

UNIVERSIDAD SALESIANA DE BOLIVIA

INGENIERÍA DE SISTEMAS

PROYECTO DE GRADO



**Para obtener el Grado Académico de
Licenciatura en Ingeniería de Sistemas**

SISTEMA WEB DE INFORMACIÓN TURÍSTICA CON WEB SCRAPING

Postulante: Dennis Abad Ortuño Villazon

Docente Guía: Gustavo Adolfo Veizaga

**COCHABAMBA – BOLIVIA
2024**

DEDICATORIA

El presente proyecto va dedicado a las personas que siempre confiaron en mí, a mis padres y familiares que me hicieron una persona de bien.

*Dennis Abad Ortuño
Villazon*

AGRADECIMIENTO

Primeramente, agradecer a Dios, por brindarme las oportunidades.

A la universidad por abrirme las puertas y permitir culminar mis estudios

A mis padres que me apoyaron y guiaron por el buen camino y a mi tutor Ing. Gustavo Adolfo Veizaga por brindarme las herramientas y el apoyo siempre para culminar el proyecto.

¡MUCHAS GRACIAS!

FICHA RESUMEN

El proyecto de Ingeniería de Sistemas tiene como objetivo abordar la falta de información actualizada y la comunicación ineficiente en el sector turístico de Cochabamba, Bolivia. Para ello, se propone implementar un chatbot basado en Inteligencia Artificial y técnicas de web scraping. Este chatbot permitirá a los turistas acceder a información precisa y actualizada sobre los destinos turísticos del departamento, mejorando significativamente su experiencia de viaje. Además, el chatbot estará disponible las 24 horas del día, abordando así la falta de atención continua al cliente y facilitando respuestas a consultas frecuentes de manera eficiente y amigable.

El proyecto incluye la investigación de plataformas y técnicas para desarrollar un chatbot efectivo, el diseño de una interfaz de usuario amigable, y la implementación de un sistema de web scraping para mantener la información turística actualizada. La solución tecnológica mejorará la calidad de la información disponible y optimizará la planificación de los itinerarios de los turistas, proporcionando datos geográficos, distancias y tiempos de viaje estimados. Este enfoque integral no solo beneficiará a los turistas, sino que también promoverá el turismo en la región, contribuyendo al desarrollo económico y social del departamento de Cochabamba.

ÍNDICE

Contenido

1 INTRODUCCIÓN	1
CAPÍTULO I	3
MARCO REFERENCIAL	3
1.2. ANTECEDENTES	4
1.3. DEFINICIÓN DEL PROBLEMA	4
1.3.2. Árbol de problemas.....	6
1.3.2.1 Formulación del problema	6
1.2. OBJETIVOS	7
1.2.1. OBJETIVO GENERAL	7
1.2.2. OBJETIVOS ESPECÍFICOS	7
1.3. ALCANCES	7
1.4. DELIMITACIONES DEL TEMA	8
1.4.1. DELIMITACION TEMATICA	8
1.5. JUSTIFICACION	9
1.5.1 Justificación Social.....	9
1.5.2 Justificación técnica	9
1.5.3 Justificación Económica.....	10
CAPÍTULO II	11
MARCO TEÓRICO	11
2.1 Marco Teórico	12
2.1.1 Elementos Principales	12
2.1.1.1 Chatbots Inteligentes.....	12
2.1.1.2 Web Scraping.....	14
2.1.1.3 Páginas Web.....	15
2.1.2 Técnicas Recolección Información.....	17
2.1.2.1 Observación Directa	17
2.1.2.2. Web Scraping.....	18
2.1.2.3 Procesamiento del lenguaje natural (NLP)	19
2.1.3 Metodologías de Desarrollo	20
2.1.3.1 Metodología Extreme Programming (XP).....	20

2.1.3.2 Ciclo de Vida de XP:	20
2.1.4 Plataformas de desarrollo	22
2.1.4.1. Java Script.....	22
2.1.4.2. HTML	23
2.1.4.3. CSS.....	24
2.1.4.4. MongoDB.....	24
2.1.4.5. Elastisearch.....	25
2.1.5. Herramientas.....	27
2.1.5.1. Visual studio Code	27
2.1.5.2. Orimon.ai.....	28
2.1.6. Formatos de datos	29
2.1.6.1. JSON	29
2.1.6.2. XML.....	31
2.1.6.3. CSV.....	32
2.1.7 API de integración	33
2.1.7.1. OpenAi.....	33
CAPÍTULO III	36
MARCO PRÁCTICO	36
3.1 Análisis de Requerimientos del Sistema.....	37
3.1.1 Proceso Actual.....	37
3.1.2 Historias de Usuario:.....	38
3.1.3 Diagrama de Caso de Uso General Del Sistema	41
3.1.4 Plan de Iteraciones	42
3.2 Fase de Construcción	43
3.2.1 Módulo 1	43
3.2.1.1 Planificación	43
3.2.1.1.2 Observación directa	43
3.2.2 Módulo 2.....	44
3.2.2.1 Diseño	44
3.2.2.1.1 Estructura de datos.....	44
3.2.2.1.2 Modelo de Entrenamiento	47
1.Recolección de datos	48
2. Preprocesamiento de datos	49

3. Etiquetado y Análisis de Datos	50
4. División de datos	50
5. Selección del Modelo y Técnica.....	51
6. Entrenamiento del modelo	52
7. Evaluación del modelo.....	53
8. Ajuste Fino y Optimización	53
9. Implementación y Despliegue	54
10. Monitoreo y Mejora Continua	55
Ciclo de Retroalimentación y Actualización del Modelo	55
3.2.2.1.3 Arquitectura Página Web	56
3.2.2.1.4 Requisitos Funcionales	57
3.2.2.1.5 Requisitos No Funcionales	60
CAPÍTULO IV.....	72
FASE DE DESARROLLO	72
4.1. Desarrollo	73
4.1.1. Diseños del proyecto	73
4.2.1.1. Diseño Página Web	73
4.2.1.2. Título y portada de un sitio de la página web.....	74
4.2.1.3. Contenido de un sitio de la página web	75
4.2.1.4. Pie de Página	76
4.2. Diseño ChatBot.....	77
4.2.1. Diseño de ventana emergente.....	77
4.2.3 Módulo 3.....	78
4.2.3.1 Desarrollo.....	78
4.2.3.1.1. Código de conexión	78
4.2.3.1.2. Estilo de la ventana Emergente	78
4.2.3.1.3. Enlace de vinculación	79
4.2.3.1.4. Código de búsquedas de las páginas web externas	80
4.2.4.Módulo 4.....	82
4.2.4.1 Pruebas.....	82
4.2.4.2 Entrenamiento de Chatbot.....	82
4.1.4.3. Fluidez en conversación	84
4.2.4.4. Fluidez de chatbot y pagina web	88

5 conclusiones y Recomendaciones	89
5.1 Conclusiones	89
5.2 Recomendaciones	89
Bibliografía	90
WEBGRAFÍA	92
ANEXOS	1

ÍNDICE FIGURAS

Figura 1 Estructura de un chatbot Inteligente	13
Figura 2 ¿Qué es web escraping?	15
Figura 3 Estructura web	17
Figura 4 Ciclo de vida programacion extrema (XP)	21
Figura 5 Logo Java Script	22
Figura 6 Logo HTML	23
Figura 7 Logo CSS	24
Figura 8 Logo de MongoDB	25
Figura 9 Logo de Elastisearch	26
Figura 10 Logo Visual Studio Code	28
Figura 11 Logo de Orimon.ai	29
Figura 12 Logo de JSON	30
Figura 13 Logo de JSON	32
Figura 14 Logo de CSV	33
Figura 15 Logo OpenAi	35
Figura 16 Diagrama Casos de uso del uso del programa	41
Figura 17 Recolección de datos de usuario	48
Figura 18 Flujo de Chatbot	49
Figura 19 Análisis de datos	50
Figura 20 Validación de pruebas	51
Figura 21 Redes neuronales recurrentes	52
Figura 22 Descenso de gradiente	52

Figura 23 Métricas de evaluación	53
Figura 24 Optimización de hiperparámetros	54
Figura 25 Integración continua.....	54
Figura 26 Monitoreo continuo	55
Figura 27 Retroalimentación continua	56
Figura 28 Arquitectura Web	57
Figura 29 Wireframe de la Pagina web	59
Figura 30 Requisitos No funcionales	61
Figura 31 Procesamiento NLP	62
Figura 32 Recepción de consultas de usuarios	63
Figura 33 Resumen de la consulta de un usuario.....	64
Figura 34 Datos extras de la consulta de un usuario.	65
Figura 35 Grafica de Frecuencia.....	66
Figura 36 Ejemplo de solicitud	67
Figura 37 Respuesta en formato Json	68
Figura 38 Nube de Palabras	69
Figura 39 Pregunta de Usuario	70
Figura 40 Web Scraping Ejemplo	70
Figura 41 Respuesta del chatbot	71
Figura 42 Portada Página Web.....	73
Figura 43 Menú desplegable Página Web	74
Figura 44 Titulo y portada de un sitio web	74
Figura 45 Contenido de sitio sección geografía y clima.....	75
Figura 46 Contenido del sitio sección Biodiversidad y ¿Cómo llegar?.....	76

Figura 47 Contenido en la sección pie de página.	76
Figura 48 Diseño Chatbot Emergente.....	77
Figura 49 Código de Conexión del bot.....	78
Figura 50 Estilo ventana emergente	79
Figura 51 Código de vinculación Orimon.Ai y la Página web.....	80
Figura 52 Código de Biblioteca BeautifulSoup.....	81
Figura 53 Sitios de búsqueda Web Scraping	82
Figura 54 Entrenamiento de mensajes programados	83
Figura 55 PDF de folletos turísticos	83
Figura 56 Conversación casual 1.....	84
Figura 57 Conversación casual 2.....	85
Figura 58 Conversación casual 3.....	86
Figura 59 Conversación casual 4.....	86
Figura 60 Conversación casual 5.....	87
Figura 61 Portada web y chatbot	88

ÍNDICE DE TABLAS

Tabla 1 Historia de Usuario 1.....	38
Tabla 2 Historia de Usuario 2.....	39
Tabla 3 Historia de Usuario 3.....	39
Tabla 4 Historia de Usuario 4.....	40
Tabla 5 Historia de Usuario 5.....	40
Tabla 6 Iteraciones del proyecto	42

1 INTRODUCCIÓN

El presente proyecto de Ingeniería de Sistemas tiene como objetivo abordar una problemática creciente en el ámbito turístico del departamento de Cochabamba, Bolivia. La falta de información actualizada y la dificultad en la comunicación con turistas y visitantes han generado ineficiencias que afectan negativamente la experiencia de los viajeros. Para resolver este desafío, se propone la implementación de un chatbot respaldado por técnicas de Inteligencia Artificial y la incorporación de web scraping para mantener actualizada la información sobre los destinos turísticos.

Así mismo este proyecto ayudará proporcionando a los turistas una herramienta accesible y eficiente que les permita obtener información precisa y actualizada sobre los lugares de interés en el departamento de Cochabamba. Esto no solo mejorará la experiencia de los visitantes, sino que también facilitará el acceso a datos relevantes, como distancias, tiempos y presupuestos. Además, chatbot permitirá a los usuarios obtener respuestas a sus consultas en cualquier momento del día, lo que abordará la falta de atención 24 horas al cliente.

El proyecto abordará un análisis detallado de las plataformas y técnicas más apropiadas para la creación de un chatbot de alto rendimiento. Es importante destacar que la implementación de la IA se llevó a cabo mediante una solución comercial, lo que significa que no se desarrolló desde sus fundamentos. En este caso, se seleccionó **Orimon.ai**, una plataforma de pago que permite la integración de capacidades avanzadas de procesamiento de lenguaje natural, potenciando la capacidad de respuesta eficiente a las consultas de los usuarios. La elección de esta tecnología asegura un proceso de implementación ágil y eficaz, aprovechando una infraestructura probada y robusta en el sector. Además, se desarrollará una interfaz de usuario intuitiva que facilite la interacción con el sistema. La implementación del chatbot seguirá un enfoque

centrado en la eficiencia y efectividad en la comunicación con los usuarios, mediante técnicas avanzadas de inteligencia artificial y programación.

Además, se establecerá un sistema de web scraping para mantener actualizada la información turística, garantizando que los datos proporcionados por el chatbot estén siempre al día. Este proyecto aborda una necesidad real y contribuirá a mejorar la calidad de la información y la experiencia de los turistas que visitan Cochabamba.

El presente proyecto abordará el proceso de investigación, diseño, desarrollo e implementación de este chatbot, así como las etapas para la recopilación y actualización de información mediante web scraping. El resultado final será una solución tecnológica que responde eficazmente a la problemática planteada, brindando a los turistas una herramienta valiosa para su viaje y promoviendo el turismo en la región.

CAPÍTULO I

MARCO REFERENCIAL

1.2. ANTECEDENTES

En 2019, la Alcaldía de La Paz implementó un chatbot basado en inteligencia artificial para promover el turismo local. Este chatbot proporcionaba información actualizada sobre sitios turísticos, eventos culturales, y servicios como transporte y alojamiento. Utilizaba técnicas de procesamiento del lenguaje natural (NLP) para entender consultas en varios idiomas y ofrecer respuestas relevantes a los usuarios. La iniciativa mejoró significativamente la accesibilidad a la información turística y recibió una buena aceptación por parte de los turistas y residentes.

En 2020, Garibay Ornelas, investigador del INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación en México, desarrolló un proyecto sobre "Diseño e Implementación de un Asistente Virtual (Chatbot) para Ofrecer Atención a los Clientes de una Aerolínea Mexicana por Medio de sus Canales Conversacionales". Este proyecto tenía como objetivo demostrar las aplicaciones de la inteligencia artificial en el campo de la comunicación empresarial y también identificar las principales necesidades y áreas de oportunidad para el uso de un programa de este tipo en la interacción con los clientes. El proyecto agregó el diseño e implementación de un chatbot en Facebook, WhatsApp y Web, utilizando tecnología provista por el proveedor Aivo para fortalecer la interacción de la empresa con sus clientes.

Estos antecedentes muestran cómo la implementación de chatbots puede mejorar la experiencia del usuario en el sector turístico, facilitando el acceso a información y servicios relevantes de manera eficiente y personalizada.

1.3. DEFINICIÓN DEL PROBLEMA

En el contexto actual, los turistas que visitan el departamento de Cochabamba enfrentan dificultades para acceder a información precisa, actualizada y estructurada sobre los sitios turísticos de la región. Esta carencia de una fuente centralizada y confiable de datos dificulta la planificación eficiente de itinerarios

y la toma de decisiones durante su visita, afectando negativamente la experiencia del usuario.

Aunque existen múltiples plataformas y fuentes que ofrecen información turística, a menudo se encuentran desactualizadas, dispersas y carecen de herramientas interactivas que permitan a los usuarios acceder rápidamente a la información relevante. Además, la falta de un sistema de asistencia virtual que pueda proporcionar respuestas personalizadas y orientaciones en tiempo real agrava el problema, dejando a los visitantes con una experiencia fragmentada y poco optimizada.

Por otro lado, la ausencia de una integración eficiente de tecnologías de procesamiento de lenguaje natural (NLP) y aprendizaje automático (Machine Learning) en las plataformas existentes limita la capacidad de ofrecer interacciones dinámicas y contextuales con los usuarios. Esto genera una brecha significativa en la forma en que se gestiona la información turística, haciendo que los datos disponibles no se presenten de manera adecuada ni cumplan con las expectativas de los usuarios modernos que buscan inmediatez y precisión.

El proyecto propuesto busca abordar estos desafíos mediante el desarrollo de un chatbot inteligente y una plataforma web interactiva que funcione de manera coordinada. Esta solución no solo utilizará técnicas avanzadas de inteligencia artificial para optimizar la interacción con los usuarios, sino que también integrará un sistema automatizado de web scraping para asegurar que la información turística esté siempre actualizada y disponible en tiempo real. Además, se diseñará un asistente virtual accesible a través de una ventana flotante, capaz de guiar a los visitantes y ofrecer una experiencia de usuario mejorada, con acceso intuitivo a itinerarios, mapas interactivos y recomendaciones personalizadas.

La falta de una herramienta centralizada y eficiente que combine estas capacidades tecnológicas representa un obstáculo significativo para maximizar

el potencial turístico de Cochabamba. Por lo tanto, este proyecto se presenta como una solución innovadora y necesaria para transformar la forma en que se accede e interactúa con la información sobre los destinos turísticos de la región.

1.3.2. Árbol de problemas



1.3.2.1 Formulación del problema

¿Cuál es la estrategia más adecuada para implementar un sistema de Inteligencia Artificial mediante un chatbot, complementado con técnicas de web scraping, que garantice el acceso a información turística precisa y actualizada en Cochabamba, Bolivia? El objetivo es mejorar la experiencia del usuario mediante un acceso eficiente y dinámico a datos relevantes, optimizando la planificación de itinerarios y la toma de decisiones. ¿Cómo se pueden integrar tecnologías de procesamiento de lenguaje natural (NLP) y un diseño de interfaz

intuitiva para ofrecer interacciones personalizadas y permitir un acceso rápido a itinerarios, mapas interactivos y recomendaciones precisas?

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Desarrollar un sistema de Inteligencia Artificial mediante un chatbot, integrado con técnicas de web scraping y tecnologías de procesamiento de lenguaje natural (NLP), que proporcione información turística precisa y actualizada sobre Cochabamba, Bolivia, optimizando la experiencia del usuario a través de interacciones personalizadas, y facilitando el acceso eficiente a itinerarios, mapas interactivos y recomendaciones relevantes.

1.2.2. OBJETIVOS ESPECÍFICOS

- **Implementar técnicas de Inteligencia Artificial:** Desarrollar un chatbot que utilice algoritmos de procesamiento de lenguaje natural (NLP) para entender y responder eficazmente a las consultas de los turistas sobre los destinos turísticos en Cochabamba.
- **Integrar un sistema de web scraping:** Diseñar e implementar un sistema de web scraping que recopile y actualice constantemente la información sobre los destinos turísticos, garantizando que los datos proporcionados sean siempre precisos y actuales.
- **Optimizar la interfaz de usuario del chatbot:** Cree una interfaz de usuario intuitiva y amigable que permita a los turistas interactuar fácilmente con el chatbot, accediendo rápidamente a la información relevante sobre los destinos turísticos.
- **Mejorar la experiencia del usuario:** Diseñar y ajustar una Página web para la navegación cómoda y que proporcione información útil y mejore la satisfacción general de los turistas al visitar Cochabamba.

1.3. ALCANCES

- **Implementación del Chatbot:** El proyecto se centrará en la implementación de un chatbot capaz de proporcionar información

turística sobre el departamento de Cochabamba. Se abordarán aspectos técnicos, como el diseño de la interfaz, el entrenamiento de la inteligencia artificial para el chatbot.

- **Web Scraping:** Se llevará a cabo la implementación de un sistema de web scraping para recopilar información actualizada sobre los sitios turísticos en Cochabamba. Esto permitirá que el chatbot ofrezca datos precisos y actualizados a los usuarios.
- **Investigación de Plataformas y Técnicas:** Se investigarán las plataformas y técnicas más adecuadas para el desarrollo del chatbot, incluyendo opciones de lenguajes de programación, marcos de desarrollo.
- **Diseño de Interfaz Amigable:** Se diseñará una interfaz de usuario amigable que permita a los usuarios interactuar de manera efectiva con el chatbot y obtener información sobre los sitios turísticos. Esto incluye el diseño de una interfaz de chat atractiva y funcional.

1.4. DELIMITACIONES DEL TEMA

1.4.1. DELIMITACION TEMATICA

El proyecto está enfocado al área del turismo, brindando información y para mejorar la información del cliente y sus distintas consultas, facilitar la información y las preguntas más frecuentes que se tengan.

- **Alcance Geográfico:** El proyecto se enfoca exclusivamente en el departamento de Cochabamba, Bolivia. No se consideran destinos turísticos fuera de esta región.
- **Idioma:** El chatbot se implementará en chat gpt la cual cubre dos idiomas (español e inglés).
- **Actualización de Datos:** La actualización de la información turística a través del sistema de web scraping estará limitada a fuentes públicas y

accesibles en línea. No se incluirán sistemas de retroalimentación en tiempo real por parte de turistas.

- **Recopilación de Datos:** El proyecto se centrará en la recopilación de información general sobre los sitios turísticos, como descripciones, horarios y ubicaciones. No se incluirán datos personales de los usuarios.
- **Tiempo de Atención:** Se busca mejorar la atención al cliente, el chatbot estará disponible las 24 horas.

1.5. JUSTIFICACION

1.5.1 Justificación Social

La implementación del proyecto propuesto tendrá un impacto significativo en el sector turístico del departamento al proporcionar un sistema avanzado de geolocalización y navegación. A través del uso de técnicas de georreferenciación y algoritmos de optimización de rutas, el chatbot permitirá a los usuarios acceder a información precisa y detallada sobre los sitios turísticos de interés. Esta información comprenderá datos geográficos, distancias y tiempos de viaje estimados para cada destino. La integración de estos datos permitirá a los usuarios planificar de manera efectiva sus itinerarios, tomar decisiones informadas y minimizar posibles contratiempos durante sus desplazamientos.

1.5.2 Justificación técnica

Al momento de implementar un proyecto, es fundamental realizar una evaluación exhaustiva de las opciones tecnológicas disponibles. Esto implica analizar y seleccionar las herramientas más adecuadas para abordar el problema planteado y lograr una resolución exitosa. El objetivo es identificar las tecnologías y metodologías que permitan desarrollar el sistema de manera eficiente y efectiva, garantizando su viabilidad y capacidad para enfrentar los desafíos específicos del proyecto. Además, se deben tener en cuenta aspectos como la capacidad de escalabilidad, la interoperabilidad y la disponibilidad de

los recursos tecnológicos necesarios para la implementación exitosa del proyecto.

