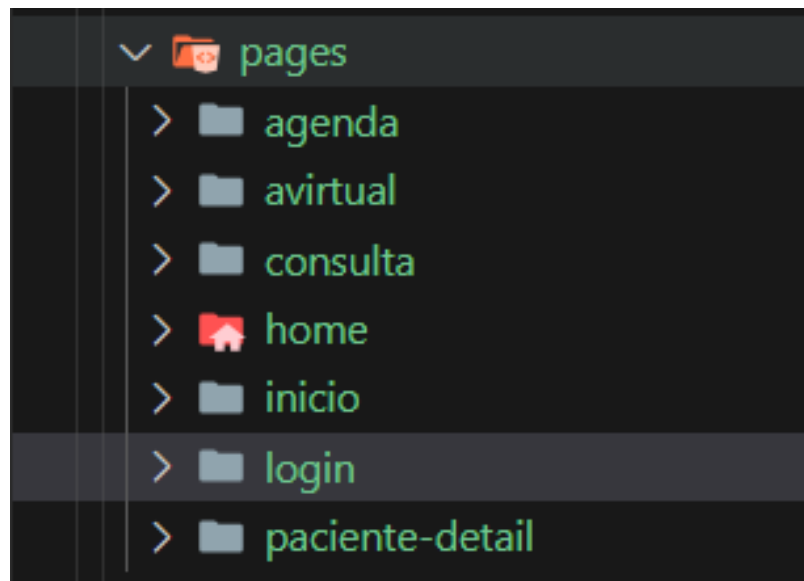
//agenda por id
getAgendaID(id: string): Observable<any> {
  return this.http.get<any>(`${this.apiUrl}/${id}`);
}

//actualizar
updateAgenda(id: string, agenda:Agenda): Observable<any> {
  return this.http.put<any>(`${this.apiUrl}/update/${id}`, agenda);
}

//eliminar
deleteAgenda(id: string): Observable<any> {
  return this.http.delete(`${this.apiUrl}/delete/${id}`);
}
}
```

Nota, definen los servicios que conectan con el Backend, incluyendo las URLs, parámetros y cuerpos para ejecutar métodos GET, POST, PUT y DELETE desde las páginas correspondientes.

Anexo HH. Pages



Nota, Dentro de las carpetas de pages contaremos con todas las vistas que serán usadas en toda la plataforma web. Cada una de las vistas contara con su respectivo archivo .css, html y component.ts.

De igual manera se mantiene por separado la configuración para cada una de las vistas, manejando las funciones, librerías, condiciones, validaciones y el llamado a los servicios dentro del archivo component, quedando así la parte estética en él HTML.

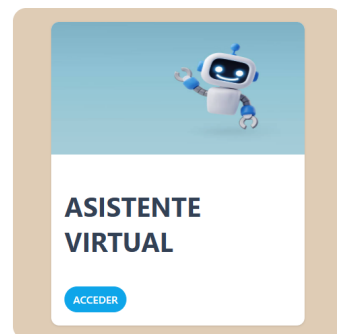
En la vista de Home, se visualizará las tres vistas esenciales del proyecto por donde las cuales el administrador podrá acceder a cada uno de ellos.

```
<body>
  <div class="container">
    <div class="box">
      <div class="mb-4 p-8 flex items-center justify-center">
        <ng-template #header>
          
        </ng-template>

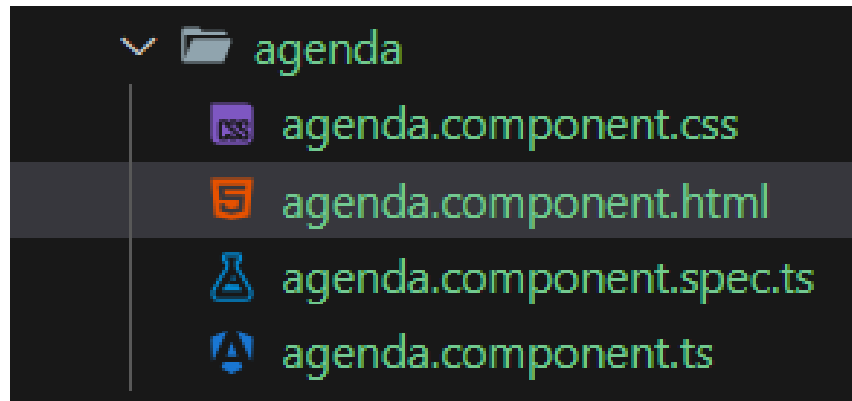
        <ng-template #title>
          <h1>AGENDA DE CONSULTAS </h1>
        </ng-template>

        <ng-template #footer>
          <div class="flex gap-4 mt-1">
            <p-button (onClick)="onAgenda()" label="ACCEDER" [rounded]="true" severity="info"/>
          </div>
        </ng-template>
      </p-card>
    </div>
  </div>
  <div class="box">
    <div class="mb-4 p-8 flex items-center justify-center">
      <p-card [style]="{ width: '25rem', overflow: 'hidden' }">
```

```
})  
export class HomeComponent {  
  router = inject(Router);  
  
  onAgenda(): void{  
    this.router.navigateByUrl("home/agenda");  
  }  
  
  onConsulta(): void{  
    this.router.navigateByUrl("home/consulta");  
  }  
  
  onAsistente(): void{  
    this.router.navigateByUrl("home/asistentevirtual");  
  }  
}
```



Para la vista de agenda de consultas, donde el usuario administrador podrá observar las citas agendadas con sus fechas, horas, y el estado. Además de poder realizar modificaciones en caso de verlo necesario y eliminarlas.



```
CargarAgendas() {
  this.customerService.getAgenda().subscribe((data) => {
    this.agendas = data;
    console.log(data);
  })
}

AgregarConsulta() {
  if (this.addConsulta.valid) {
    const consultaData = this.addConsulta.value;
    const hora = consultaData.hora;

    if (!this.validarHora(hora)) {
      alert('La hora debe estar entre las 08:30 y las 17:59.');
```

return;

}

//agregar el 593 al telefono

let telefono = consultaData.telefono;

if (telefono.startsWith('0')) {

telefono = '593' + telefono.slice(1); // reemplaza solo el primer 0

}

// Llamar al servicio para agregar la consulta

this.customerService.addAgenda(telefono, consultaData.fecha, consultaData.hora)

.subscribe({

next: (response) => {

console.log('Consulta agendada correctamente:', response);

alert('Consulta agendada correctamente');

this.visible = false; // Cierra el modal

this.CargarAgendas(); // Recarga la lista de agendas después de agregar

this.addConsulta.reset(); // Resetea el formulario

},

error: (err) => {

console.log(consultaData.value)

```

<div class="card flex justify-center">
  <h1>AGENDA DE CONSULTAS</h1>

  <p-button (click)="showDialog()" label="+ AGENDAR CONSULTA" severity="info"/>
  <p-dialog header="AGENDAR CITA" [modal]="true" [(visible)]="visible" [style]="{ width: '25rem', height: '50rem' }">

    <form [formGroup]="addConsulta" (ngSubmit)="AgregarConsulta()">
      <div class="mb-4 flex items-center">
        <label for="telefono" class="font-semibold w-24">Numero de telefono: </label>
        <div>
          <input pInputText id="telefono" formControlName="telefono" class="w-full p-2 border border-gray-300 rounded-md" />
        </div>
      </div>
      <br>
      <div class="mb-4 flex items-center">
        <label for="fecha" class="font-semibold w-24">Fecha: </label>
        <div>
          <p-datepicker
            formControlName="fecha"
            [showButtonBar]="true"
            [minDate] = "minDate2"
          />
        </div>
      </div>
      <br>
      <div class="mb-10 flex items-center">
        <label for="hora" class="font-semibold w-24">Hora: </label>
        <div>
          <input type="time" formControlName="hora" class="form-control" />
        </div>
      </div>
    </form>
  </div>
</div>

```

 INICIO CONSULTA ASISTENTE VIRTUAL					
AGENDA DE CONSULTAS					
+ AGENDAR CONSULTA					
#	Nombre	Fecha	Hora	Estado	
1	Marc	2025-06-29	15:00	pendiente	 
2	Mary	2025-07-28	17:30	pendiente	 
3	Mell De la cruz	2025-07-22	13:00	pendiente	 
4	Marc	2025-06-30	16:00	pendiente	 
5	Marc	2025-06-30	17:00	pendiente	 

Nota, También cuenta con poder ingresar una nueva cita, desde la vista en caso de que la reserva sea de forma presencial.

AGENDAR CITA

Numero de telefono:

Fecha:

Hora:

Agendar

En la vista del historial del paciente se podrá visualizar todos los pacientes que se encuentran registrados en el sistema, con un botón de direccionamiento que la redirigirá al perfil personal de cada uno de ellos donde se podrá visualizar información básica del paciente, las citas que ha reservado y los tratamientos realizados.