- **Selección de herramientas:** Se realizará una evaluación exhaustiva de las opciones disponibles en el mercado para identificar y seleccionar las herramientas más adecuadas y actualizadas para el desarrollo del sistema de web scraping y el chatbot. Se tendrán en cuenta aspectos como la capacidad de extracción de datos, la integración con otras tecnologías, la facilidad de uso y la compatibilidad con el entorno tecnológico del proyecto.
- **Mantenimiento y actualización:** Se establecerá un plan de mantenimiento y actualización continua de las herramientas utilizadas, con el fin de asegurar su funcionamiento óptimo a lo largo del tiempo. Esto incluirá monitoreo de actualizaciones, parches de seguridad, optimización del rendimiento y la adopción de nuevas funcionalidades y mejoras que puedan surgir en el panorama tecnológico.

1.5.3 Justificación Económica

El presupuesto de este proyecto contempla principalmente los costos asociados con la adquisición de dominios y el acceso completo a la tecnología Orimon.ai. La inversión en estos elementos es esencial para garantizar un entorno digital sostenible y seguro, proporcionando la estabilidad necesaria para el funcionamiento del proyecto.

CAPÍTULO II

MARCO TEÓRICO

2.1 Marco Teórico

2.1.1 Elementos Principales

2.1.1.1 Chatbots Inteligentes

Los chatbots inteligentes han experimentado una notable evolución desde los primeros sistemas como ELIZA en la década de 1960, diseñado para simular una conversación terapéutica básica mediante procesamiento de lenguaje natural (NLP). ELIZA utilizaba patrones de coincidencia de texto y reglas predefinidas para generar respuestas simples, lo que limitaba su capacidad para mantener una conversación coherente y relevante. Con el tiempo, se desarrollaron técnicas más avanzadas de NLP, como los modelos de Markov y las máquinas de soporte vectorial (SVM), que mejoraron la capacidad de los chatbots para analizar y responder a las entradas del usuario de manera más sofisticada.

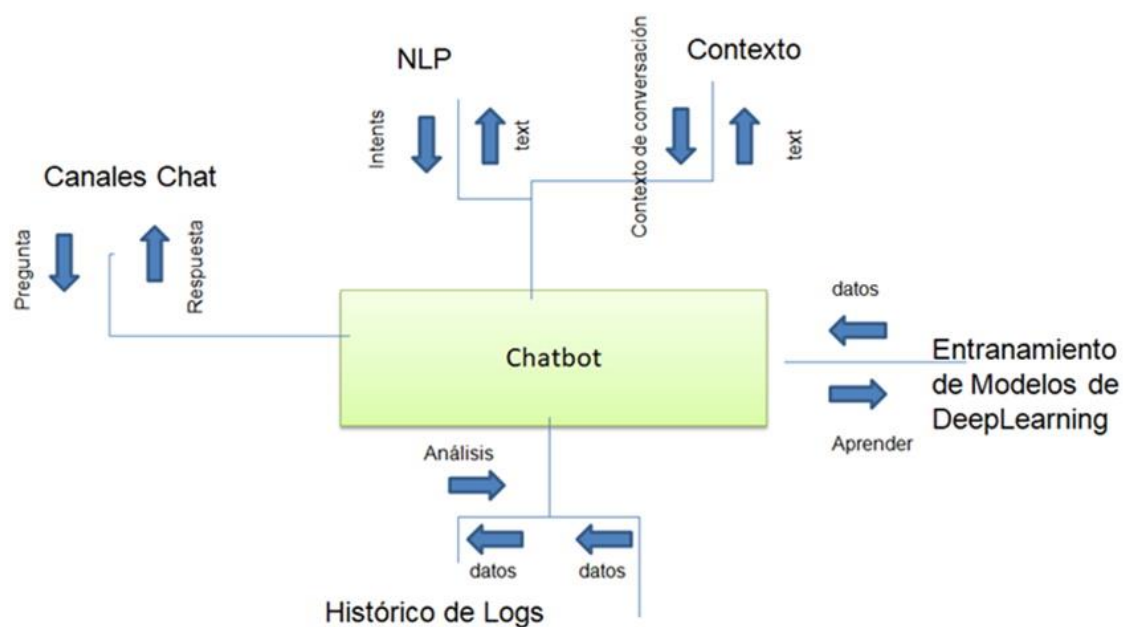
La implementación de arquitecturas avanzadas de inteligencia artificial, como las redes neuronales recurrentes (RNN) y los modelos de transformadores, ha permitido a los chatbots modernos generar y entender lenguaje natural con una precisión sin precedentes. Las RNN, con su capacidad para procesar secuencias de datos temporales, mejoraron significativamente la comprensión contextual de los chatbots. Sin embargo, los modelos de transformadores, como BERT y GPT, han revolucionado el campo al permitir el procesamiento paralelo y el análisis de grandes cantidades de datos textuales, mejorando la coherencia y relevancia de las respuestas generadas.

Actualmente, plataformas avanzadas como GPT-4 de OpenAI se utilizan en aplicaciones diversas, incluyendo la automatización de atención al cliente, la asistencia virtual personalizada y la generación de contenido. Estos chatbots son capaces de manejar tareas complejas, como la resolución de problemas técnicos, la recomendación de productos y la creación de textos creativos, gracias a su capacidad para entender contextos amplios y generar respuestas humanas. Además, la integración de técnicas de aprendizaje profundo y el

acceso a grandes bases de datos de texto han permitido que estos chatbots sean cada vez más precisos y útiles en una variedad de sectores.

Figura 1

Estructura de un chatbot Inteligente



Nota: Instituto Tecnológico de ARagón. (2023, 19 octubre). *Chatbots o asistentes conversacionales basados en Inteligencia Artificial y técnicas de Procesamiento de Lenguaje Natural - ITA*. ITA. <https://www.ita.es/blog/chatbots-o-asistentes-conversacionales-basados-en-inteligencia-artificial-y-tecnicas-de-procesamiento-de-lenguaje-natural/>

2.1.1.2 Web Scraping

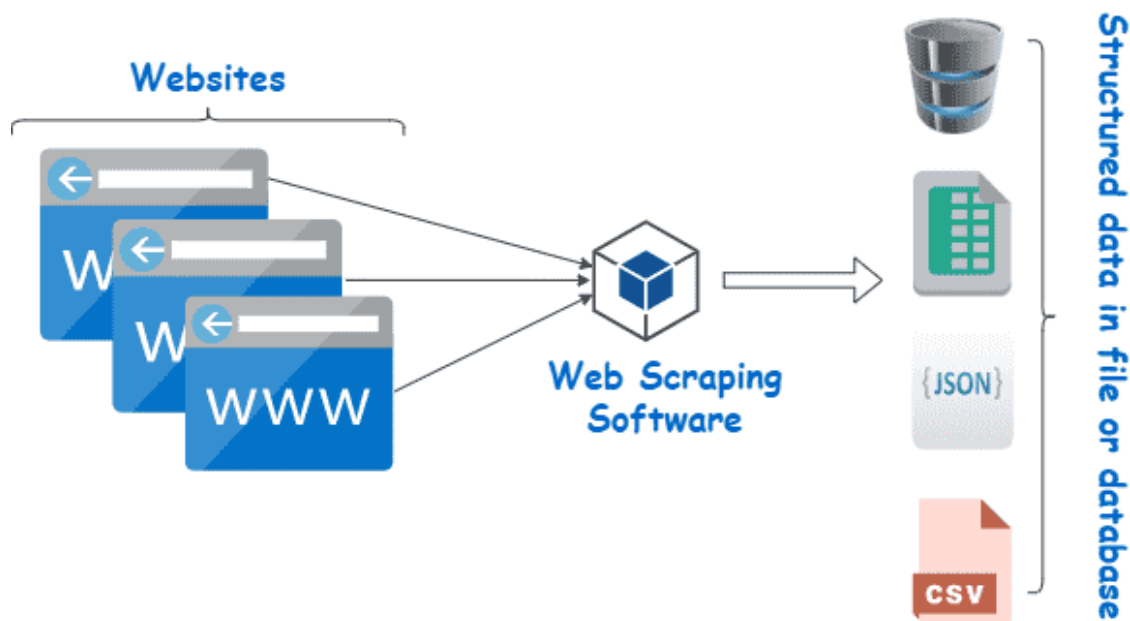
El web scraping es una técnica de extracción automatizada de datos de sitios web, esencial para la recopilación masiva de información estructurada. Los primeros motores de búsqueda en la década de 1990 empleaban técnicas rudimentarias de web scraping para indexar contenido web, utilizando scripts simples para recuperar y almacenar datos. Con el tiempo, la complejidad y la diversidad de los sitios web aumentaron, lo que llevó al desarrollo de herramientas más sofisticadas para manejar la estructura HTML y la dinámica de las páginas web modernas.

Hoy en día, el web scraping ha evolucionado gracias a bibliotecas y frameworks avanzados como BeautifulSoup, Scrapy y Puppeteer. BeautifulSoup facilita la navegación y manipulación del árbol de sintaxis del HTML, permitiendo a los desarrolladores extraer datos específicos de manera eficiente. Scrapy, por otro lado, es un framework de scraping más robusto que permite la gestión de proyectos completos de extracción de datos, ofreciendo capacidades de rastreo web, manejo de solicitudes y almacenamiento de datos. Puppeteer, una herramienta de automatización basada en Node.js, permite la navegación de páginas web complejas mediante la simulación de un navegador, lo que es útil para extraer datos de sitios web dinámicos que dependen de JavaScript.

Sin embargo, el web scraping plantea desafíos legales y éticos, ya que la propiedad intelectual y las políticas de uso de datos deben ser cuidadosamente consideradas para evitar infracciones y garantizar el uso ético de la información. Muchos sitios web implementan medidas técnicas para evitar el scraping, como el uso de CAPTCHAs y la detección de bots, lo que añade una capa adicional de complejidad para los desarrolladores de scraping. Además, es crucial que las prácticas de scraping cumplan con las leyes de protección de datos, como el Reglamento General de Protección de Datos (GDPR) en Europa, para evitar sanciones legales y preservar la privacidad de los usuarios.

Figura 2

¿Qué es web scraping?



Nota: Kinsta. (2022, 19 diciembre). *¿Qué es el web scraping? Cómo extraer legalmente el contenido de la web*. Kinsta®. <https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/>

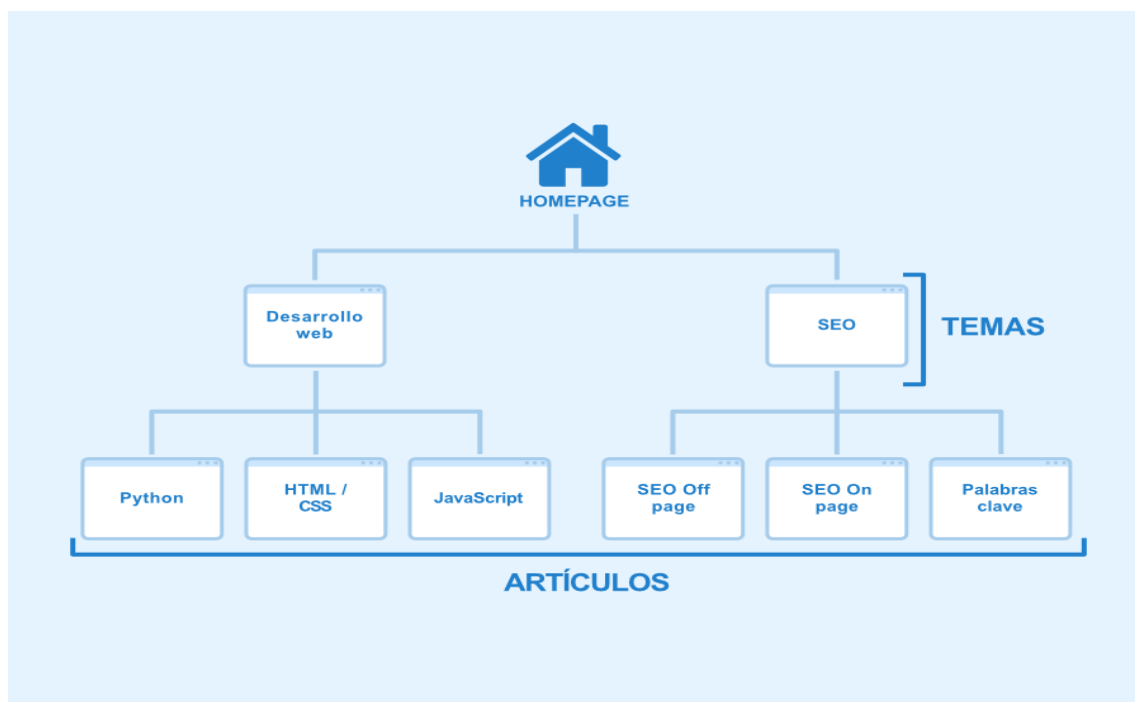
2.1.1.3 Páginas Web

El desarrollo de páginas web ha avanzado desde simples documentos HTML hasta sofisticadas aplicaciones interactivas impulsadas por tecnologías como HTML5, CSS3 y JavaScript. En sus inicios, las páginas web eran estáticas y se componían principalmente de texto e imágenes con enlaces hipertextuales. Con la introducción de CSS, los desarrolladores pudieron separar el contenido de la presentación, mejorando el diseño y la accesibilidad de las páginas web. La evolución continuó con la incorporación de JavaScript, que permitió la creación de contenido dinámico y la interacción en tiempo real, sentando las bases para las aplicaciones web modernas.

Frameworks modernos como React, Angular y Vue.js han revolucionado el front-end, permitiendo la creación de interfaces de usuario dinámicas y reactivas. React, desarrollado por Facebook, facilita el desarrollo de interfaces de usuario componibles mediante el uso de componentes reutilizables. Angular, una plataforma de desarrollo de aplicaciones web mantenida por Google, proporciona un framework completo para construir aplicaciones web de una sola página (SPA) con capacidades de enlace de datos bidireccional e inyección de dependencias. Vue.js, conocido por su enfoque progresivo, permite a los desarrolladores adoptar gradualmente su ecosistema para mejorar partes específicas de un proyecto existente o para desarrollar aplicaciones complejas desde cero.

En paralelo, metodologías como DevOps y desarrollo ágil han optimizado el ciclo de vida del desarrollo de software, facilitando despliegues continuos y mejorando la calidad del código. DevOps integra el desarrollo y las operaciones para automatizar y mejorar la eficiencia del flujo de trabajo, mientras que el desarrollo ágil se centra en la entrega iterativa y la colaboración entre equipos. Además, la integración de inteligencia artificial y aprendizaje automático está personalizando la experiencia del usuario, adaptando el contenido y la interacción según el comportamiento y las preferencias del usuario. Estas innovaciones están redefiniendo el rol de las páginas web como plataformas multifuncionales que van más allá de la simple presentación de información, convirtiéndose en aplicaciones complejas que interactúan de manera inteligente con los usuarios.

Figura 3

Estructura web

Nota: Seobility. (s. f.). *¿Qué es la Estructura Web de un sitio?* - Seobility Wiki.
https://www.seobility.net/es/wiki/Estructura_web

2.1.2 Técnicas Recolección Información

2.1.2.1 Observación Directa

La Observación Directa es una técnica de recolección de datos que implica observar directamente el comportamiento de personas, eventos o situaciones sin intervenir en ellos. Según Hammersley y Atkinson (2007), se trata de "la recogida sistemática de datos mediante la observación de las personas en su ambiente natural, tratando de respetar lo más posible la normalidad de su conducta". Es una técnica de investigación cuantitativa que permite entender el comportamiento de cierto objeto o entender la estructura de su comportamiento sin la necesidad de tener una comunicación verbal. Ahora una herramienta tan eficiente tiene ciertos puntos a favor y en contra a continuación algunas:

Ventajas:

- Datos de comportamientos reales: Permite obtener datos auténticos y no influenciados por la percepción del observado.
- Estudio de interacciones no verbales: Es útil para estudiar interacciones no verbales y contextos sociales que pueden no ser captados por otros métodos.
- Visión detallada: Proporciona una visión detallada de los procesos y comportamientos observados, permitiendo captar aspectos que podrían pasarse por alto con otros métodos.

Desventajas:

- Posible invasión de privacidad: Puede ser percibido como invasivo o poco ético en algunos contextos, especialmente si no se obtiene el consentimiento adecuado de los observados.
- Influencia del observador: Los observadores pueden influir inadvertidamente en el comportamiento de las personas observadas, alterando la naturalidad de sus acciones.
- Costoso y requiere tiempo: Puede ser un método costoso y que requiere mucho tiempo para la recolección de datos, especialmente si se necesita observar durante largos períodos o en múltiples ubicaciones.

2.1.2.2. Web Scraping

La extracción de información automatizada de páginas web es un proceso mediante el cual se utilizan bots para analizar y extraer datos de manera estructurada a partir del contenido HTML. Según Choudhary et al. (2019), este

enfoque permite la recopilación eficiente de grandes volúmenes de datos de forma rápida y automatizada, facilitando la obtención continua de información actualizada en tiempo real. Este método es particularmente útil cuando las fuentes de datos no ofrecen APIs o interfaces programáticas accesibles.

Ventajas:

- Recopilación eficiente de datos: Permite obtener grandes cantidades de datos de manera rápida y automatizada.
- Actualización continua: Facilita la obtención de información actualizada en tiempo real.
- Acceso a fuentes sin API: Útil para obtener datos de fuentes que no proporcionan APIs o interfaces programáticas.

Desventajas:

- Bloqueos de seguridad: Puede enfrentar bloqueos por parte de medidas de seguridad implementadas en los sitios web.
- Mantenimiento requerido: Requiere mantenimiento constante para ajustarse a cambios en el diseño y estructura de los sitios web.
- Datos no estructurados: Los datos extraídos pueden no estar limpios o estructurados, lo que demanda procesamiento adicional.

2.1.2.3 Procesamiento del lenguaje natural (NLP)

El Procesamiento del Lenguaje Natural (NLP) es un campo de la inteligencia artificial que se enfoca en la interacción entre las computadoras y el lenguaje humano, permitiendo a las máquinas procesar, entender y generar lenguaje humano de manera efectiva. Como señala Jurafsky y Martin (2020), "NLP se ocupa de la manipulación computacional del lenguaje natural (es decir, texto) por parte de programas de computadora".

Ventajas:

- Permite una interacción más natural y personalizada con los usuarios.
- Facilita la comprensión y generación de texto en sistemas conversacionales.
- Es útil para analizar grandes cantidades de texto y extraer información relevante.

Desventajas:

- Puede tener dificultades con lenguaje no estructurado o ambiguo.
- Requiere modelos de aprendizaje automático entrenados y ajustados para mejorar la precisión.
- La interpretación del contexto y la intención del usuario puede ser compleja.

2.1.3 Metodologías de Desarrollo**2.1.3.1 Metodología Extreme Programming (XP)**

Extreme Programming (XP) es una metodología ágil que facilita el desarrollo rápido y de alta calidad de software, centrándose en la entrega de valor en el menor tiempo posible. Sus pilares fundamentales son el costo, el tiempo, la calidad y el alcance. Esta metodología promueve la interacción continua entre desarrolladores y stakeholders, mejorando la simplicidad y la calidad del desarrollo del software.

2.1.3.2 Ciclo de Vida de XP:

- a. Planificación:** En esta fase, los desarrolladores y el cliente se reúnen para definir las funcionalidades a implementar, las cuales se reflejan en las historias de usuario. También se exploran las tecnologías y las arquitecturas que se utilizarán en el sistema. Se establece la duración de los hitos del proyecto.
- b. Diseño:** XP promueve diseños simples y claros, enfocados en la simplicidad y la resolución de problemas. El objetivo es implementar únicamente las funcionalidades necesarias, evitando añadir

características superfluas que puedan prolongar el tiempo de desarrollo. El enfoque es encontrar soluciones de la manera más eficiente posible.

- c. **Desarrollo:** Durante esta etapa, la participación activa del usuario es crucial. No basta con las historias de usuario; el cliente debe estar disponible para proporcionar información directa y aclarar cualquier malentendido de los requisitos, lo que previene problemas que en otras metodologías podrían detectarse demasiado tarde y ser más difíciles de corregir.
- d. **Pruebas:** Las pruebas comienzan en la etapa de planificación y continúan hasta la entrega final del software. Los desarrolladores crean pruebas unitarias para verificar cada componente del código, mientras que un equipo de control de calidad realiza pruebas de aceptación para asegurar que el software cumpla con las expectativas y necesidades del cliente.

Figura 4

Ciclo de vida programación extrema (XP)



Nota. Líderes en formación tecnológica, reskilling y upskilling | OpenWebinars.
(s. f.). OpenWebinars.net. <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

2.1.4 Plataformas de desarrollo

2.1.4.1. Java Script

JavaScript es un lenguaje de programación interpretado y orientado a objetos, basado en prototipos, ampliamente utilizado en el desarrollo de aplicaciones web. Originalmente desarrollado por Netscape, JavaScript ha evolucionado significativamente para convertirse en una herramienta esencial para el desarrollo front-end, permitiendo a los desarrolladores crear experiencias interactivas y dinámicas en los navegadores web. Este lenguaje es conocido por su capacidad de manipular el Document Object Model (DOM), facilitando la actualización y gestión del contenido de las páginas web sin necesidad de recargar la página completa. Además, su integración con HTML y CSS permite a los desarrolladores construir interfaces de usuario sofisticadas y responsivas. Además de su uso en el lado del cliente, JavaScript también ha ganado prominencia en el desarrollo del lado del servidor gracias a plataformas como Node.js. Esta expansión ha permitido a los desarrolladores utilizar JavaScript para construir aplicaciones completas, desde la interfaz de usuario hasta la lógica del servidor y la gestión de bases de datos. La flexibilidad y la extensa comunidad de desarrolladores que soportan JavaScript han contribuido a la creación de una vasta cantidad de bibliotecas y frameworks, como React, Angular y Vue.js, que facilitan y aceleran el proceso de desarrollo. En resumen, JavaScript es un lenguaje versátil y potente, fundamental en la ingeniería de sistemas para el desarrollo de aplicaciones web modernas y eficientes.

Figura 5

Logo Java Script



2.1.4.2. HTML

HTML (HyperText Markup Language) es el lenguaje estándar utilizado para crear y estructurar el contenido en la web. Desarrollado originalmente por Tim Berners-Lee en 1991, HTML permite a los desarrolladores definir la estructura de una página web mediante el uso de elementos y etiquetas que representan diferentes tipos de contenido y componentes, como encabezados, párrafos, enlaces, imágenes, y formularios. HTML es fundamental para la creación de documentos web que pueden ser interpretados y visualizados por navegadores web, proporcionando la base sobre la cual se aplican estilos (mediante CSS) y comportamientos dinámicos (mediante JavaScript).

En el contexto de la ingeniería de sistemas, HTML es crucial para el desarrollo front-end de aplicaciones web, permitiendo la definición clara y semántica del contenido, lo cual facilita tanto el desarrollo como el mantenimiento del código. La semántica de HTML mejora la accesibilidad y el SEO (Search Engine Optimization), asegurando que el contenido web sea más comprensible para los motores de búsqueda y accesible para los usuarios con discapacidades. Además, HTML5, la versión más reciente del lenguaje, introduce nuevas funcionalidades y APIs que permiten una mejor integración con multimedia, gráficos y aplicaciones interactivas, expandiendo significativamente las capacidades del desarrollo web moderno.

Figura 6

Logo HTML



2.1.4.3. CSS

Cascading Style Sheets (CSS) es un lenguaje utilizado para definir el estilo y el formato visual de los documentos HTML y XML. CSS permite a los desarrolladores separar la estructura de un documento web de su presentación, controlando aspectos como el diseño, los colores y las fuentes. Además, CSS facilita la creación de sitios web adaptables a diferentes dispositivos y tamaños de pantalla mediante el uso de reglas y selectores que aplican estilos de forma jerárquica y cascada (W3C, 2017).

Figura 7

Logo CSS



2.1.4.4. MongoDB

Mongo DB es una plataforma de desarrollo de bases de datos no relacionales que se distingue por su enfoque orientado a documentos y su estructura de datos flexible. Permite a los desarrolladores almacenar información en un formato similar a JSON (BSON en su versión interna), ideal para manejar datos semiestructurados y no estructurados. Esta flexibilidad es especialmente beneficiosa para aplicaciones modernas donde los datos son heterogéneos y pueden evolucionar sin la necesidad de ajustar esquemas rígidos, como ocurre en las bases de datos relacionales. MongoDB facilita la creación, implementación y escalabilidad de aplicaciones mediante herramientas

avanzadas, destacándose MongoDB Atlas , una solución de base de datos en la nube que simplifica la configuración y el escalado en plataformas como AWS, Azure y Google Cloud.

El ecosistema de MongoDB incluye controladores oficiales para múltiples lenguajes de programación como JavaScript , Python y Java , permitiendo una fácil integración con las aplicaciones. Además, ofrece soporte para consultas avanzadas, índices multidimensionales y capacidades de agregación, lo que lo hace altamente eficaz para proyectos que requieren acceso rápido y flexible a grandes volúmenes de datos. En resumen, MongoDB no solo proporciona una base de datos potente y adaptable, sino que también incluye un conjunto completo de herramientas y servicios que potencian el desarrollo y gestión de aplicaciones modernas.

Figura 8

Logo de MongoDB



2.1.4.5. Elastisearch

Elasticsearch es una plataforma de búsqueda y análisis de datos en tiempo real, conocida por su capacidad para manejar grandes volúmenes de información de manera rápida y eficiente. Basada en el motor de búsqueda Apache Lucene , Elasticsearch permite a los usuarios indexar, buscar y analizar datos de forma escalable y distribuida. Es ampliamente utilizado en aplicaciones

que requieren búsqueda avanzada, como sistemas de monitoreo, análisis de registros, comercio electrónico y servicios de inteligencia empresarial.

Uno de los principales beneficios de Elasticsearch es su capacidad para realizar búsquedas de texto completo, filtrados de datos y consultas analíticas complejas con alta velocidad. La plataforma permite la creación de índices inversos, una estructura de datos que optimiza la búsqueda rápida en grandes conjuntos de información. Además, soporta el almacenamiento de datos en documentos de tipo JSON , lo que facilita la integración con aplicaciones modernas y la gestión de datos semiestructurados.

Elasticsearch se integra de manera efectiva con otras herramientas de su ecosistema, como Kibana , que proporciona visualización de datos y creación de paneles interactivos, y Logstash , que permite la ingesta y transformación de datos desde múltiples fuentes. La escalabilidad horizontal es otra de sus fortalezas, ya que permite distribuir datos en Múltiples nodos, garantizando redundancia y alta disponibilidad. Gracias a su flexibilidad y rendimiento, Elasticsearch se ha convertido en una solución de referencia para proyectos que requieren capacidades de búsqueda y análisis en tiempo real.

Figura 9

Logo de Elastisearch



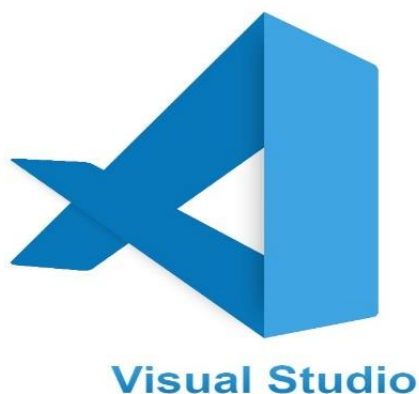
2.1.5. Herramientas

2.1.5.1. Visual studio Code

Visual Studio Code (VS Code) es un entorno de desarrollo integrado (IDE) de código abierto desarrollado por Microsoft. Es una herramienta ligera y gratuita que se adapta a una amplia variedad de lenguajes de programación y tecnologías. VS Code está diseñado para ser altamente personalizable y versátil, ofreciendo las siguientes características y ventajas:

- **Código abierto y gratuito:** Disponible para descarga y uso sin costo, con el código fuente accesible para la comunidad.
- **Soporte multiplataforma:** Compatible con Windows, macOS y Linux, lo que lo hace accesible para una amplia gama de desarrolladores.
- **Resaltado de sintaxis y autocompletado:** Proporciona resaltado de sintaxis para numerosos lenguajes de programación, además de funciones de autocompletado que mejoran la eficiencia en la escritura de código.
- **Integración con control de versiones:** Facilita la integración con sistemas de control de versiones como Git, simplificando así la gestión y seguimiento de cambios en el código fuente.
- **Depuración incorporada:** Ofrece herramientas de depuración integradas para ayudar a los desarrolladores a detectar y resolver problemas en su código.
- **Administración de paquetes y entornos virtuales:** Permite la gestión de paquetes y entornos virtuales para proyectos, lo cual es fundamental en el desarrollo de aplicaciones.

Figura 10

Logo Visual Studio Code**2.1.5.2. Orimon.ai**

- Orimon.ai es una herramienta avanzada de inteligencia artificial diseñada para crear y gestionar chatbots de manera eficiente y efectiva. Esta plataforma permite a las empresas y desarrolladores construir chatbots personalizados sin necesidad de tener conocimientos profundos en programación, gracias a su interfaz intuitiva y amigable. Orimon.ai se destaca por su capacidad para integrar múltiples canales de comunicación, como sitios web, aplicaciones móviles y redes sociales, lo que facilita la interacción con los usuarios en diferentes plataformas.

Ventajas:

- Interfaz Intuitiva: Permite a usuarios con poco conocimiento técnico crear y personalizar chatbots fácilmente.
- Multicanal: Soporta la integración con diversos canales de comunicación, facilitando una presencia omnicanal.
- Personalización Avanzada: Ofrece opciones de personalización detalladas para ajustar el chatbot a las necesidades específicas del negocio.

- Escalabilidad: Puede manejar un gran volumen de interacciones, lo que es ideal para empresas en crecimiento.

Desventajas:

- Costo: Las funcionalidades avanzadas y el soporte pueden tener un costo elevado, lo que podría no ser accesible para pequeñas empresas o startups.
- Dependencia de la Plataforma: La personalización y funcionalidad del chatbot están limitadas a lo que ofrece la plataforma, lo que puede restringir la flexibilidad para desarrollos muy específicos.
- Curva de Aprendizaje Inicial: Aunque la plataforma es intuitiva, los usuarios nuevos pueden necesitar tiempo para familiarizarse completamente con todas las funcionalidades y opciones de personalización.

Figura 11

Logo de Orimon.ai



2.1.6. Formatos de datos

2.1.6.1. JSON

JSON (JavaScript Object Notation) es un formato de intercambio de datos extremadamente popular debido a su simplicidad, legibilidad y amplia compatibilidad con múltiples lenguajes de programación. Basado en texto,

JSON representa los datos como pares clave-valor y estructuras de arreglos, lo que lo hace intuitivo tanto para desarrolladores como para sistemas. Su estructura es directamente compatible con los objetos de JavaScript, pero también es ampliamente aceptada en lenguajes como Python, Java, C#, Ruby, entre otros. JSON se utiliza combinado en aplicaciones web y APIs RESTful para el intercambio de información entre el cliente y el servidor, especialmente en arquitecturas basadas en microservicios, debido a su

A pesar de sus ventajas, JSON presenta ciertas limitaciones desde una perspectiva técnica. Uno de los principales inconvenientes es su falta de soporte nativo para tipos de datos complejos, como fechas o valores binarios, lo que requiere soluciones de serialización o formatos adicionales (como ISO 8601 para fechas). Además, como es un formato textual, no puede garantizar la seguridad intrínseca contra vulnerabilidades como la inyección de código, lo que requiere un manejo adecuado en el proceso de deserialización. La ausencia de tipado explícito en la estructura de datos también puede generar problemas en sistemas grandes, donde la validación de datos y la consistencia entre los distintos servicios y aplicaciones resulta crucial. A pesar de estos inconvenientes, JSON sigue siendo uno de los formatos más utilizados en la transmisión de datos en aplicaciones modernas debido a su flexibilidad, facilidad de implementación.

Figura 12

Logo de JSON

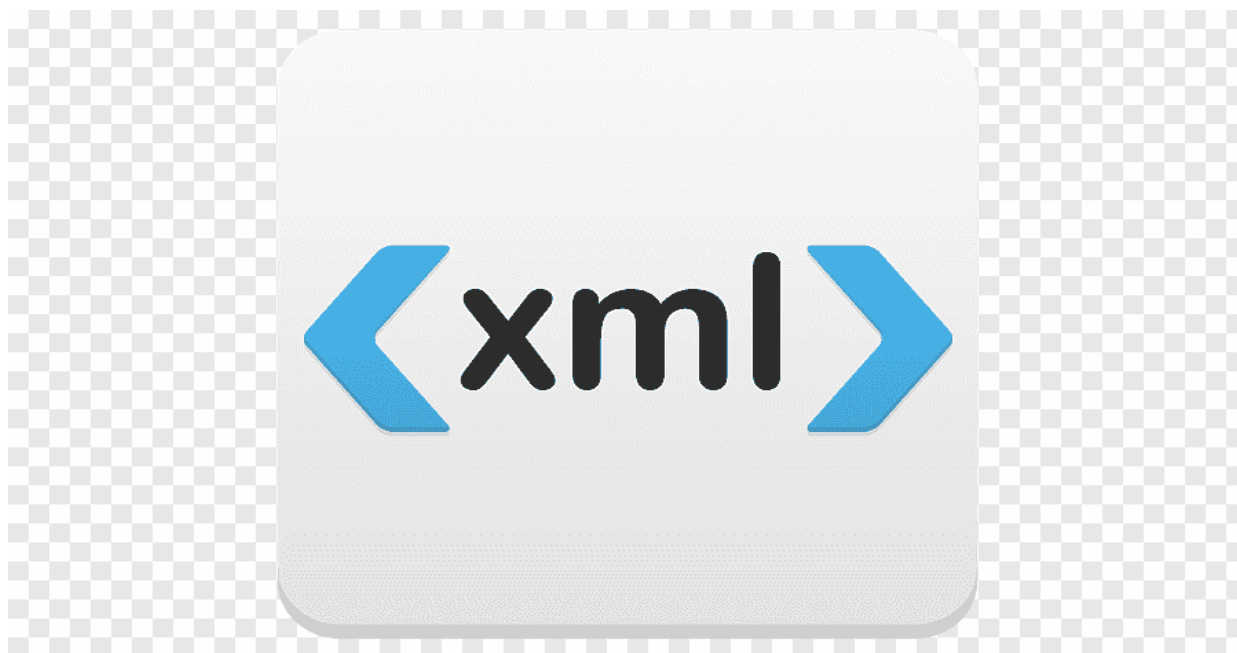


2.1.6.2. XML

XML (Extensible Markup Language) es un formato de intercambio de datos que permite representar de manera estructurada y jerárquica información en forma de texto. A diferencia de JSON, XML se basa en un modelo de etiquetas (tags) para definir los elementos y sus relaciones. Este formato es extremadamente flexible, ya que permite crear etiquetas personalizadas para representar cualquier tipo de datos, lo que lo hace adecuado para una amplia variedad de aplicaciones, especialmente en entornos empresariales, sistemas de información complejos y estándares de interoperabilidad como SOAP y RSS. La capacidad de XML para representar estructuras jerárquicas permite crear documentos complejos que pueden incluir metadatos, atributos y relaciones entre diferentes elementos, lo que lo hace ideal para aplicaciones que requieren un alto grado de complejidad en la representación.

Desde una perspectiva técnica, XML ofrece ventajas significativas en cuanto a su capacidad de validación y esquemas, como DTD (Document Type Definición) y XML Schema, que proporcionan una forma de definir la estructura y los tipos de datos de los documentos XML. Esto garantiza una validación rigurosa de los datos, lo que es fundamental en aplicaciones empresariales y sistemas que requieren un alto grado de consistencia y control sobre la información. Sin embargo, XML presenta también varias desventajas: es más detallado que JSON, lo que puede llevar a un mayor consumo de ancho de banda y tiempo de procesamiento en aplicaciones web. Además, su estructura jerárquica, aunque flexible, puede resultar más compleja de manipular en comparación con formatos más simples como JSON, lo que requiere herramientas adicionales para procesar y analizar los datos. También, aunque XML ofrece una buena capacidad de validación, su sintaxis puede resultar más propensa a errores de interpretación debido a su mayor complejidad y rigor en la definición de etiquetas y atributos. A pesar de estos desafíos, XML sigue siendo ampliamente utilizado en aplicaciones que requieren robustez, validación avanzada y una estructura.

Figura 13

Logo de JSON**2.1.6.3. CSV**

El formato CSV (Valores separados por comas) es un formato de datos en texto plano muy utilizado para representar y almacenar información tabular, donde cada línea corresponde a un registro y los campos dentro de cada fila están separados por un delimitador, generalmente una coma, aunque También pueden utilizarse otros caracteres como el punto y coma. Es muy popular en aplicaciones como hojas de cálculo, bases de datos y plataformas de análisis debido a su estructura simple y su compatibilidad con diversos lenguajes y herramientas. CSV destaca por su ligereza y facilidad de manejo, siendo eficiente para grandes volúmenes de datos sin estructuras complejas. Sin embargo, carece de un esquema formal y no soporta de manera adecuada datos jerárquicos o anidados, lo que lo limita frente a otros formatos como JSON o XML. A pesar de estas restricciones, sigue siendo ampliamente adoptado por su simplicidad y efectividad en el manejo de datos tabulares.

Figura 14

Logo de CSV**2.1.7 API de integración****2.1.7.1. OpenAi**

Orimon.ai integra la API de OpenAI, especializada en procesamiento de lenguaje natural (NLP), lo que permite la interacción efectiva con los usuarios mediante la comprensión y generación de texto. Esta API es fundamental para el chatbot, ya que facilita la interpretación de consultas, la extracción de intenciones y la creación de respuestas contextuales y coherentes.

La integración de una API que emplea técnicas avanzadas de procesamiento del lenguaje natural (NLP) será clave para mejorar la precisión y efectividad del chatbot en la interpretación de las consultas de los usuarios. Esta integración permitirá que el sistema comprenda las preguntas de manera más eficiente, independientemente de cómo estén formuladas, lo que nos habilita a proporcionar respuestas más personalizadas y contextualmente precisas. Este

enfoque elevará significativamente la experiencia del usuario, asegurando una interacción más intuitiva y fluida con el sistema.

Además, la API proporcionará un acceso optimizado a modelos que no solo comprenden el contexto de las consultas, sino que generan respuestas coherentes y altamente relevantes. Dado el entorno turístico en el que operamos, es crucial que estas respuestas se ajusten dinámicamente a la información más reciente sobre destinos, rutas y actividades, manteniendo al sistema alineado con las necesidades cambiantes de los usuarios.

Otro factor decisivo es la capacidad de integrar esta API con técnicas de web scraping para la extracción en tiempo real de datos actualizados desde diversas fuentes. Esta funcionalidad nos garantizará que el chatbot ofrecerá siempre información de primera mano sobre sitios turísticos, eventos y novedades locales, lo que lo convierte en una herramienta indispensable para los turistas que confían en nuestra plataforma para planificar sus viajes.

En términos de escalabilidad y flexibilidad, esta API nos permite agregar nuevas funcionalidades sin alterar la arquitectura central del sistema. Esto incluye la incorporación de recomendaciones personalizadas, análisis avanzados de tendencias de usuarios y futuras expansiones tecnológicas que puedan surgir. La modularidad del sistema facilita la adopción de nuevos modelos de lenguaje o la integración con APIs adicionales que complementan la oferta de servicios.

Figura 15

Logo OpenAi

CAPÍTULO III

MARCO PRÁCTICO

3.1 Análisis de Requerimientos del Sistema

3.1.1 Proceso Actual

Las páginas web del Departamento de Cochabamba y las agencias de turismo han avanzado en diseño y usabilidad, con interfaces responsivas adaptadas para diversos dispositivos y navegación intuitiva mediante menús y enlaces claros. El contenido se actualiza regularmente para reflejar eventos, atracciones y servicios, y se ofrece en múltiples idiomas para atender a una audiencia global. La interactividad se ha mejorado con mapas interactivos y secciones para comentarios y opiniones de los usuarios. Además, estas páginas están bien integradas con las redes sociales, facilitando la conexión con cuentas oficiales y permitiendo compartir contenido fácilmente. En términos de seguridad y privacidad, se utilizan conexiones seguras (HTTPS) y se mantienen políticas claras sobre el uso de datos personales.

Sin embargo, estas páginas enfrentan varias dificultades. El mantenimiento y actualización constante son retos debido a recursos limitados, lo que puede resultar en información desactualizada. Problemas técnicos y la variabilidad en la calidad del diseño afectan la accesibilidad y usabilidad. La seguridad es una preocupación continua, con amenazas cibernéticas que requieren mejoras constantes y cumplimiento de regulaciones de privacidad de datos. El multilingüismo también presenta desafíos, ya que las traducciones pueden ser imprecisas y no todo el contenido está disponible en múltiples idiomas, limitando así el alcance global.

Las agencias de turismo han modernizado su presencia en línea con sitios web atractivos y funcionales, y utilizan estrategias de SEO y SEM para optimización y marketing digital. Publican contenido digital relevante, como blogs, artículos, fotos y videos de alta calidad. La atención al cliente se mejora mediante chat en vivo y respuestas rápidas en redes sociales. Además, emplean técnicas de marketing digital como email marketing y publicidad en redes sociales. Las colaboraciones con hoteles y restaurantes, así como con influencers, ayudan a

ofrecer paquetes combinados y mejorar la visibilidad. También se implementan sistemas de gestión de reservas (RMS) y análisis de datos para optimización basada en el comportamiento de los clientes.

Las agencias de turismo enfrentan competencia global y la necesidad de diferenciarse en un mercado saturado. Los recursos financieros limitados dificultan la inversión en tecnología y marketing, y se requiere formación continua del personal en nuevas tecnologías y tendencias. La implementación de nuevas tecnologías puede ser desafiante, al igual que el mantenimiento técnico de los sistemas. La ciberseguridad y la protección de datos de los clientes son preocupaciones críticas. Además, la logística y coordinación de servicios entre múltiples proveedores, así como la gestión de reservas para evitar errores y duplicaciones, son tareas complejas que requieren una gestión eficiente.

3.1.2 Historias de Usuario:

A continuación, se muestra las historias de usuario empleadas según el requerimiento del usuario.

Tabla 1

Historia de Usuario 1

Historias de Usuario		
Nro 1.	Actor	Un visitante de la pagina web
Historias de usuario		Historia de Usuario para Información General de Sitios Turísticos
Descripción	Quiero obtener información general sobre los sitios turísticos en Cochabamba, Para decidir cuáles lugares visitar durante mi viaje.	
Observaciones		

Nota: Historia de usuario número uno, propuesto por el postulante.

Tabla 2

Historia de Usuario 2

Historias de Usuario		
Nro 2.	Actor	El administrador del sitio web
Historias de usuario		Historia de Usuario para Actualización Automática de Información
Descripción	<p>Quiero que la información sobre datos turísticos nuevos se actualice automáticamente,</p> <p>Para asegurarme de que los visitantes tengan acceso a los datos más recientes.</p>	
Observaciones		

Nota: Historia de usuario número dos, propuesta por el postulante.