```

</div>
<div class="mb-4">
  <p-button type="button" icon="pi pi-chevron-left" (click)="prev()" [disabled]="isFirstPage()" text />
  <p-button type="button" icon="pi pi-refresh" (click)="reset()" text />
  <p-button type="button" icon="pi pi-chevron-right" (click)="next()" [disabled]="isLastPage()" text />
</div>
<div class="card">
  <p-table
    [value]="pacientes"
    [paginator]="true"
    [rows]="5"
    [first]="first"
    [showCurrentPageReport]="true"
    [tableStyle]="{ 'min-width': '50rem' }"
    currentPageReportTemplate="Showing {first} to {last} of {totalRecords} entries"
    (onPage)="pageChange($event)"
    [rowsPerPageOptions]="[10, 25, 50]"
  >
    <ng-template #header>
      <tr>
        <th style="width:5%">#</th>
        <th style="width:25%">Paciente</th>
        <th style="width:10%"></th>
      </tr>
    </ng-template>
    <ng-template #body let-paciente let-i="rowIndex">
      <tr>
        <td> {{ i+1 }} </td>
        <td>{{ paciente.nombre }}</td>
        <td>
          <p-button (onClick)="OnPerfil(paciente)" label="Perfil" icon="pi pi-user" />
        </td>
      </tr>
    </ng-template>
  </p-table>

```

```

isLastPage(): boolean {
  return this.pacientes ? this.first + this.rows >= this.pacientes.length : true;
}

pageChange(event: any) {
  this.first = event.first;
  this.rows = event.rows;
}

traerData():void{
  this.pacienteService.getPacientes().subscribe((data) =>{
    this.pacientes = data;
  })
}

OnPerfil(paciente: Paciente): void{
  this.router.navigateByUrl(` /home/consulta/paciente/${paciente.telefono}`);
  console.log(this.router.navigateByUrl);
}
}

```

INICIO AGENDA ASISTENTE VIRTUAL		
< >		
#	Paciente	
1	Marc	Perfil
2	Anthony	Perfil
3	Leonela	Perfil
4	Michael	Perfil
5	Mauricio Moreno	Perfil
Showing 1 to 5 of 8 entries << < 1 2 > >>		



[Editar](#)
[Guardar](#)

Paciente:
Información Básica
Nombre completo:
 Marcell Andre Moreno Salcedo
Fecha de nacimiento:
 2000-06-14T00:00:00.000Z
Edad:
 24

Historial de consultas

Nombre	Fecha	Hora
Marc	2025-06-29	15:00
Marc	2025-06-30	16:00
Marc	2025-06-30	17:00
Marc	2025-07-12	18:00
Marc	2025-07-12	12:00
Marc	2025-07-14	13:00

Historial de procedimientos

+PROCEDIMIENTO

Fecha	Tipo de tratamiento	Recomendaciones post-sesión	Evolución o seguimiento	Observaciones		
2025-06-28T17:21:43.096Z	acne	nuevas	evalua	verse cada seman	Editar	Eliminar
2025-06-28T17:43:21.630Z	Limpieza facial profunda y tratamiento antiacné	No tocar el rostro, evitar exposición al sol por 48h, usar protector solar FPS 50	La piel respondió bien, disminuyó el enrojecimiento	Se detectaron comedones activos en zona T. Posible seguimiento en 15 días	Editar	Eliminar

Procedimiento

Tratamiento:

Recomendaciones:

Evaluación:

Observaciones:

Añadir

Y por último la vista del asistente virtual, donde el administrador podrá alimentar de información al ChatBot, donde definirá datos importantes de cada tratamiento que en este caso están agrupados por categorías.

Combos

Descripción

Contamos con combos a buenos precios por una limpieza facial te llevas un masaje relajante post limpieza.

Save

Tratamientos de Depilación

Descripción

Información general para tratamientos de estética y depilación. Cada procedimiento incluye detalles clave para brindar una atención personalizada y segura. La información está organizada en las siguientes categorías:

Áreas del cuerpo disponibles: Indica qué zonas se pueden tratar.

Beneficios principales: Ventajas del tratamiento, como suavidad, durabilidad o efectos estéticos visibles.

Save

Tratamientos Especiales

Descripción

Eres un asistente virtual especializado en terapias especiales. Tu objetivo es brindar respuestas claras, detalladas y profesionales sobre los tratamientos disponibles en esta categoría. Proporciona información clave sobre los siguientes puntos:

Descripción del tratamiento: Explica de manera breve y comprensible en qué consiste cada terapia especial.

Beneficios principales: Resalta los resultados esperados, como alivio, bienestar, o mejoras físicas o emocionales.

Condiciones ideales: Indica para qué tipo de personas o

Save

Tratamientos Corporales

Descripción

Descripción del tratamiento: Explica en qué consiste cada procedimiento de forma breve y fácil de entender.

Beneficios principales: Detalla qué mejoras o resultados puede esperar el cliente tras el tratamiento.

Recomendaciones previas: Indica si hay algo que el cliente deba hacer o evitar antes de realizar el tratamiento.

Recomendaciones posteriores: Brinda consejos para maximizar los resultados después del tratamiento.

Duración del tratamiento: Indica cuánto tiempo dura cada sesión.

Save

Tratamientos de extremidades inferiores

Descripción

Eres un asistente virtual especializado en tratamientos de glúteos y escultura corporal. Brinda respuestas claras y detalladas sobre esta categoría, enfocándose en los beneficios, procedimientos y recomendaciones. Asegúrate de incluir la siguiente información clave:

Descripción del tratamiento: Explica en qué consiste cada procedimiento de manera sencilla y profesional.

Objetivo principal: Indica cuál es el propósito del tratamiento (tonificación, aumento de volumen, reducción de grasa, etc.).

Save

Tratamientos Faciales

Descripción

Eres un asistente virtual experto en tratamientos faciales de un centro de estética. Al cliente, proporciona información clara y breve sobre los tratamientos faciales disponibles. Para cada tratamiento, menciona:

Su función principal.

Los beneficios para la piel.

La duración estimada.

Las recomendaciones según el tipo de piel o necesidades (si aplica).

Save

```

constructor(
  private asistvirtualService: AsistvirtualService,
  private http: HttpClient

) {

}

ngOnInit() {
  this.items = [
    {
      label: 'INICIO',
      icon: 'pi pi-home',
      command: () =>this.router.navigate(['/home']),
    },
    {
      label: 'AGENDA',
      icon: 'pi pi-envelope',
      command: () =>this.router.navigate(['/home/agenda']),
    },
    {
      label: 'CONSULTA',
      icon: 'pi pi-envelope',
      command: () =>this.router.navigate(['/home/consulta']),
    }
  ]

  this.getServicioPorId(this.combosID, 'combos');
  this.getServicioPorId(this.depilacionID, 'depilacion');
  this.getServicioPorId(this.especialID, 'especial');
  this.getServicioPorId(this.corporalID, 'corporal');
  this.getServicioPorId(this.inferioeresID, 'inferioeres');
  this.getServicioPorId(this.facialID, 'facial');