Tabla 3

Historia de Usuario 3

Historias de Usuario		
Nro 3.	Actor	Un turista interesado en visitar sitios al aire libre
Historias de usuario		Historia de Usuario para Consulta de Clima
Descripción	<p>Quiero obtener información del clima actual y pronóstico en sitios turísticos,</p> <p>Para planificar mejor mis visitas y actividades.</p>	
Observaciones		

Nota: Historia de usuario número tres, propuesta por el postulante

Tabla 4

Historia de Usuario 4

Historias de Usuario		
Nro 4.	Actor	Un usuario con intereses específicos
Historias de usuario		Historia de Usuario para Recomendaciones Personalizadas
Descripcion	Quiero obtener recomendaciones de sitios turísticos personalizadas según mis intereses específicos, Para optimizar mi experiencia de viaje de acuerdo a mis preferencias personales.	
Observaciones		

Nota: Historia de usuario número cuatro, propuesta por el postulante

Tabla 5

Historia de Usuario 5

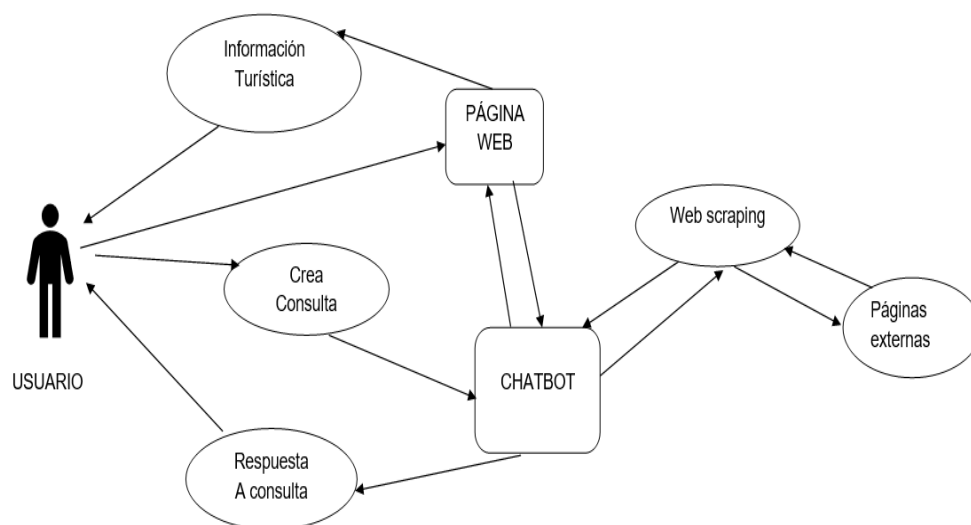
Historias de Usuario		
Nro 5.	Actor	Un administrador del sitio web
Historias de usuario		Historia de Usuario para Verificar Información Extraída
Descripcion	Quiero revisar la exactitud de la información extraída mediante web scraping, Para asegurar que los datos presentados en el sitio web sean correctos	
Observaciones		

Nota: Historia de usuario número cinco, propuesta por el postulante

3.1.3 Diagrama de Caso de Uso General Del Sistema

Figura 16

Diagrama Casos de uso del uso del programa



Nota: diagrama de casos de uso propuesto por el postulante

3.1.4 Plan de Iteraciones

Tabla 6

Iteraciones del proyecto

Iteraciones	Nro	Historias	Inicio	Final	Observaciones
Primera	1	Para Información General de Sitios Turísticos	14/03/2023	20/03/2023	
	2	Diseño de la Interfaz inicial	20/03/2023	23/03/2023	
	3	Desarrollo del primer prototipo del chatbot	23/03/2023	27/03/2023	Usando la tecnología Orimon.ai
Segunda	4	Revisión y retroalimentación	27/03/2023	28/03/2023	
	5	Para actualización automática de Información	28/03/2023	30/04/2023	
	6	Refinamiento del prototipo de web scrapping	30/03/2023	05/04/2023	Usando la tecnología html y php
Tercera	7	Diseño detallado	05/04/2023	09/04/2023	
	8	Revisión y retroalimentación	09/04/2023	10/04/2023	
	9	Consulta de Clima	10/04/2023	17/04/2023	Entrenar el chatbot con preguntas casuales
Cuarta	10	Desarrollo del sistema de web scraping	17/04/2023	22/04/2023	Usando la tecnología html
	11	Pruebas iniciales	22/04/2023	23/04/2023	Ver la facilidad de respuestas
	12	Revisión y retroalimentación	23/04/2023	24/04/2023	
Quinta		Recomendaciones Personalizadas	24/04/2023	26/04/2023	
		Pruebas de Usabilidad	26/04/2023	28/04/2023	
		Análisis de resultados	28/04/2023	30/04/2023	Se observa que responde de manera correcta a preguntas casuales y sacando información gracias a web scraping
Quinta		Revisión y retroalimentación	30/04/2023	08/05/2023	
		Verificar Información Extraída	08/05/2023	15/05/2023	Se obtiene información solicitada y se actualiza la información

3.2 Fase de Construcción

3.2.1 Módulo 1

3.2.1.1 Planificación

3.2.1.1.2 Observación directa

El desarrollo de una página web de turismo se centra en optimizar la experiencia del usuario y ofrecer contenido atractivo. En la fase de planificación, se recopilan datos sobre las necesidades e intereses del público objetivo.

El diseño debe priorizar la usabilidad y accesibilidad, utilizando navegación intuitiva y un diseño responsivo. La elección de colores adecuados y la inclusión de imágenes de calidad son esenciales para crear un entorno visual atractivo.

La recopilación de contenido relevante es un desafío constante, donde se asegura la precisión de la información sobre destinos y servicios mediante investigación exhaustiva. Se implementan técnicas de SEO para mejorar la visibilidad en motores de búsqueda, enfocándose en palabras clave del sector.

La inclusión de funcionalidades interactivas, como mapas y sistemas de reservas, facilita la planificación del viaje y mejora la conversión. La recolección de feedback del usuario, a través de pruebas de usabilidad, permite identificar áreas de mejora y realizar ajustes.

Finalmente, se establece un sistema de actualización continua que incorpora técnicas de web scraping y colaboración con proveedores locales, asegurando que la información esté alineada con las tendencias del sector.

3.2.2 Módulo 2

3.2.2.1 Diseño

3.2.2.1.1 Estructura de datos

En el desarrollo de chatbots conversacionales, la gestión y procesamiento de datos son fundamentales para garantizar una interacción fluida y efectiva con los usuarios, una plataforma avanzada de chatbots basada en inteligencia artificial, emplea diversas estructuras de datos que optimizan el rendimiento y la experiencia del usuario. Este documento describe las principales estructuras de datos utilizadas por Orimon.AI, enfocándose en su relevancia para la creación de chatbots eficientes.

- **Formatos de Datos Estructurados y Semiestructurados**

Los formatos de datos son esenciales para la comunicación entre el chatbot y las APIs externas. Orimon.AI utiliza principalmente este formato:

- JSON (JavaScript Object Notation) como el formato principal para la implementación de chatbots que requiere la integración de web scraping y la comunicación con API externas. Este formato es particularmente valioso debido a su ligereza y estructura simple, lo que lo hace fácil de leer y escribir tanto para humanos como para máquinas. La naturaleza de JSON permite que los datos se representen de manera clara a través de objetos y arreglos, simplificando el proceso de codificación y el intercambio de datos. Esta simplicidad no solo reduce la complejidad en el desarrollo, sino que también contribuye a una comunicación más ágil y fluida entre el chatbot y las plataformas con las que interactúa.

Gracias a estas propiedades, JSON se convierte en una herramienta ideal para el procesamiento de grandes volúmenes de datos, especialmente cuando se trata de la integración de información extraída mediante web scraping. La

capacidad de representar los datos de forma estructurada y coherente permite que Orimon procese y organice los datos de manera eficiente, garantizando que el chatbot pueda responder de forma rápida y precisa a las consultas de los usuarios. Además, el uso de JSON en el flujo de datos asegura que el chatbot pueda adaptarse a diferentes tipos de APIs y estructuras de información, proporcionando una solución flexible y robusta para la gestión de datos en tiempo real.

- **Modelos de Datos para Procesamiento de Lenguaje Natural (NLP)**

Los modelos de datos diseñados para el Procesamiento de Lenguaje Natural (NLP) son fundamentales para optimizar los chatbots que combinan capacidades de web scraping. Estos modelos permiten a los chatbots interpretar y generar lenguaje natural de forma precisa, así como extraer y procesar automáticamente información proveniente de páginas web. Esta integración convierte al chatbot en un sistema integral capaz de acceder a datos en tiempo real, analizarlos y responder eficientemente a las consultas de los usuarios, ofreciendo una experiencia interactiva y actualizada.

Para realizar estas tareas complejas, se utilizan modelos de PNL basados en arquitecturas avanzadas como los Transformers. Estas arquitecturas, que impulsan tecnologías modernas como GPT (Generative Pre-trained Transformer), son altamente eficaces para procesar grandes volúmenes de texto, extraer información clave y comprender el contexto de manera profunda. La integración de estos modelos con técnicas de web scraping permite al chatbot usar datos recién recopilados y procesarlos de inmediato, garantizando respuestas precisas y actualizadas. Este enfoque fortalece la capacidad del chatbot para proporcionar información relevante y flexible, respondiendo de forma eficiente a las demandas de los usuarios y mejorando la interacción en tiempo real.

- **Estructura de Bases de Datos No relacional**

1. **MongoDB** es una base de datos no relacional que se caracteriza por su capacidad para gestionar datos semiestructurados o no estructurados de manera eficiente. Utiliza un modelo basado en documentos, en el cual los datos se almacenan en un formato similar a JSON (BSON, en su implementación interna). Esta característica permite manejar contenido dinámico y variado extraído a través de web scraping, proporcionando flexibilidad para almacenar datos con estructuras que pueden cambiar con frecuencia. MongoDB es particularmente útil cuando se necesita almacenar y consultar grandes volúmenes de datos sin la rigidez de los esquemas predefinidos, lo que lo convierte en una elección ideal para proyectos de chatbots que requieren una integración rápida y adaptable con datos obtenidos de la web.
2. **Elasticsearch.** por otro lado, es una base de datos no relacional optimizada para la búsqueda y análisis de grandes volúmenes de datos en tiempo real. Esta herramienta se basa en el motor de búsqueda Lucene y es ampliamente reconocida por su capacidad de indexar documentos y realizar consultas de manera extremadamente rápida y eficiente. Elasticsearch es particularmente útil cuando se trata de integrar web scraping con un chatbot, ya que permite gestionar y buscar datos semiestructurados a gran escala. Al usar índices inversos y capacidades avanzadas de búsqueda, el sistema puede extraer y responder con información relevante en fracciones de segundo, mejorando la experiencia del usuario al ofrecer respuestas rápidas y contextualizadas. Su uso es esencial cuando el rendimiento en la búsqueda y la consulta de datos es una prioridad para el sistema.

Esta flexibilidad proporciona al chatbot la capacidad de acceder rápidamente a grandes volúmenes de datos, procesarlos de manera eficiente y manejar la

información en tiempo real. La integración de bases de datos no relacionales con el procesamiento de web scraping permite que el sistema gestione grandes conjuntos de datos heterogéneos y los ponga a disposición del chatbot para análisis y respuestas rápidas. Esto no solo mejora la eficiencia operativa del chatbot, sino que también optimiza la experiencia del usuario al ofrecer respuestas precisas, relevantes y actualizadas, adaptándose a las consultas en un entorno dinámico.

- **Flujos de Datos en Tiempo Real**

Esto permite el manejo de flujos de eventos en tiempo real, lo que es esencial para su funcionalidad de chat en vivo. Los mensajes y las interacciones de los usuarios se procesan instantáneamente, garantizando respuestas rápidas y precisas. Esta capacidad es fundamental para mejorar la experiencia del usuario y fomentar interacciones más fluidas.

La combinación de estructuras de datos robustas y eficientes en Orimon.AI no solo permite un manejo efectivo de la información, sino que también mejora la experiencia del usuario en la interacción con los chatbots. El uso de formatos como JSON y XML, junto con modelos avanzados de NLP y la integración de bases de datos NoSQL, contribuye a la flexibilidad, escalabilidad y eficiencia de la plataforma.

3.2.2.1.2 Modelo de Entrenamiento

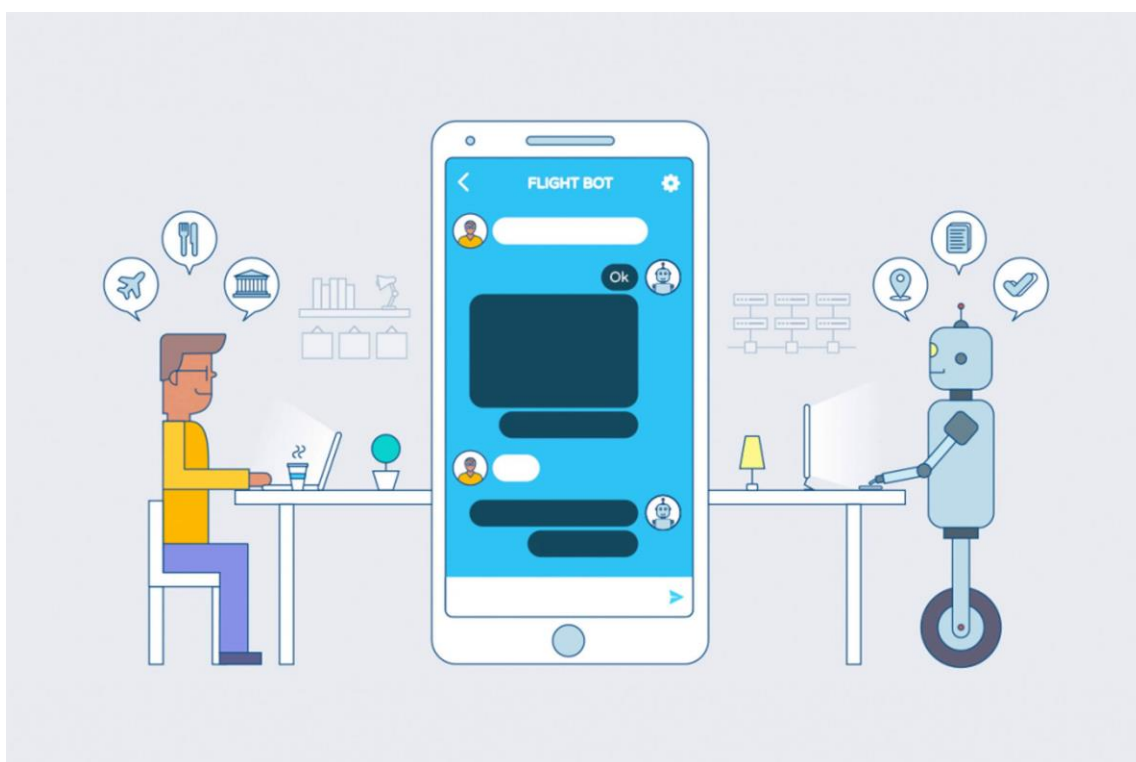
El proceso de entrenamiento del modelo de chatbot y su posterior despliegue se organiza en una serie de etapas estructuradas que permiten optimizar la experiencia del usuario, así como garantizar la precisión y actualización constante de los datos turísticos. Este enfoque está diseñado para proporcionar respuestas en tiempo real a los usuarios que buscan información detallada sobre lugares turísticos, actividades y eventos en Cochabamba, utilizando la capacidad de extracción dinámica de datos desde fuentes externas. A continuación, se detalla cada fase del proceso, adaptada a las necesidades del sistema propuesto.

1.Recolección de datos

El primer paso en el proceso es la recolección de datos, que constituye la base de conocimiento del sistema. Orimon.ai comienza extrayendo datos de diversas fuentes como sitios web turísticos, utilizando técnicas de web scraping. Estos datos incluyen información sobre lugares de interés turístico, actividades culturales, eventos, horarios de apertura, tarifas, descripciones de servicios y opiniones de usuarios. Esta extracción masiva de datos permite que el sistema mantenga información actualizada, ofreciendo respuestas oportunas y confiables a los usuarios que buscan recomendaciones personalizadas sobre destinos turísticos en Cochabamba. La recopilación de información en tiempo real es una clave en el desarrollo del chatbot, ya que optimiza la capacidad de proporcionar respuestas contextualizadas basadas en los datos más recientes disponibles.

Figura 17

Recolección de datos de usuario

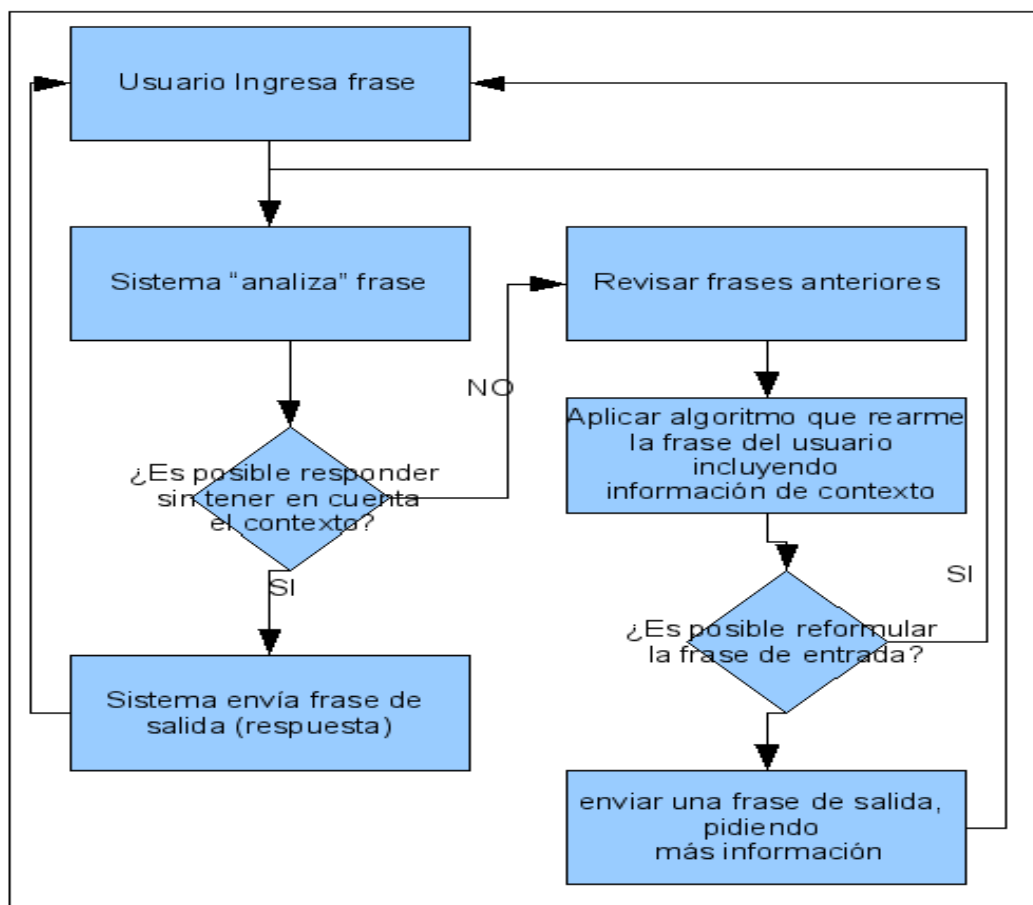


2. Preprocesamiento de datos

Una vez que los datos han sido recolectados, se procede al preprocesamiento. Esta fase es crucial para transformar la información sin procesar en un formato adecuado para el modelo de procesamiento de lenguaje natural (PNL). En Orimon.ai, el preprocesamiento involucra la limpieza de datos, eliminando caracteres especiales, etiquetas HTML y contenido irrelevante de los datos extraídos. Posteriormente, se normaliza el texto, convirtiendo todo a minúsculas y realizando la tokenización (dividiendo las frases en tokens o unidades lingüísticas menores). También se lleva a cabo la eliminación de palabras vacías (stopwords) y la lematización para reducir las palabras a su forma base, permitiendo que el sistema procese la información de manera eficiente y genere respuestas más precisas y coherentes en los diálogos con los usuarios.

Figura 18

Flujo de Chatbot

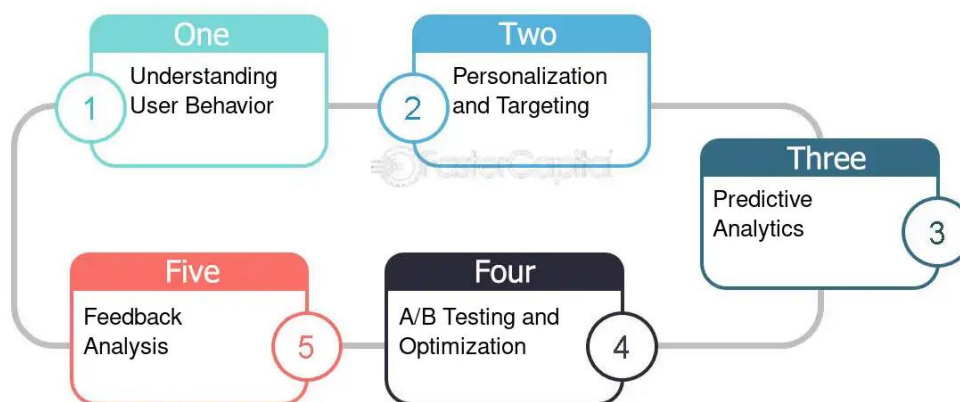


3. Etiquetado y Análisis de Datos

En la fase de etiquetado y análisis de datos, se realiza la clasificación y anotación de las entidades y las intenciones presentes en las consultas de los usuarios. A través de técnicas de anotación semántica, se etiquetan las entidades (como nombres de lugares, horarios, precios, tipos de actividades) y las intenciones (preguntas, solicitudes de recomendaciones, consultas de horarios, etc.). Este proceso permite que el chatbot de Orimon.ai pueda identificar de manera precisa lo que los usuarios desean saber, facilitando respuestas relevantes. Además, se realiza un análisis de sentimientos para identificar el tono de la conversación, permitiendo que el chatbot ajuste su respuesta dependiendo de si el usuario está buscando información técnica, recomendaciones o simplemente haciendo una consulta informal.

Figura 19

Análisis de datos



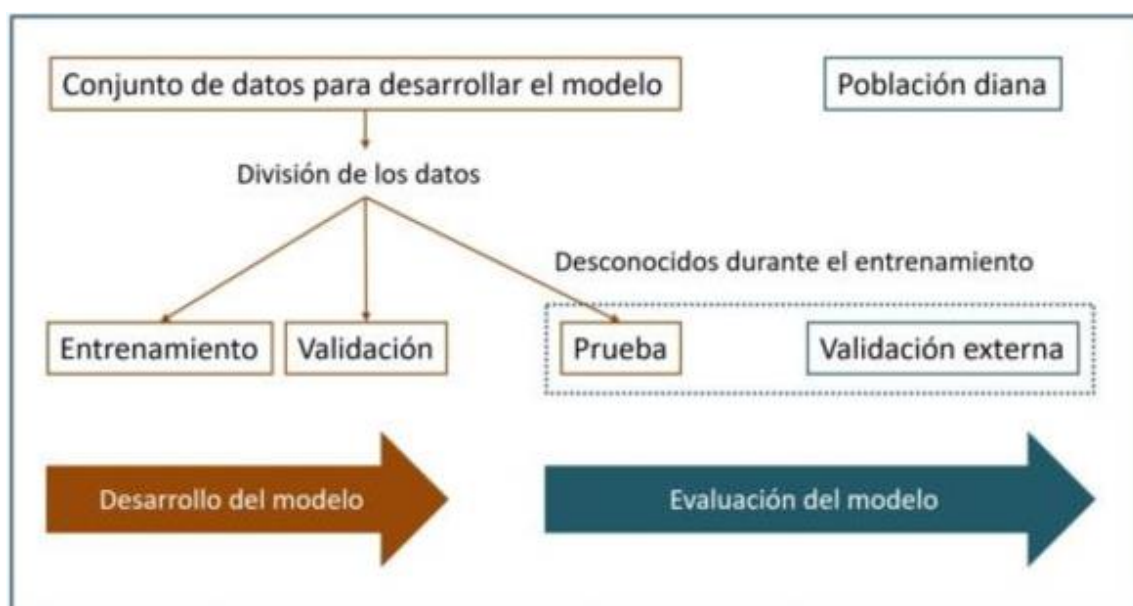
4. División de datos

Una vez etiquetados y anotados los datos, estos se dividen en tres conjuntos principales: entrenamiento, validación y prueba. El conjunto de entrenamiento se utiliza para ajustar el modelo de lenguaje, enseñándole a reconocer patrones en las intenciones y entidades relacionadas con la información turística. Los

conjuntos de validación y prueba permiten realizar ajustes al modelo, como la optimización de hiperparámetros, y verificar que el modelo generalice correctamente a nuevas entradas no vistas durante el entrenamiento. En Orimon.ai, se emplean técnicas de balanceo de datos para asegurar que el modelo esté expuesto a una diversidad de consultas y pueda manejar de manera efectiva una amplia gama de interacciones con los usuarios.

Figura 20

Validación de pruebas

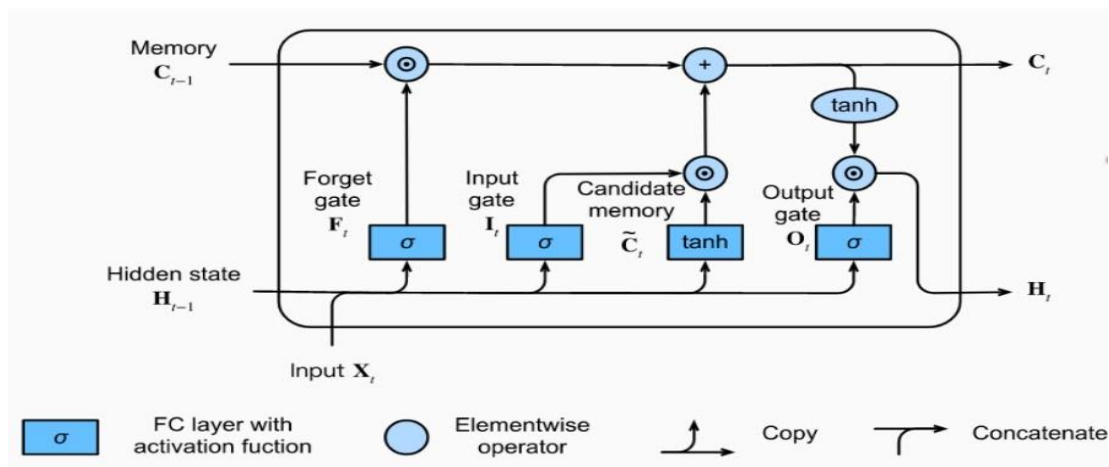


5. Selección del Modelo y Técnica

Orimon.ai selecciona el modelo de procesamiento de lenguaje natural (PNL) más adecuado para el chatbot. Debido a la complejidad de las interacciones y la necesidad de generar respuestas coherentes y naturales, se opta por modelos avanzados como Transformers (p. ej., BERT o GPT) o LSTM (Redes Neuronales Recurrentes de Larga Memoria). Estos modelos permiten al chatbot manejar consultas complejas y realizar inferencias de contexto más precisas, lo que resulta fundamental para ofrecer una experiencia fluida en la consulta de información turística. La capacidad de estos modelos de entender y generar lenguaje humano permite que el chatbot ofrezca respuestas detalladas y personalizadas a las consultas de los usuarios.

Figura 21

Redes neuronales recurrentes



6. Entrenamiento del modelo

Durante la fase de entrenamiento, Orimon.ai optimiza los hiperparámetros del modelo utilizando técnicas de descenso de gradiente como Adam, ajustando parámetros clave para mejorar la precisión y la eficiencia del modelo. Además, se aplican técnicas de regularización, como el abandono, para evitar el sobreajuste y garantizar que el modelo pueda generalizar a nuevos datos. El entrenamiento del modelo se realiza de manera iterativa, ajustando los parámetros hasta que el sistema alcanza un nivel de rendimiento satisfactorio, capaz de responder de manera confiable y precisa a las consultas turísticas.

Figura 22

Descenso de gradiente

Desafíos en la implementación del descenso de gradiente



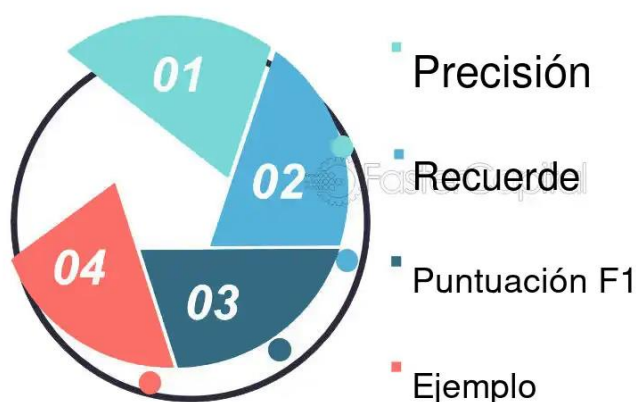
7. Evaluación del modelo

Una vez entrenado el modelo, Orimon.ai realiza una evaluación exhaustiva utilizando el conjunto de prueba. Las métricas claves de evaluación incluyen precisión, recuperación, puntuación F1 y la matriz de confusión. Estas métricas permiten medir la capacidad del modelo para identificar correctamente las intenciones y entidades y para generar respuestas relevantes y adecuadas. Se realiza un análisis detallado de las consultas más frecuentes y de los casos en los que el chatbot puede haber fallado, permitiendo realizar ajustes adicionales para mejorar su rendimiento.

Figura 23

Métricas de evaluación

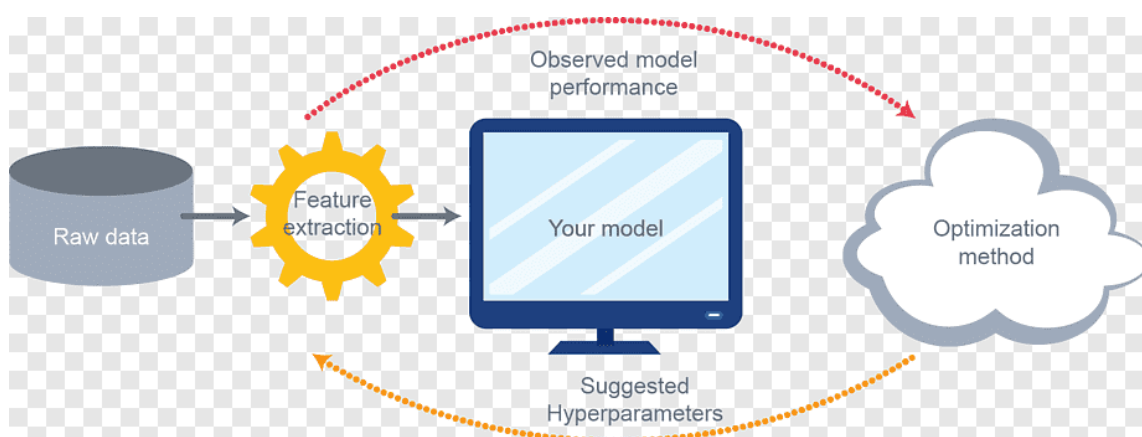
Precisión, recuperación y puntuación F1



8. Ajuste Fino y Optimización

Si el rendimiento del modelo no es óptimo, se lleva a cabo un ajuste fino y optimización adicional. Este proceso puede implicar la optimización de los hiperparámetros, la ampliación del conjunto de datos con más ejemplos de conversaciones o la mejora en la calidad de los datos recopilados a través de web scraping. Además, se actualizan las respuestas del chatbot con la integración de nuevos datos turísticos, garantizando que la información proporcionada esté siempre actualizada y sea precisa.

Figura 24

Optimización de hiperparámetros**9. Implementación y Despliegue**

Después del entrenamiento y la evaluación, Orimon.ai despliega el modelo en un entorno de producción. El chatbot se integra a plataformas web y móviles donde los usuarios pueden interactuar con él. Se habilitan APIs que permiten la interacción directa con el sistema para realizar consultas sobre puntos turísticos en Cochabamba. La integración de web scraping en tiempo real asegura que las respuestas siempre se basen en la información más actualizada disponible, ofreciendo a los usuarios datos relevantes y detallados.

Figura 25

Integración continua

10. Monitoreo y Mejora Continua

El monitoreo continuo es esencial para garantizar que el chatbot siga brindando respuestas precisas y de calidad. Orimon.ai realiza un seguimiento de las interacciones con los usuarios, identificando posibles fallos, áreas de mejora y patrones emergentes. Además, se lleva a cabo una retroalimentación constante para ajustar el modelo y mejorar sus capacidades de respuesta. A medida que se recopilan más datos durante las interacciones, estos se incorporan al proceso de entrenamiento, permitiendo que el modelo continúe mejorando y adaptándose a las necesidades cambiantes de los usuarios.

Figura 26

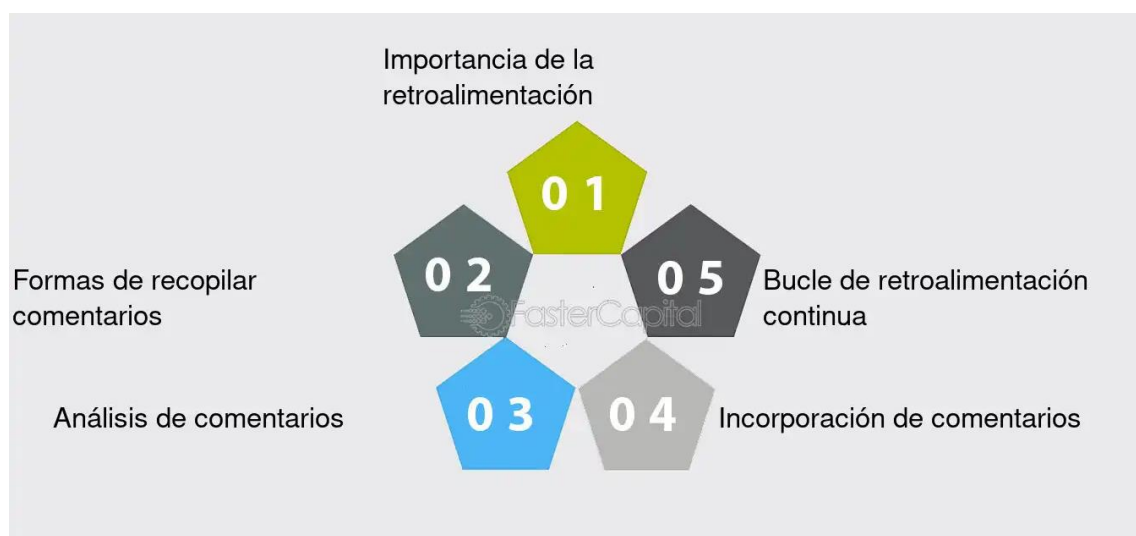
Monitoreo continuo



Ciclo de Retroalimentación y Actualización del Modelo

A través de un ciclo de retroalimentación continua, Orimon.ai optimiza su sistema, brindando una experiencia turística cada vez más precisa y personalizada. La integración de web scraping asegura que el modelo esté siempre actualizado con los últimos cambios en la información turística, y el modelo de PNL se ajusta periódicamente para gestionar una mayor variedad de consultas, mejorando la experiencia del usuario y la eficiencia del sistema.

Figura 27

Retroalimentación continua

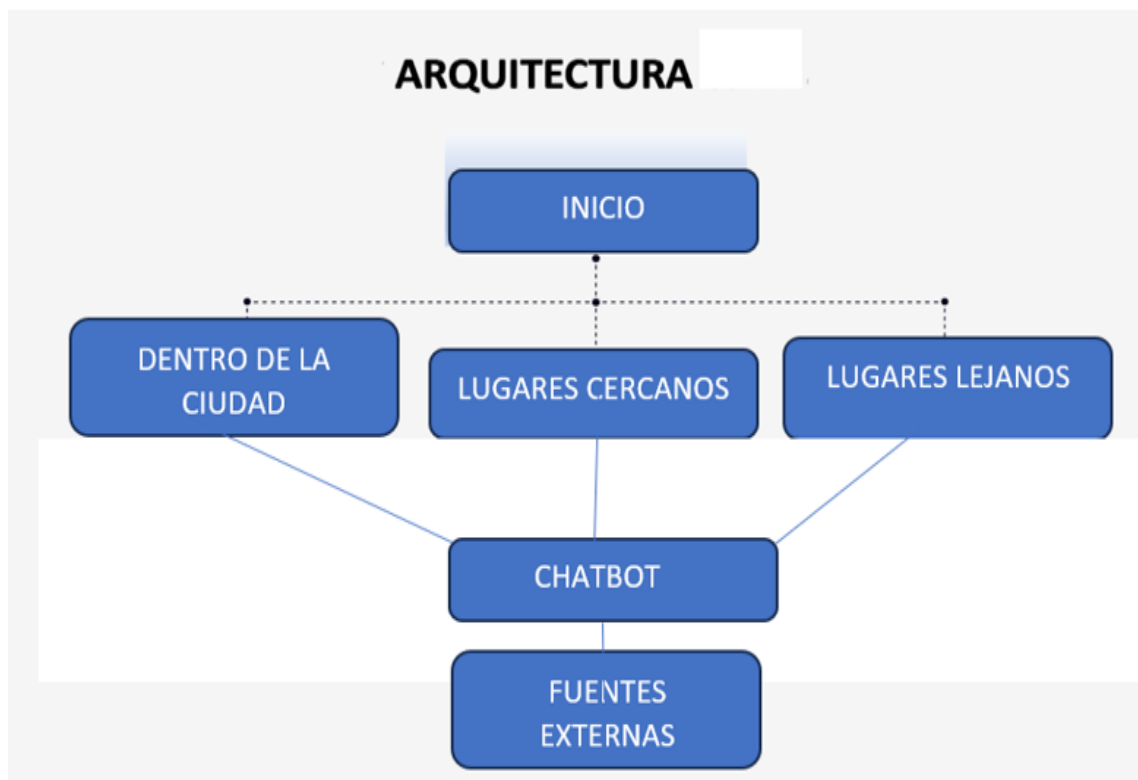
Este proceso asegura que el chatbot de Orimon.ai proporcione información turística precisa y relevante, mejorando la experiencia del usuario y optimizando el acceso a información actualizada para los turistas en Cochabamba.

3.2.2.1.3 Arquitectura Página Web

Se optó por una arquitectura que permita un alto grado de escalabilidad y flexibilidad, lo que facilite la incorporación de nuevas fuentes de información o funcionalidades sin necesidad de modificar su estructura. Esto garantiza que el sistema pueda gestionar la extracción de datos en tiempo real desde múltiples fuentes, asegurando que el asistente virtual ofrecerá información actualizada

No obstante, al implementar esta arquitectura, fue necesario considerar las limitaciones relacionadas con la gestión de grandes volúmenes de datos y la eficiencia en la respuesta a las consultas. La solución asegura que el sistema pueda adaptarse a futuros desarrollos sin afectar su rendimiento ni su capacidad de expansión.

Figura 28

Arquitectura Web

Nota: Arquitectura web propuesta por el postulante

3.2.2.1.4 Requisitos Funcionales**Interfaz de Usuario:**

- La página web cuenta con una interfaz amigable y clara que facilite a los usuarios la exploración de sus distintas secciones, tales como destinos, actividades, clima y como llegar.
- Está diseñada para ser completamente responsiva, adaptándose de forma dinámica a distintos dispositivos y resoluciones de pantalla, garantizando una experiencia coherente tanto en computadoras como en móviles.

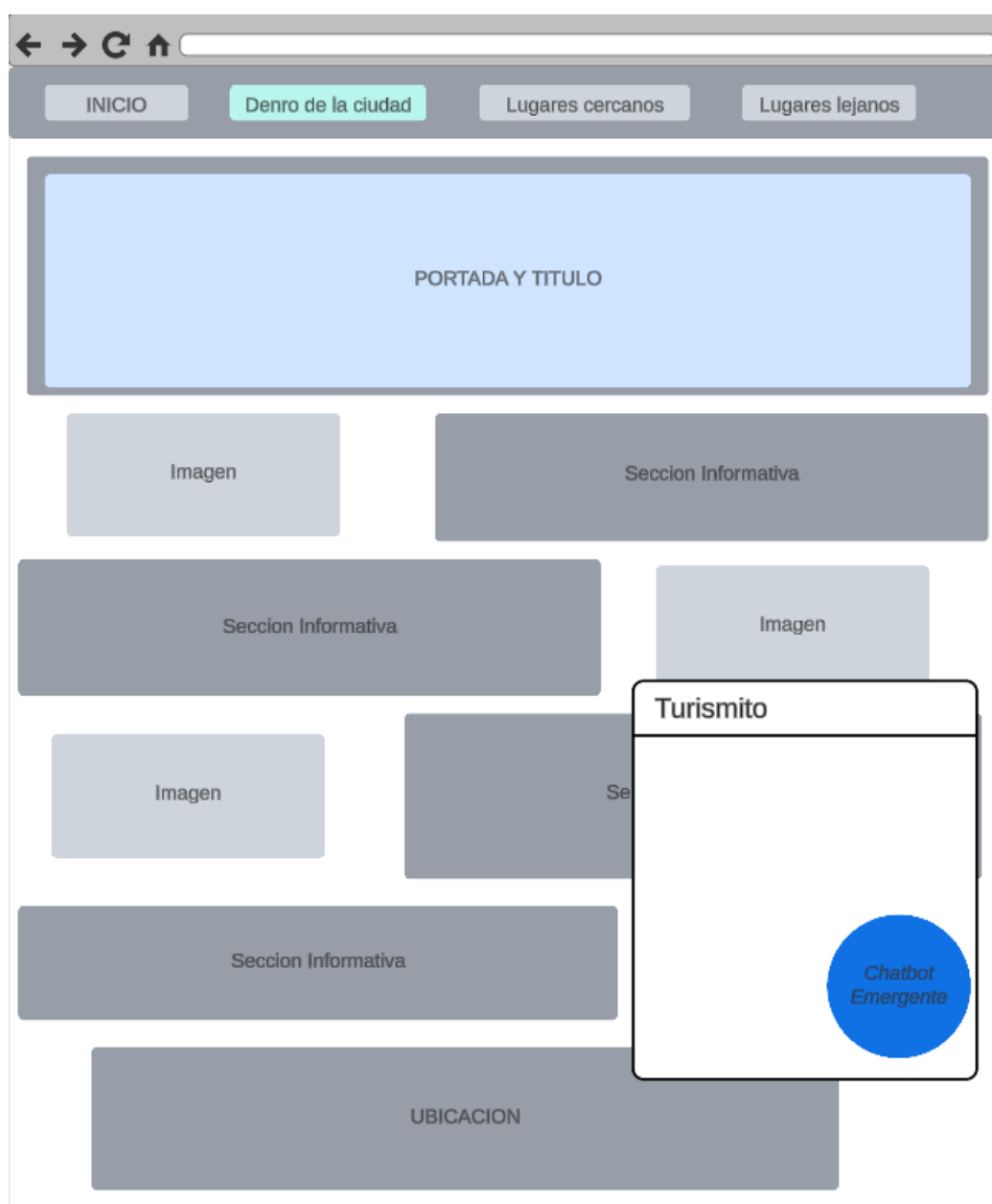
Chatbot:

- El sitio integrará un chatbot, que se mostrará como una ventana emergente en la esquina inferior derecha, listo para responder a preguntas en tiempo real sobre información turística.
- Este asistente virtual no estará conectado a una base de datos, sino que utilizará técnicas de web scraping para obtener información actualizada de fuentes en línea, permitiéndole responder de manera precisa sobre las consultas del usuario.

Web Scraping:

- Se emplearán técnicas de web scraping para extraer datos actualizados y relevantes de diversas fuentes externas, incluyendo información clave publicada en diferentes sitios web. Esta metodología permite que el sistema acceda de manera eficiente a contenidos valiosos, como anuncios de eventos, noticias y otros datos contextuales en línea. La integración de estos datos fortalece la capacidad del chatbot de proporcionar respuestas más precisas y relevantes, enriqueciendo la interacción del usuario con información verificada y de última hora proveniente de Múltiples páginas y plataformas.
- Los datos recolectados deben ser procesados y organizados para ser presentados al usuario en tiempo real, asegurando que la información que se muestra sea precisa y pertinente.

Figura 29

Wireframe de la Pagina web

Nota: Se muestra el modelo de la página web con su menú y sus secciones que estará enlazado con su asistente virtual que estará presente en todas las secciones de la página web, Diseño propuesto por el postulante.

3.2.2.1.5 Requisitos No Funcionales

Rendimiento: La página ofrece tiempos de carga menores a tres segundos, brindando así una experiencia rápida y eficiente, incluso bajo condiciones de alto tráfico.

- Maneja múltiples solicitudes al chatbot de manera simultánea sin afectar el desempeño general del sistema.