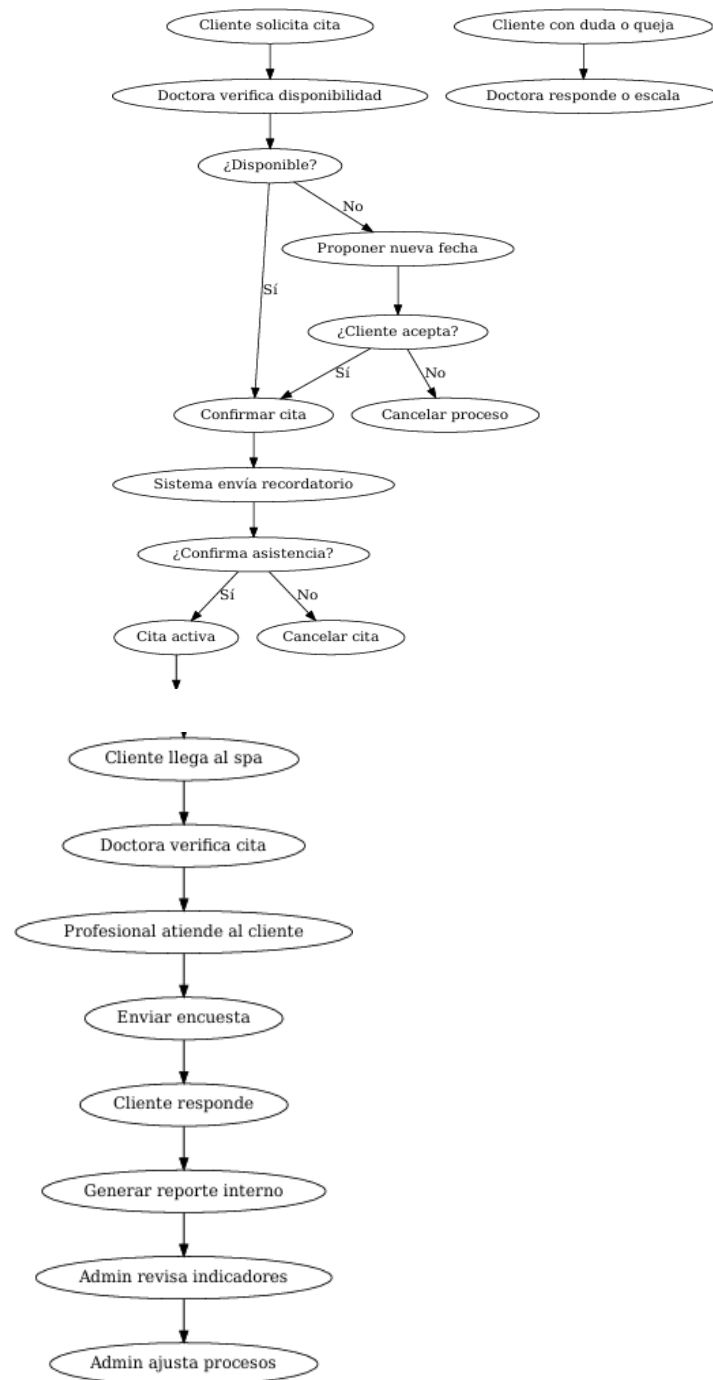
```

```

<div class="container">
  <div class="box">
    <div class="mb-4 p-8 flex items-center justify-center">
      <p-card [style]="{ width: '35rem', height: '35rem', overflow: 'hidden' }">
        <ng-template #header>
          