Usabilidad:

- El diseño es intuitivo y accesible para personas con distintos niveles de habilidad tecnológica, con una navegación sencilla y un flujo lógico de información.
- Se realizarán evaluaciones periódicas de la usabilidad para detectar posibles problemas y optimizar la experiencia de uso continuamente.

Disponibilidad:

- Tanto la página web como el chatbot deben estar operativos las 24 horas del día, los 7 días de la semana, minimizando cualquier tiempo fuera de servicio.
- Es crucial implementar mecanismos para una recuperación ágil y eficaz en caso de que se produzcan fallas o interrupciones.

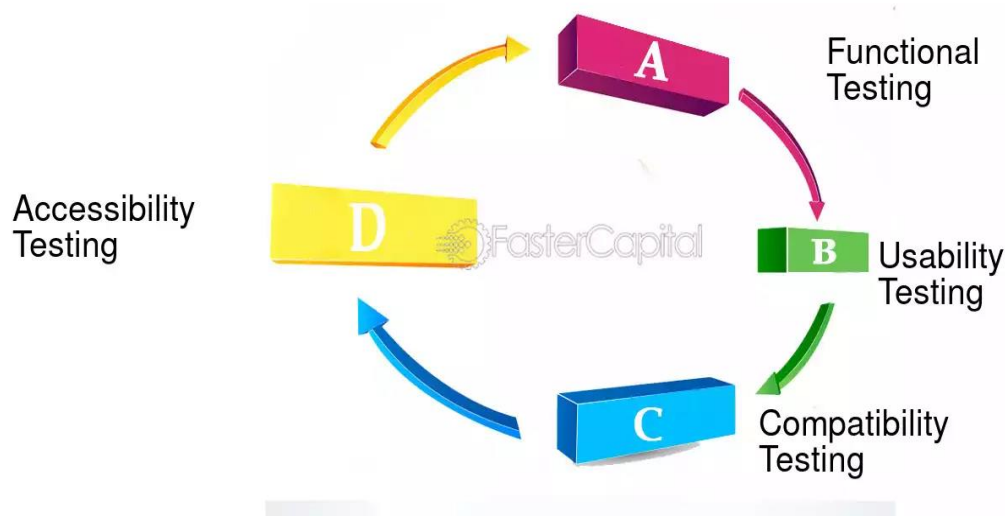
Compatibilidad:

- El sitio es compatible con los navegadores más utilizados (como Chrome, Firefox, Safari y Edge), así como con los principales sistemas operativos de escritorio y móviles.
- El chatbot y otras funciones principales funcionarán correctamente en diversos dispositivos, incluyendo smartphones, tablets y computadoras, sin perder ninguna de sus capacidades.

Figura 30

Requisitos No funcionales

Functional, Usability, Compatibility, and Accessibility



Características de la API:

- **La interpretación del lenguaje natural** permite a la API analizar el texto ingresado por el usuario, identificando intenciones y elementos clave para comprender el contexto de la conversación. Este proceso permite al chatbot interpretar y responder de manera más precisa, adaptándose al flujo dinámico del diálogo y mejorando la interacción con el usuario.
- **Producción de Texto:** Mediante el uso de modelos sofisticados, la API es capaz de generar respuestas elaboradas y naturales, lo que mejora la experiencia del usuario al interactuar con el sistema.
- **Personalización y Flexibilidad:** La plataforma puede ser adaptada utilizando datos específicos del ámbito turístico, permitiendo que el chatbot ofrezca información precisa y actualizada sobre destinos y actividades disponibles.

- **Integración de Funcionalidades Adicionales:** Además del procesamiento del texto, la API puede combinarse con otras tecnologías y servicios. El web scraping se utiliza para recopilar datos en tiempo real, enriqueciendo la experiencia del usuario y ofreciendo respuestas más completas.

Figura 31

Procesamiento NLP

Proceso de Trabajo de la API:

- 1. **Recepción de Consultas:** El usuario envía una pregunta a través del chatbot.

Figura 32
Recepción de consultas de usuarios

Historial de chats Oportunidades de entrenamiento de bots ⓘ

Fecha: 14 de noviembre de 2024 - 14 de noviembre de 2024 Filtros Refrescar

Nombre	Resumen	Idioma	Último mensaje	
Compañera mariquita ro...	Descubre los mejores lugares turísticos y gastronómicos en ...	Español	Hace 2 horas	
ardilla índigo considerab...	Descubre cómo llegar a la hermosa Laguna Corani y dónde s...	Español	Hace 5 horas	
Suela gris que se pone n...	Descubre los mejores lugares de Cochabamba y cómo llegar ...	Español	Hace 5 horas	
Guacamayo de chocolat...	Descubre cómo llegar a la hermosa Laguna Angostura.	Español	Hace 6 horas	
lobo rosa calificado	Descubre los mejores lugares turísticos en Cochabamba.	Español	Hace 6 horas	
Alce azul acuático	Saludos, ¿en qué puedo ayudarte?	Español	Hace 6 horas	
Cefalópodo contemporá...	Descubre cómo llegar a Cochabamba en solo unos pasos.	Español	Hace 6 horas	
Resumen de la mosca d...	Descubre los mejores guarapos en Cliza con nuestra guía de ...	Español	Hace 6 horas	
Guacamayo de chocolat...	Descubre el clima actual en Cochabamba y planifica tu día co...	Español	Hace 6 horas	

Nota: Se recibe las consultas de los usuarios hechas en distintas cuentas y a diversas consultas sobre el departamento de Cochabamba, pueden ser consultas de manera simultanea, Propuesta por el postulante.

- 2. **Análisis de la Consulta:** La plataforma envía la consulta para su evaluación.

Una vez hechas las consultas de distintos usuarios se logra obtener datos importantes de las distintas conversaciones que se tuvieron a lo largo de los mensajes realizados.

Figura 33

Resumen de la consulta de un usuario

Resumen del chat	
Resumen	Descubra los mejores lugares turísticos y gastronómicos en Cochabamba, como el Cristo de la Concordia, Villa Tunari, la Laguna Corani y el Parque de la Familia. ¡No te lo pierdas!
Idioma	Español
Detalles del usuario	
Nombre	Compañera mariquita rosa
Correo electrónico	Desconocido
Identidad del usuario	desconocido

Nota: Las conversaciones que tienes el usuario con el chatbot son registradas ya que te brinda un resumen de las consultas realizadas y de los temas que se trataron en la conversación para un análisis más rápido de la conversación, así mismo se logra ver el idioma en el que fue escrito y además que se puede ver un nombre al azar otorgado por la herramienta para separar distintas conversaciones, Propuesta hecha por el postulante.

A continuación, también se logra ver el resto de información obtenida de la conversación con un usuario determinado mostrando desde que plataforma se conecto la fecha y la hora, desde donde se inicio la conversación y un ID de identificación de plataformas, como también ahí mismo se logra percibir el tipo de dispositivo y la versión de su sistema operativo del dispositivo en el cual inicio la conversación.

Figura 34

Datos extras de la consulta de un usuario.

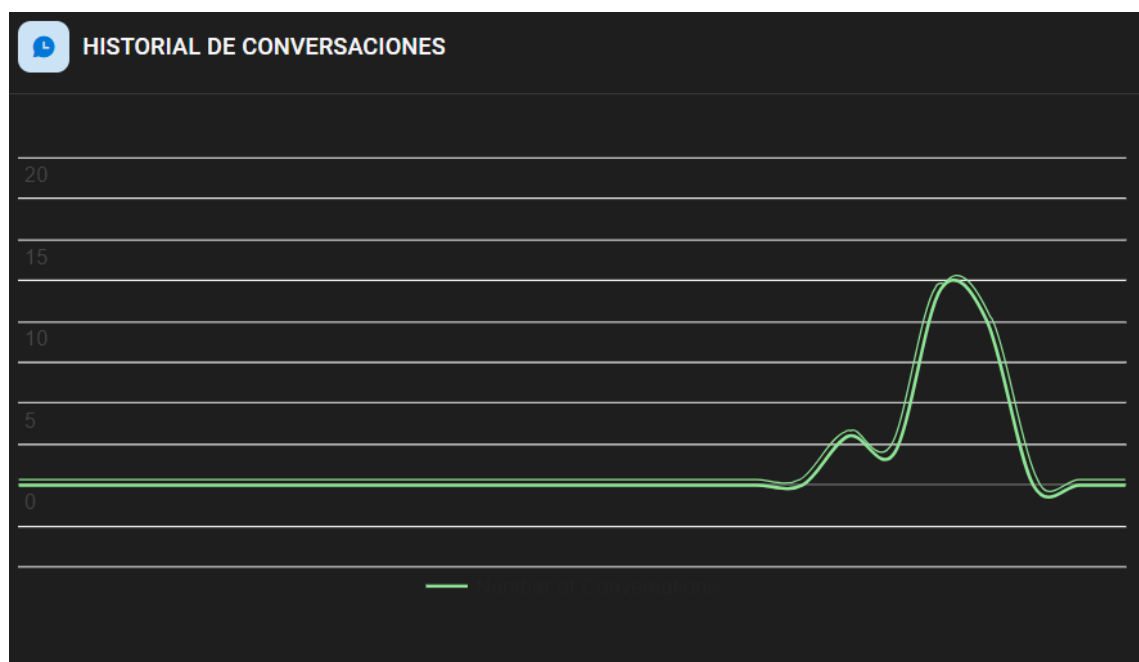
Perfil de uso	
Plataforma	web
Inicio de sesión	15 de noviembre de 2024 a las 00:50:57.182Z
Sesión iniciada desde	https://www-cochabambaturism-o-blog-bo.filesusr.com/html/acc892_8fdadb633e4751c03c7f855c0d7f0086.html
ID de sesión	67369af17ee0365e901f85bd
Identificación de la plataforma	7bf8d9ebf39c88_f2832d61-b39f-4262-ac06-97edccd177da

Propiedades del dispositivo	
Dispositivo	De oficina
Sistema operativo	Ventanas
Versión del sistema operativo	10

Nota: Se muestra el perfil de uso de un usuario que entablo una conversación con el chatbot donde, cuando y desde donde la inicio.

De igual manera podemos ver en el grafico que se muestra a continuación la frecuencia de conversaciones iniciadas en distintos momentos del día con los distintos usuarios que ingresaron a la pagina y dieron uso del chatbot emergente.

Figura 35

Grafica de Frecuencia

Nota: Se muestra en la parte izquierda el número de conversaciones y la línea verde demuestra el inicio de distintos chats y se puede observar en la línea de color verde como al iniciar una conversación de distintos lugares va subiendo y teniendo una elevación de hasta de 13 chats simultáneamente y a medida que dejan de usarlo se va volviendo a 0 chats conectados. Propuesta hecha por el postulante.

3. **Generación de Respuesta:** La API devuelve una respuesta adaptada al contexto de la conversación.

A continuación, se muestra un ejemplo de como se hace la solicitud de la API para obtener una respuesta.

Figura 36

Ejemplo de solicitud

```

// in JS
const axios = require('axios');
let data = JSON.stringify({
  "type": "message",
  "info": {
    "psid": <some_random_value + '_' + tenantId>,
    "sender": "user",
    "tenantId": <id_of_your_bot>,
    "platformName": "web"
  },
  "message": {
    "id": <some_unique_value>,
    "type": "text",
    "payload": {
      "text": "What is Orimon?"
    }
  }
});

let config = {
  method: 'post',
  url: 'api_endpoint_mentioned_on_top_of_this_page',
  headers: {
    'authorization': 'apiKey <you_apiKey_here>',
    'Content-Type': 'application/json'
  },
  data : data
};

axios.request(config)
.then((response) => {
  console.log(JSON.stringify(response.data));
})
.catch((error) => {
  console.log(error);
});

```

Nota: id: un identificador único para cada entrada que proviene del usuario.

type: esto indica el tipo de entrada que tu bot recibe del usuario. Actualmente está fijado en "text".

payload: La entrada real que recibirá el bot. La textclave dentro de este objeto incluirá el mensaje del usuario. Será un valor de cadena.

A continuación, se muestra un ejemplo de una respuesta con una estructura:

Figura 37

Respuesta en formato Json

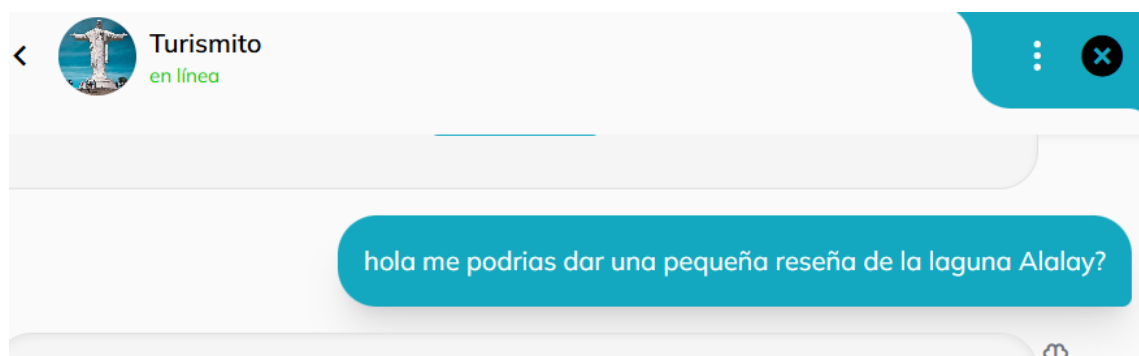
```
{
  "status": "success",
  "data": {
    "type": "message",
    "timestamp": "2023-06-29T13:04:57.041Z",
    "psid": <some_random_value + '_' + tenantId>,
    "sessionId": <generated_session_id_of_current_conversation>,
    "messages": [
      {
        "chatLogId": <generated_chatlog_id_of_current_conversation>,
        "sessionId": <generated_session_id_of_current_conversation>,
        "psid": <some_random_value + '_' + tenantId>,
        "tenantId": <your_tenant_ID>,
        "id": <some_unique_value>,
        "type": "text",
        "payload": {
          "text": "Orimon.ai is a powerful tool for businesses looking to impro
          "buttons": []
        },
        "timestamp": "2023-06-29T13:04:57.041Z",
        "customMessage": true
      }
    ],
    "statuses": []
  },
  "message": "success"
}
```

Nota: chatLogId :Esta es una identificación generada por el sistema para la consulta específica del usuario y la respuesta de su bot y se puede usar para identificarlos.

sessionId :Esta es una identificación generada por el sistema para toda la sesión que tiene lugar entre su usuario y su bot y se puede utilizar para identificarlos.

payload: Esta es la respuesta real y contiene 2 valores "text"y una matriz "buttons". "text"Contiene un valor de cadena que es la respuesta de texto sin

Figura 39

Pregunta de Usuario

Nota: Se realiza la pregunta del usuario pidiendo una reseña de un sitio del cual desea saber, diseño propuesto por el postulante.

A continuación, se hace un web scraping de una pagina seleccionada que contenga la información del sitio buscado tomando en cuenta su URL se indaga en la información de la pagina y se obtiene la respuesta a la pregunta realizada.

Figura 40

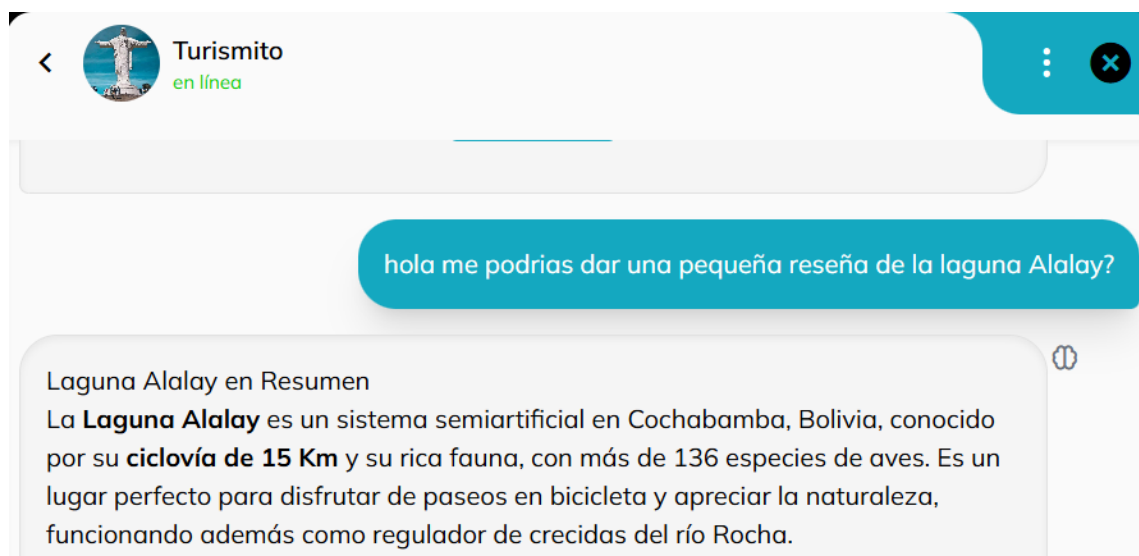
Web Scraping Ejemplo

Nota: Se muestra como la web scraping hace la labor de obtener y mostrar la información en un mensaje como se logrará ver a continuación.

5. **Presentación de la Respuesta:** El chatbot muestra la respuesta al usuario, permitiendo una interacción fluida y natural.

Figura 41

Respuesta del chatbot



Nota: Se puede observar como la respuesta dada esta siendo obtenida de una pagina externa mostrada anteriormente respondiendo de manera corta y clara sin ser extenso en su respuesta facilitando la comprensión del usuario y aclarando o recibiendo información en tiempo real de su sitio buscado.

CAPÍTULO IV

FASE DE

DESARROLLO

4.1. Desarrollo

4.1.1. Diseños del proyecto

4.2.1.1. Diseño Página Web

La arquitectura de la página web fue diseñada con una estructura modular que permitiera la integración fluida del web scraping y el chatbot.

Primeramente, se cuenta con la portada de bienvenida de la página que recibe al usuario una vez que ingresa a la página con una reseña del departamento de Cochabamba. Al mismo tiempo en la parte inferior derecha se logra percibir el botón desplegable del chatbot que está presente en todas las secciones de la página.

Figura 42

Portada Página Web



Nota: Se muestra en pantalla la portada y sección de menús desplegables de los distintos sitios turísticos tomados en cuenta este modelo es propuesta por el postulante.

En la parte superior se tiene el menú de navegación que esta ordenado por los lugares dependiendo de la distancia que se encuentra del centro de la ciudad así como también se ve un menú desplegable y mostrando los sitios que se encuentran en esa sección y se logra percibir las otras secciones que cuenta el

menú de navegación con un diseño manejable y estructurado de manera que el usuario pueda manejarlo de manera adecuada.

Figura 43

Menú desplegable Página Web



Nota: Se tiene la presentación de los lugares que se presenta en un menú desplazable de una sección del menú mostrando los sitios tomados en cuenta propuesto por el postulante.

4.2.1.2. Título y portada de un sitio de la página web

Se muestra primeramente el nombre del sitio con una pequeña reseña del lugar y con una imagen llamativa del sitio a mostrarse con un orden y estructura que se repite en los distintos sitios que contiene la página web.

Figura 44

Título y portada de un sitio web



Nota: Se muestra el diseño de portada de un sitio Turístico escogido por un usuario que muestra el título y una reseña del lugar con una imagen llamativa, Propuesto por el postulante.

4.2.1.3. Contenido de un sitio de la página web

Se muestra a continuación el contenido estructurado que se tiene en todos los sitios de la página logrando percibir que se ordenada como ser con la geografía del sitio, el clima, la biodiversidad y Sobre todo como llegar al sitio dependiendo a la necesidad del usuario con esto se puede mostrar una información corta y clara y sobre todo relevante de un sitio a la hora de tomar decisiones y evitar percances.

Con esta estructura se puede aligerar la búsqueda de datos o información de un sitio determinado facilitando la obtención de datos de un sitio al que el usuario desee visitar y tomar en cuenta las distancias y sobre todo el tiempo.

Figura 45

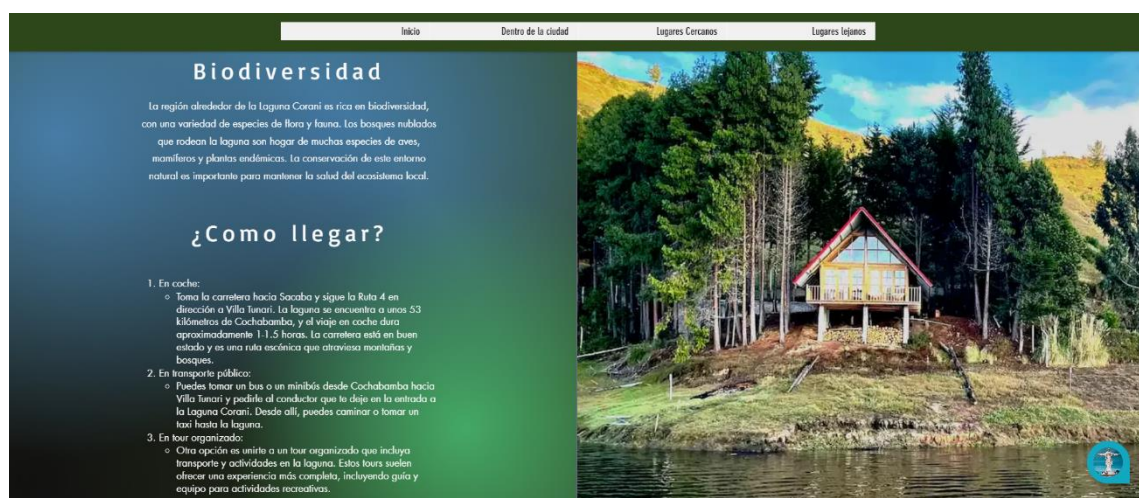
Contenido de sitio sección geografía y clima



Nota: Se muestra la primera parte del contenido de un sitio de la página web mostrando el orden estructura y datos necesarios a la hora de decidir sobre un sitio al que se quiere visitar, propuesto por el postulante.

Figura 46

Contenido del sitio sección Biodiversidad y ¿Cómo llegar?