        </ng-template>
        <ng-template #title> Combos </ng-template>
        <ng-template #subtitle> Descripcion </ng-template>
        <div class="card flex justify-center">
          <p-iftalabel>
            <textarea pTextarea id="description_combos"
              [(ngModel)]="value_combos"
              rows="10"
              cols="55"
              style="resize: none"
              class="mi-textarea">
            </textarea>
          </p-iftalabel>
        </div>
        <ng-template #footer>
          <div class="flex gap-4 mt-1">
            <p-button label="Save"
              class="w-full"
              styleClass="w-full"
              (onClick)="updateServicioPorId(combosID, value_combos)" />
          </div>
        </ng-template>
      </p-card>
    </div>
  </div>
</div>

```

Anexo II. DIAGRAMA DE FLUJO DE STELLA SPA



Anexo JJ. GRÁFICOS DE ENCUESTAS

1. ¿Con qué frecuencia visitas Stella Spa?

40 respuestas

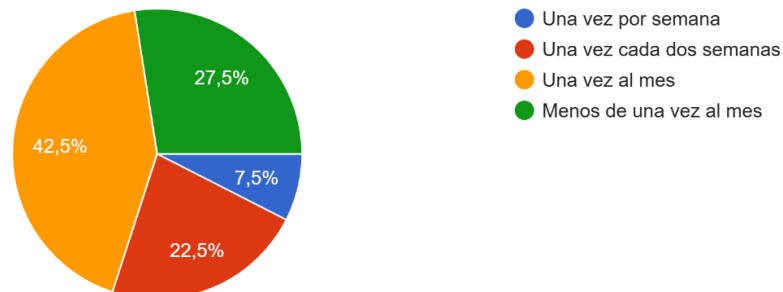


Figura 20. Pregunta 1. ¿Con qué frecuencia visitas Stella Spa?

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

2. ¿Cómo agendas actualmente tus citas en Stella Spa?

39 respuestas

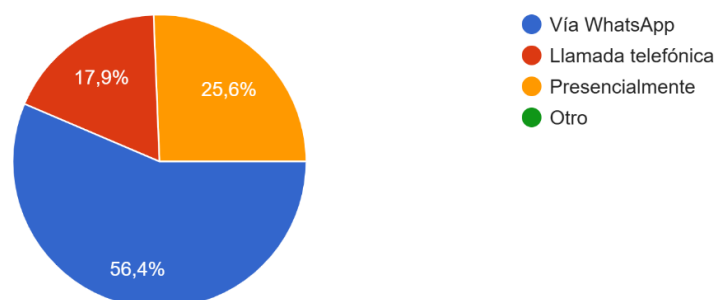


Figura 21. Pregunta 2. ¿Cómo agendas actualmente tus citas en Stella Spa?

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

¿Has tenido dificultades al momento de agendar tu cita?

40 respuestas

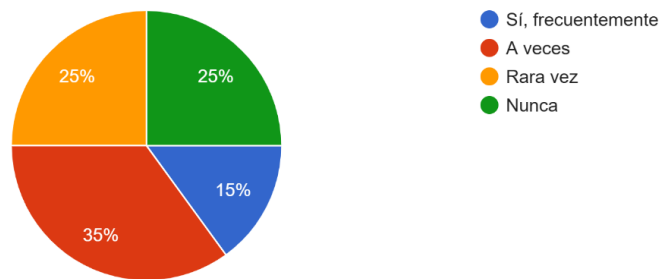


Figura 22. Pregunta 3. ¿Has tenido dificultades al momento de agendar tu cita?

¿Te gustaría poder agendar citas mediante un asistente virtual disponible 24/7?

40 respuestas

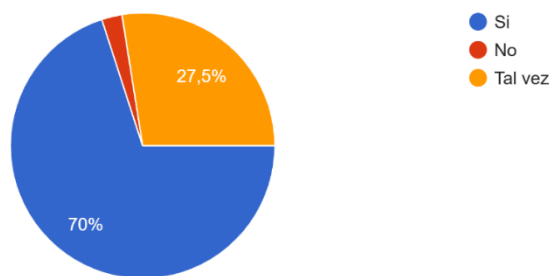


Figura 23. Pregunta 4. ¿Te gustaría poder agendar citas mediante un asistente virtual disponible 24/7?

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

¿Consideras importante recibir recordatorios automáticos de tus citas?

39 respuestas