Nota: Se muestra la otra parte de la sección de contenido con la información de biodiversidad y ¿Cómo llegar?, diseño propuesto por el postulante.

4.2.1.4. Pie de Página

Al final de cada sección se tiene el apartado de la ubicación exacta del sitio que es totalmente usable y navegable para facilitar aun mas la visita al sitio.

Figura 47

Contenido en la sección pie de página.



Nota: Se muestra la ubicación de un sitio para que el usuario pueda acceder desde su dispositivo y le marque la ruta más rápida o el camino más accesible para llegar al destino, Diseño propuesto por el postulante.

4.2. Diseño ChatBot

4.2.1. Diseño de ventana emergente

El diseño del chatbot fue planeado para tener un acceso fácil y sobre todo en cualquier sección o en cualquier momento que se haga la navegación por la página web tomando en cuenta que se usó la tecnología de Orimon.ai para diseñarlo es el siguiente:

Figura 48

Diseño Chatbot Emergente



Nota: Con este diseño se logra hacer una ventana emergente en la página con la tecnología Orimon.ai, que también se cuenta con las opciones de historial de chats y crear una nueva conversación, Propuesto por el postulante.

4.2.3 Módulo 3

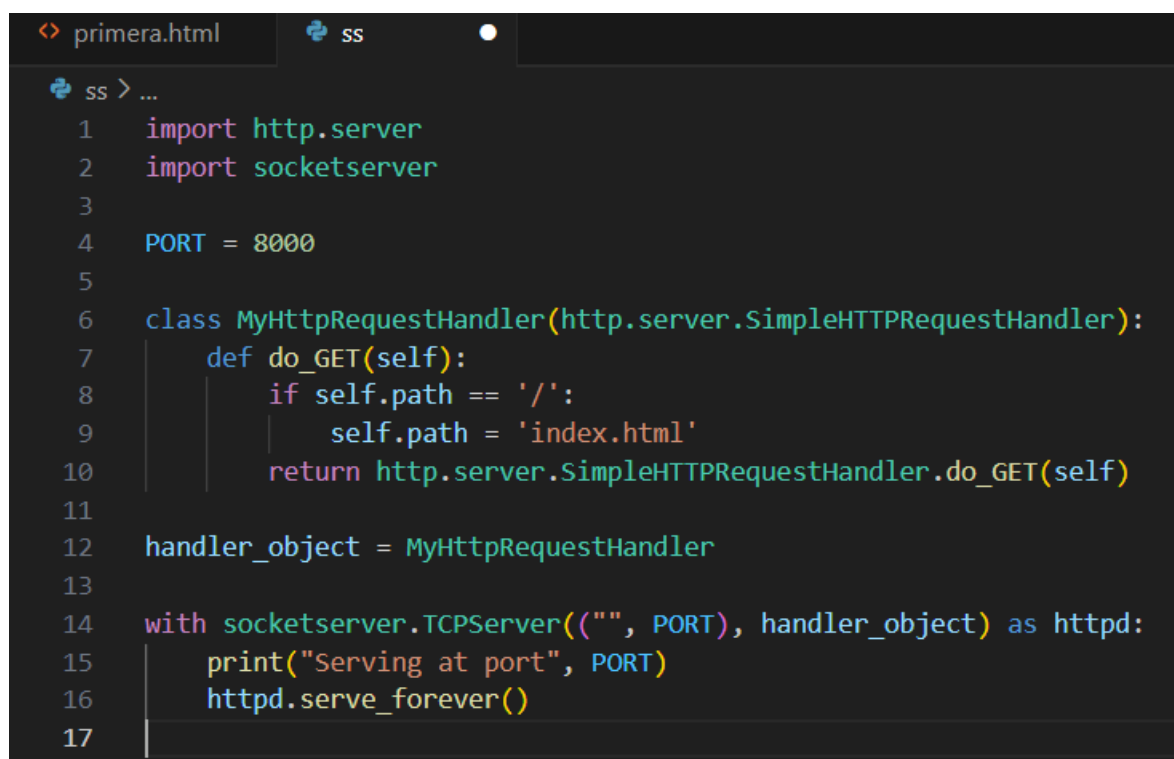
4.2.3.1 Desarrollo

4.2.3.1.1. Código de conexión

Se muestra a continuación la conexión del Chatbot que entrega el servidor con la tecnología Orimon.ai a la pagina Web mediante un código Index.html que logra que se pueda usar desde la página web el bot creado.

Figura 49

Código de Conexión del bot



```

primer.html  ss
ss > ...
1  import http.server
2  import socketserver
3
4  PORT = 8000
5
6  class MyHttpRequestHandler(http.server.SimpleHTTPRequestHandler):
7      def do_GET(self):
8          if self.path == '/':
9              self.path = 'index.html'
10             return http.server.SimpleHTTPRequestHandler.do_GET(self)
11
12  handler_object = MyHttpRequestHandler
13
14  with socketserver.TCPServer(("", PORT), handler_object) as httpd:
15      print("Serving at port", PORT)
16      httpd.serve_forever()
17

```

Nota: El servidor entrega el archivo index.html al navegador del usuario que contiene el código para incrustar el bot de Orimon.ai y hacer visible el chatbot en la pagina web, propuesto por el postulante.

4.2.3.1.2. Estilo de la ventana Emergente

Se muestra el estilo de la ventana emergente del chatbot para que esté presente en todo momento y en la esquina inferior derecha de pantalla de la página web y al mismo tiempo que estará presente en todas las ventanas de la pagina web.

Figura 50

Estilo ventana emergente

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Embed Orimon AI Bot</title>
  <style>
    /* Set the body and html to 100% to ensure the container fills the entire page */
    html, body {
      height: 100%;
      margin: 0;
      padding: 0;
    }

    /* The container for the bot */
    #orimon-bot-container {
      height: 100%;
      width: 100%;
    }
  </style>
</head>

```

Nota: En el código están las dimensiones de ancho y alto que se tendrá en la ventana emergente del chatbot que al presionar se abrirá en una ventana emergente, propuesto por el postulante.

4.2.3.1.3. Enlace de vinculación

A continuación, se muestra el enlace de la herramienta Orimon.Ai con el cual se logra vincular al chatbot a la página web mediante un código que te brinda la herramienta para poder vincular a cualquier pagina web que se desee y luego modificarlo según el requerimiento que se tenga.

Figura 51

Código de vinculación Orimon.Ai y la Página web

```
<body>
  <!-- Container for the Orimon AI bot -->
  <div id="orimon-bot-container"></div>

  <!-- Orimon AI Bot script -->
  <script>
    (function() {
      const container = document.getElementById('orimon-bot-container');
      const script = document.createElement('script');
      script.src = "https://bot.orimon.ai/deploy/index.js";
      script.setAttribute('tenantId', 'f2832d61-b39f-4262-ac06-97edccd177da');
      container.appendChild(script);
    })();
  </script>
</body>
</html>
```

Nota: la imagen muestra la vinculación que se tiene entre Orimon.ai y la página web para su uso en cualquier momento, propuesto por el postulante.

4.2.3.1.4. Código de búsquedas de las páginas web externas

La biblioteca BeautifulSoup es una herramienta clave en proyectos de web scraping porque permite analizar y extraer datos de páginas web de forma eficiente. Básicamente, ayuda a revisar y procesar el contenido de las URL de sitios externos desglosando su estructura HTML, lo cual facilita capturar los elementos importantes que se buscan. Esto es útil cuando se implementa en soluciones como chatbots, ya que el proceso de scraping se automatiza para obtener información actualizada, mejorando así la precisión y el detalle de las respuestas a las consultas de los usuarios. Cuando un usuario hace una pregunta, el sistema usa BeautifulSoup para analizar y extraer los datos más relevantes de los sitios web, lo que asegura que las respuestas del chatbot sean precisas, útiles y basadas en información reciente y relevante.

Figura 52

Código de Biblioteca BeautifulSoup