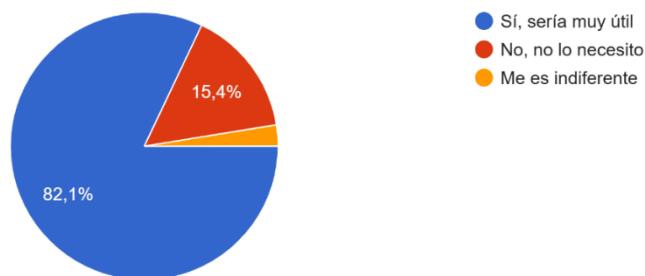


Figura 24. Pregunta 5. ¿Consideras importante recibir recordatorios automáticos de tus

citas?

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

¿Qué tan importante es para ti recibir recomendaciones de tratamientos personalizadas basadas en tu historial?

40 respuestas

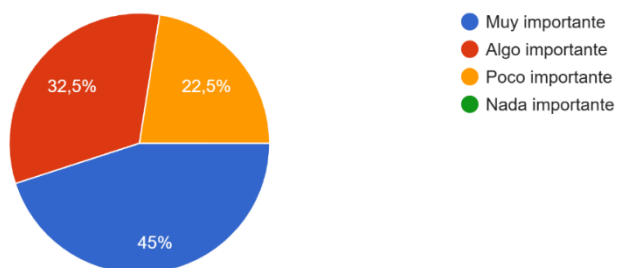


Figura 25. Pregunta 6. ¿Qué tan importante es para ti recibir recomendaciones de tratamientos personalizadas basadas en tu historial?

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

¿Crees que el servicio actual en Stella Spa está suficientemente personalizado a tus necesidades?

40 respuestas

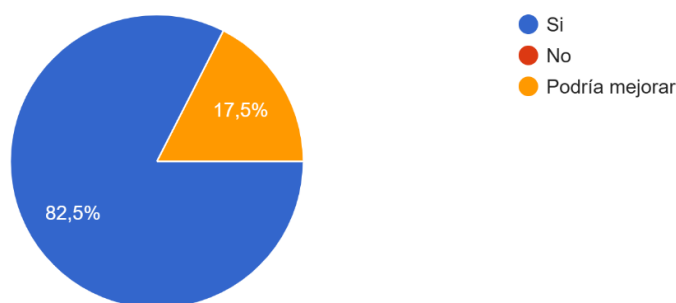


Figura 26. Pregunta 7. ¿Crees que el servicio actual en Stella Spa está suficientemente personalizado a tus necesidades?

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

¿Qué canal preferirías para interactuar con el asistente virtual?

40 respuestas

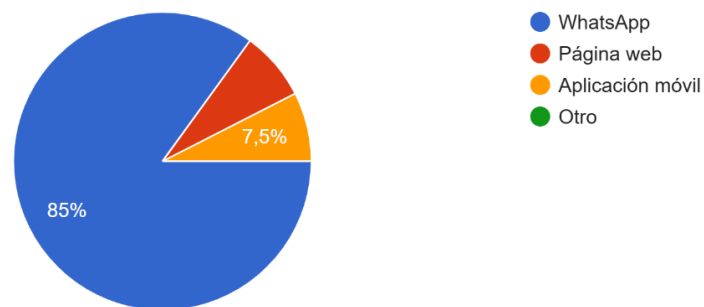


Figura 27. Pregunta 8. ¿Qué canal preferirías para interactuar con el asistente virtual?

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

¿Estarías dispuesto/a a proporcionar tus datos personales si se garantiza su protección bajo la Ley de Protección de Datos?

40 respuestas

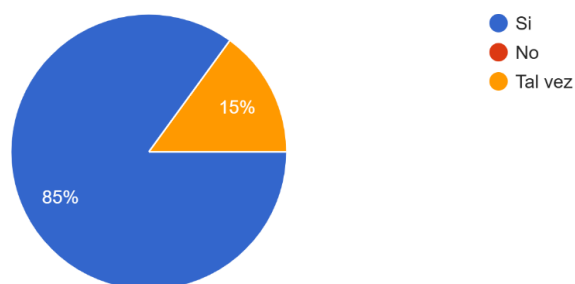


Figura 28. Pregunta 9. ¿Estarías dispuesto/a a proporcionar tus datos personales si se garantiza su protección bajo la Ley de Protección de Datos?

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

¿Recomendarías a algún familiar o amigo Stella Spa?

40 respuestas

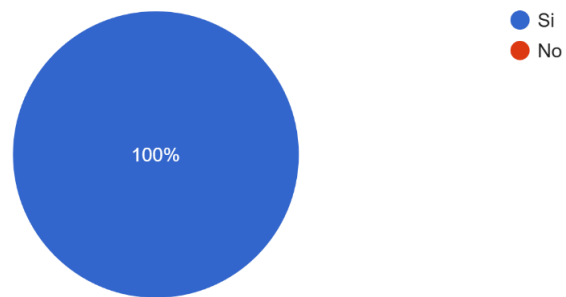


Figura 29. *Pregunta 10. ¿Recomendarías a algún familiar o amigo Stella Spa?*

Nota, Elaboración propia, (Moreno & Ramírez, 2025).

Firma del Tutor

Ing. Espinoza Mina Marcos

Firma del Estudiante 1

Moreno Salcedo Marcell Andre

Firma del Estudiante2

Ramírez Ochoa Tommy Elian