```

1  import requests
2  from bs4 import BeautifulSoup
3
4  def scrape_tripadvisor():
5      url = "https://www.tripadvisor.es/Attractions-g297315-Activities-Cochabamba_Cochabamba_Department.html"
6      response = requests.get(url)
7      soup = BeautifulSoup(response.content, 'html.parser')
8
9      title = soup.find('title').text.strip()
10     attractions = []
11     listings = soup.find_all('div', class_='listing_title')
12     for listing in listings:
13         site = listing.text.strip()
14         link = "https://www.tripadvisor.es" + listing.find('a')['href']
15         attractions.append({'site': site, 'link': link})
16
17     return title, attractions
18
19 def scrape_boliviaentusmanos():
20     url = "https://www.boliviaentusmanos.com/turismo/atractivos-departamento/cochabamba.html"
21     response = requests.get(url)
22     soup = BeautifulSoup(response.content, 'html.parser')
23
24     title = soup.find('title').text.strip()
25     attractions = []
26     cards = soup.find_all('div', class_='card-body p-4')
27     for card in cards:
28         site = card.find('h4').text.strip() if card.find('h4') else 'No title'
29         description = card.find('p').text.strip() if card.find('p') else 'No description'
30         attractions.append({'site': site, 'description': description})
31
32     return title, attractions
33
34 def generate_html(tripadvisor_data, boliviaentusmanos_data):
35     html_content = f"""
36

```

Nota: El proceso de web scraping comienza con la importación de bibliotecas necesarias, como requests para manejar solicitudes HTTP y BeautifulSoup de bs4 para analizar el contenido HTML. Se define la URL del sitio web a analizar, en este caso, Orimon.ai, y se realiza una solicitud HTTP utilizando requests.get() para obtener el contenido de la página.

Una vez obtenida la respuesta, se verifica su éxito mediante el código de estado HTTP, donde un código 200 indica una solicitud exitosa. Luego, se analiza el contenido HTML con BeautifulSoup, lo que facilita la manipulación del HTML. Finalmente, se extrae información relevante, como títulos, párrafos y enlaces, utilizando el objeto soup, permitiendo la recolección eficiente de datos para análisis posteriores. Propuesto por el postulante.

4.2.4. Módulo 4

4.2.4.1 Pruebas

4.2.4.2 Entrenamiento de Chatbot

4.2.4.2.1. Mediante web scraping

Obtener datos actualizados de sitios web turísticos para proporcionar respuestas en tiempo real sobre sitios, lugares y horarios.

El Chatbot recibe consultas de distintos temas o sitios que se plantee el usuario y mediante Web scraping se puede obtener información del sitio buscado.

Figura 53

Sitios de búsqueda Web Scraping

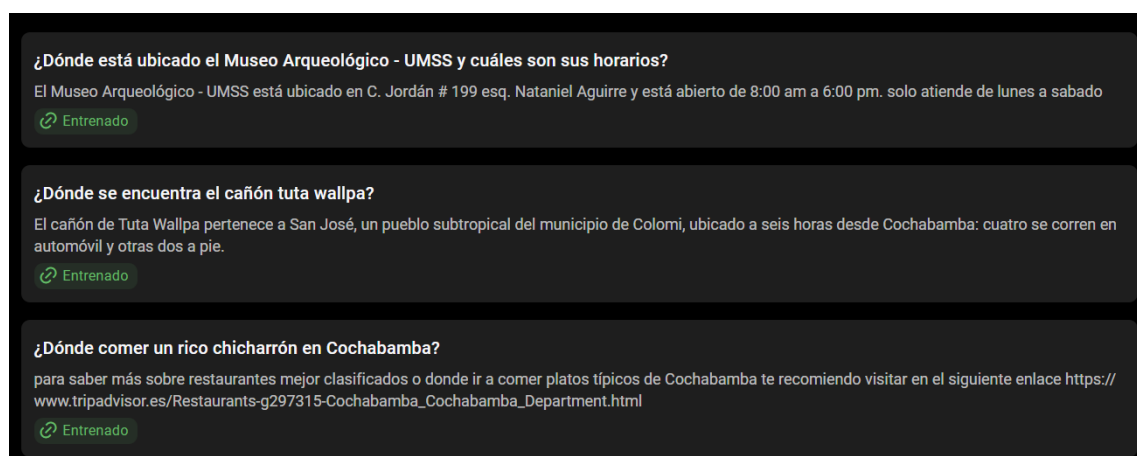
https://www.accuweather.com/es/bo/cochabamba/32716/hourly-weather-forecast/32716#google_vignette	Hace 8 horas	🔄	🟢	🗑️	👁️
https://www.boliviaentusmanos.com/hoteles-bolivia/ciudad/cochabamba/1/hoteles.html#google_vignette	Hace 10 horas	🔄	🟢	🗑️	👁️
https://www.boliviaentusmanos.com/turismo/atractivos-departamento/cochabamba.html	Hace 6 meses	🔄	🟢	🗑️	👁️
https://www.boliviaentusmanos.com/turismo/atractivos/colina-de-san-sebastian.html	Hace 10 horas	🔄	🟢	🗑️	👁️
https://www.boliviaentusmanos.com/turismo/atractivos/el-pueblito.html	Hace 10 horas	🔄	🟢	🗑️	👁️
https://www.boliviaentusmanos.com/turismo/atractivos/laguna-alalay.html	Hace 10 horas	🔄	🟢	🗑️	👁️
https://www.boliviaentusmanos.com/turismo/atractivos/museo-de-historia-natural-alcide-dorbigny.html	Hace 10 horas	🔄	🟢	🗑️	👁️
https://www.boliviaentusmanos.com/turismo/atractivos/palacio-portales-centro-simon-i-patino.html	Hace 10 horas	🔄	🟢	🗑️	👁️
https://www.boliviaentusmanos.com/turismo/cochabamba/parques-y-		🔄	🟢	🗑️	👁️

Nota: Se muestra las distintas URL's brindadas a la herramienta para su búsqueda web para extracción de información logrando ver que se tiene una variedad de páginas confiables y seguras. Propuesta hecha por el postulante.

4.2.4.2.2. Mediante mensajes programados

Crear respuestas automáticas predefinidas para preguntas frecuentes o situaciones comunes, como saludos despedidos o de algún sitio en particular del que se desea brindar la información más detallada o más precisa que no se logre encontrar en las distintas paginas externas.

Figura 54

Entrenamiento de mensajes programados

Nota: se puede ver mensajes programados ah preguntas fuera de lo común pero que se hicieron en 7algún momento que servirá para tener una conversación fluida y con respuesta a algunas mensajes especiales, propuesto por el postulante.

4.2.4.2.3. Utilizando archivos PDF

Entrenar al chatbot para extraer y responder obteniendo información contenida en documentos PDF relacionados con guías turísticas y folletos informativos.

Figura 55

PDF de folletos turísticos

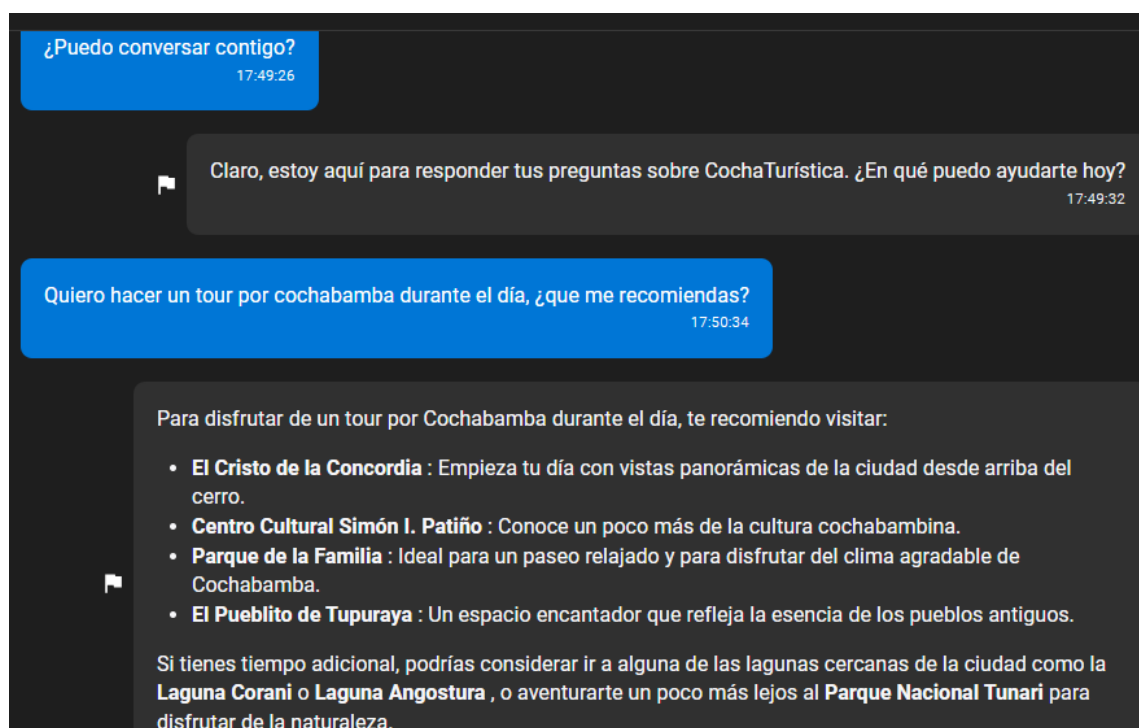
Nota: se muestra los folletos virtuales que se tienen del departamento que serán tomados en cuenta para el entrenamiento del chatbot con la información obtenida de ahí.

4.1.4.3. Fluidez en conversación

En la imagen podemos ver que las conversaciones son lo mas natural posible y la respuesta que brinda es corta clara y concisa referente a la pregunta hecha por el usuario.

Figura 56

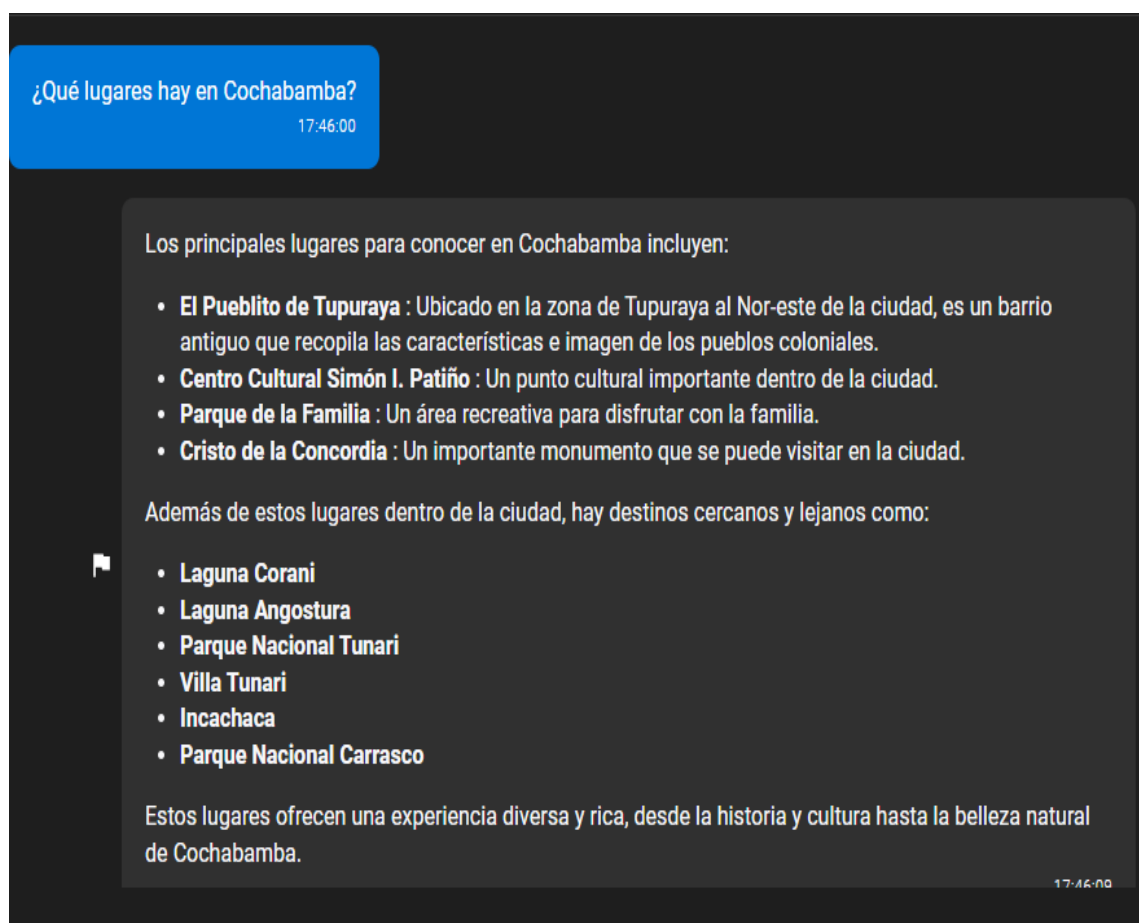
Conversación casual 1



Nota: Se puede percibir una fluidez de conversación respondiendo a preguntas casuales y siguiendo un hilo de conversación con el usuario.

De la misma manera se consultar algunas preguntas más específicas de algo en particular que se desea hacer en el departamento y se espera una respuesta estructurada y concisa.

Figura 57

Conversación casual 2

Nota: En la imagen se puede ver la conversación con el bot y el usuario obtiene la respuesta a su consulta. Propuesto por el postulante.

Si se tiene preguntas de como llegar a un destino o las formas de poder hacerlo el chatbot logra dar la información requerida a dicha consulta con su estructura clara y ordenada según la forma de llegar y a q distancia se encuentra.

Figura 58

Conversación casual 3

como llego a la laguna angostura?
17:43:02

Para llegar a la Laguna Angostura desde Cochabamba, tienes dos opciones:

En coche:

- Toma la carretera hacia Punata. La ruta principal es la carretera antigua a Santa Cruz (Ruta 7).
- La laguna está claramente señalizada y es fácil de encontrar.
- El viaje en coche tarda aproximadamente 30-40 minutos, dependiendo del tráfico.

En transporte público:

- Puedes tomar un minibús o un trufi (taxi colectivo) desde la ciudad de Cochabamba en dirección a Punata o Cliza.
- Debes bajarte en el desvío hacia la laguna y desde allí puedes caminar o tomar un taxi hasta el embalse.

17:43:11

Nota: Se muestra la conversación y respondiendo a la consulta del usuario de manera óptima y clara. Propuesta por el postulante.

Si se presenta la consulta sobre el clima en Cochabamba el Chatbot mostrara de manera ordenada los distintos datos del clima que se tiene actualmente en el departamento.

Figura 59

Conversación casual 4

dime el clima actual en cochabamba
17:03:33

El clima de la ciudad de Cochabamba actualmente tiene las siguientes condiciones:

- Temperatura: **81°F**
- Calidad del aire: **Buena**
- Ráfagas de viento: **Hasta 14 mi/h**
- Humedad: **Entre 54% y 61%**
- Punto de rocío: **Alrededor de 48° F**
- Nubosidad: **Entre 93% y 100%**
- Visibilidad: **Entre 6 y 7 Millas**
- Techo de nubes: **30000 pies**

Puede consultar las predicciones actualizadas [aquí](#) (por favor ábralo en una nueva pestaña).

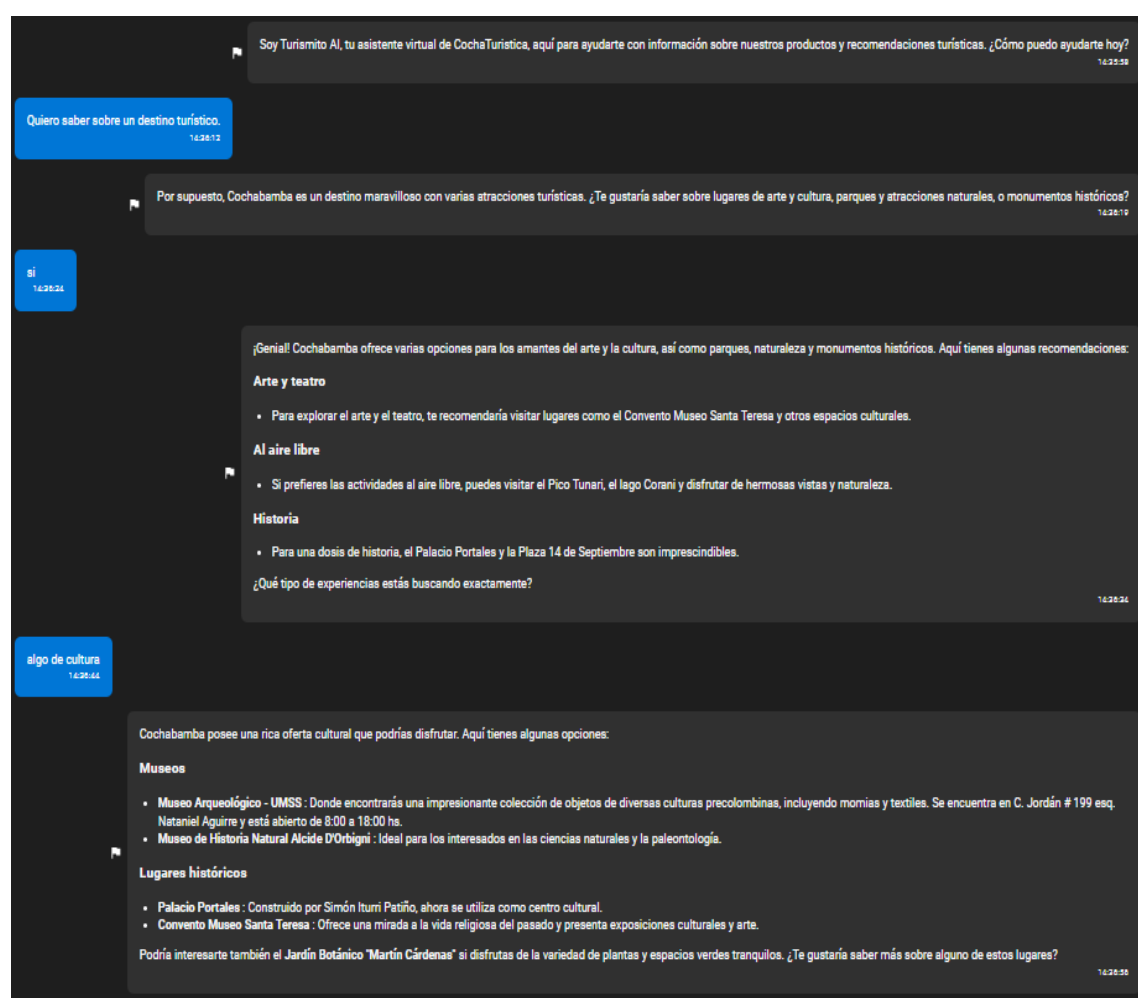
17:03:43

Nota: se muestra en la imagen anterior los distintos datos importantes para saber el clima en el departamento ordenado y claro así mismo se le brinda un enlace de una página web donde podrá encontrar más detalles sobre el clima en determinado lugar. Propuesto por el postulante.

Si se desea que el chatbot te recomiende lugares como de cultura o arte el bot te propondrá ciertos sitios que estén relacionados a lo mencionado.

Figura 60

Conversación casual 5



Nota: Se evidencia la forma en que te recomienda lugares según a tus respuestas y te va dando opciones con detalles cortos del sitios y referentes a la consulta realizada. Propuesto por el postulante.

4.2.4.4. Fluidez de chatbot y pagina web

En la siguiente imagen se puede observar desde la ventana emergente del chatbot que está presente en todas las secciones de la pagina que con un solo click sobre el chat se abrirá y dará inicio a una conversación y estará listo para las consultas que se tenga.

Figura 61

Portada web y chatbot



Nota: Se observa la pagina web de fondo y la ventana abierta del chatbot listo para recibir consultas del usuario la cual se puede cerrar y volver a abrir en cualquier momento de la navegación dentro la pagina web. Propuesto por el postulante.

5 conclusiones y Recomendaciones

5.1 Conclusiones

Una vez finalizado el proyecto, se pueden destacar las siguientes conclusiones:

- ✓ Se ha logrado desarrollar un chatbot eficaz capaz de atender consultas informales de manera precisa y rápida.
- ✓ Se implementó un sistema de web scraping que permite extraer información de fuentes confiables, presentando respuestas claras y concisas al usuario.
- ✓ Se diseñó un chatbot amigable y accesible, facilitando su uso a través de una ventana emergente intuitiva que mejora la experiencia del usuario.
- ✓ Se creó una página web informativa que ofrece contenido claro y relevante sobre los sitios más emblemáticos del departamento, contribuyendo a la difusión del turismo local.

5.2 Recomendaciones

El programa se encuentra en una fase de expansión para abarcar un mayor número de sitios turísticos, ampliando su cobertura más allá del Departamento de Cochabamba. Se contempla la integración de técnicas avanzadas de web scraping, lo cual permitirá explorar y extraer datos de un conjunto más amplio y diverso de fuentes, optimizando así la precisión y actualización de la información proporcionada. Además, el proyecto está diseñado con una arquitectura flexible y modular, lo que facilita la incorporación de nuevas tecnologías que potencien tanto el rendimiento del chatbot como la calidad de los datos obtenidos. La adopción de estas innovaciones tecnológicas se realiza con el objetivo de enriquecer la experiencia del usuario y ofrecer una plataforma más robusta e intuitiva.

Bibliografía

- Alfonzo, Á. (1997).
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- Buitrago, R. S. (junio 2018). *Desarrollo de un asistente virtual para la ciudad de madrid*. madrid.
- Choudhary, A., Singh, P. y Jain, S. (2019). Técnicas y herramientas de *web scraping: una revisión exhaustiva* . Revista internacional de aplicaciones informáticas
- Hammersley, M., y Atkinson, P. (2007). *Etnografía: principios en la práctica* (3.^a ed.). Routledge.
- Instituto Tecnológico de Aragón. (2023, 19 de octubre). *Chatbots o asistentes conversacionales basados en inteligencia artificial y técnicas de procesamiento de lenguaje natural - ITA* . ITA.
- Janert, PK (2010). Análisis de datos con herramientas de código abierto . O'Reilly Media.
- Johnson, L. (2022). *Desarrollo web moderno: marcos y metodologías* . TechPress.
- Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (2nd ed.). Pearson Prentice Hall.
- Kinsta. (2022, 19 de diciembre). *¿Qué es el web scraping? Cómo extraer legalmente el contenido de la web*. yo
- Martín, P. (2023). Evolución del desarrollo web: De páginas estáticas a aplicaciones interactivas avanzadas . Revista de Tecnología Web.
- Mitchell, R. (2018). *Web Scraping con Python: recopilación de más datos de la Web moderna* (2.^a ed.). O'Reilly Media.
- Ornelas, G. (2020). *Diseño e implementacion de un asistente virtual*. Mexico.
- Russell, MA (2013). *Minería de la Web social: minería de datos de Facebook, Twitter, LinkedIn, Google+, GitHub y más* (2.^a ed.). O'Reilly Media.

Ryan, MJ (2019). Automatización de la inteligencia de fuentes abiertas: algoritmos para OSINT . No Starch Press.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer.

W3C. (2017). *Hojas de estilo en cascada (CSS): descripción general* .

Weizenbaum, J. (1966). *ELIZA—A computer program for the study of natural language communication between man and machine*. Communications of the ACM, 9(1), 36–45. <https://doi.org/10.1145/365153.365168>

Yujra, H. A. (2014). *Web Scraping para la obtencion de informacion Actualizada en internet con push notifications para smartphone*. La Paz.

Zakas, NC (2021). Entendiendo a Puppeteer . Publicado de forma independiente.

WEBGRAFÍA

[https://www.it.es/b/chatbots -o -asistencia-conv-basados -en -inteligente-arte-y-
-tecnicas -de -procesamiento -de -lengu-natural/](https://www.it.es/b/chatbots-o-asistencia-conv-basados-en-inteligente-arte-y-tecnicas-de-procesamiento-de-lengu-natural/)

[https://www.ita.es/blog/chatbots -o -asistentes -conversacionales -basados -
en -inteligencia -artificial -y -tecnicas -de -procesamiento -de -lenguaje -natural/](https://www.ita.es/blog/chatbots-o-asistentes-conversacionales-basados-en-inteligencia-artificial-y-tecnicas-de-procesamiento-de-lenguaje-natural/)

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

<https://docs.scrapy.org/en/latest/>

<https://eur-lex.europa.eu/eli/reg/2016/679/oj>

<https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/>

[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)

[US/docs/Learn/Getting started with the web/HTML basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)

<https://react.dev>

https://en.seobi.nordeste/mi/en/Estructura_web

[https://openwebinars.net/blog/conoce -las -3 -metodologias -agiles -mas -
usadas /](https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/)

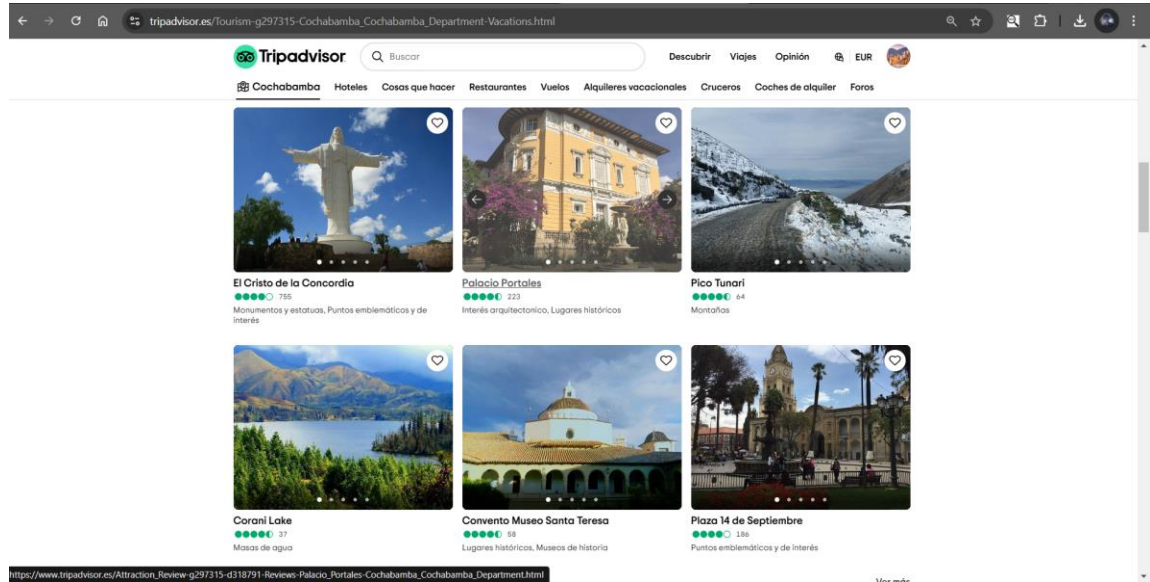
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

[https://developer.mozilla.org/en-US/do/Web /](https://developer.mozilla.org/en-US/docs/Web/)

<https://www.w3.org/Estilo/CSS/>

ANEXOS

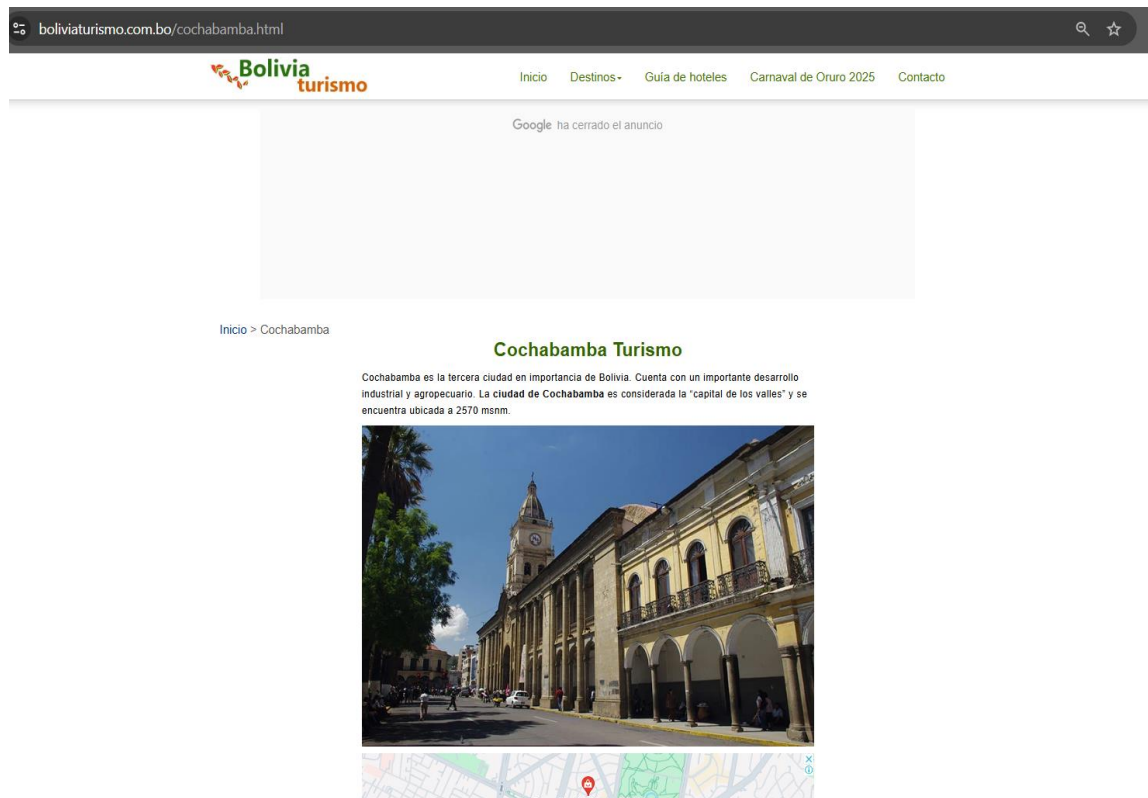
Página web externa tripadvisor de turismo en cochabamba



Página web Bolivia en tus manos de turismo en Cochabamba



Página web Bolivia turismo sección de Cochabamba



Página web Cochabamba turismo



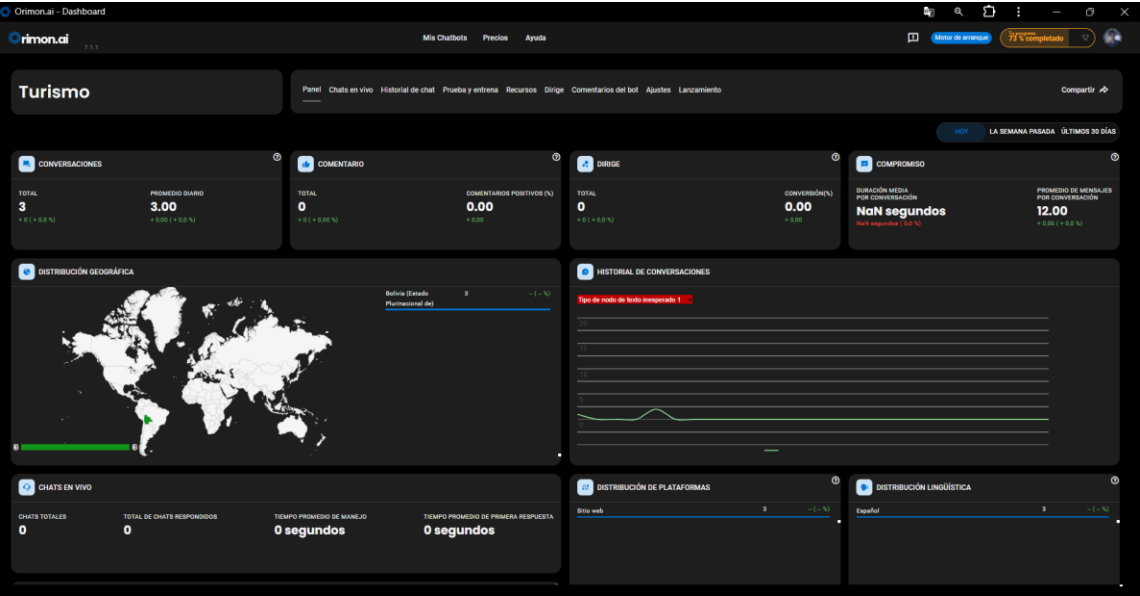
Página web de Cochabamba turismo versión movil



Panel de información de duración promedio por conversación



Panel de información de la herramienta Orimon.Ai



Panel de control de paginas web externas

https://es.wikipedia.org/wiki/Laguna_La_Angostura	Hace 11 horas	🔄	🟢	🗑️	🔒
https://www.accuweather.com/es/bo/cochabamba/32716/hourly-weather-forecast/32716#google_vignette	Hace 10 horas	🔄	🟢	🗑️	🔒
https://www.boliviaentusmanos.com/hoteles-bolivia/ciudad/cochabamba/1/hoteles.html#google_vignette	Hace 12 horas	🔄	🟢	🗑️	🔒
https://www.boliviaentusmanos.com/turismo/atractivos-departamento/cochabamba.html	Hace 6 meses	🔄	🟢	🗑️	🔒
https://www.boliviaentusmanos.com/turismo/atractivos/colina-de-san-sebastian.html	Hace 12 horas	🔄	🟢	🗑️	🔒
https://www.boliviaentusmanos.com/turismo/atractivos/el-pueblito.html	Hace 12 horas	🔄	🟢	🗑️	🔒
https://www.boliviaentusmanos.com/turismo/atractivos/laguna-alalay.html	Hace 12 horas	🔄	🟢	🗑️	🔒
https://www.boliviaentusmanos.com/turismo/atractivos/museo-de-historia-natural-alcide-dorbigny.html	Hace 12 horas	🔄	🟢	🗑️	🔒
https://www.boliviaentusmanos.com/turismo/atractivos/palacio-portales-centro-simon-i-patino.html	Hace 12 horas	🔄	🟢	🗑️	🔒
https://www.boliviaentusmanos.com/turismo/cochabamba/parques-y-espacios-naturales.html	Hace 11 horas	🔄	🟢	🗑️	🔒
https://www.boliviaturismo.com.bo/cochabamba.php	Hace 6 meses	🔄	🟢	🗑️	🔒
https://www.cochabambaturismo.blog.bo/	Hace 9 horas	🔄	🟢	🗑️	🔒

Panel de control de preguntas preprogramadas

¿Dónde está ubicado el Museo Arqueológico - UMSS y cuáles son sus horarios?

El Museo Arqueológico - UMSS está ubicado en C. Jordán # 199 esq. Natali Aguirre y está abierto de 8:00 am a 6:00 pm, solo el día de lunes a sábado

Enterado

¿Dónde se encuentra el cañón tuta wailpa?

El cañón de Tuta Wailpa pertenece a San José, un pueblo subtrófico del municipio de Cotacachi, ubicado a seis horas desde Cochabamba, cuatro se corren en automóvil y otras dos a pie.

Enterado

¿Dónde comer un rico chicharrón en Cochabamba?

para saber más sobre restaurantes mejor clasificados o donde ir a comer platos típicos de Cochabamba te recomiendo visitar en el siguiente enlace https://www.tripadvisor.es/Restaurantes-g297315-Cochabamba_Cochabamba_Department.html

Enterado

¿Dónde se encuentra el palacio portales?

El Palacio Portales está ubicado en el barrio Quera Quera de Cochabamba. Para llegar al Centro Cultural Simón I. Patiño, puede optar por varias opciones En coche: Desde el centro de Cochabamba, tome la Avenida Heroínas hacia el norte y luego la Avenida América hasta llegar a la Avenida Potosí. La mansión está situada en esta avenida. En transporte público: Varias líneas de minibuses y trufas (taxis colectivos) pasan cerca del centro cultural. Puedes preguntar por las rutas que te dejan en la Avenida Potosí. En taxi: Tomar un taxi es una opción conveniente y relativamente económica. Solo necesitas indicar la dirección al conductor.

Enterado

Mapa

¿Dónde puedo reservar un hotel en cochabamba?

Para reservar o ver hoteles en Cochabamba te recomiendo visitar el siguiente enlace: https://www.tripadvisor.es/Hoteles-g297315-Cochabamba_Cochabamba_Department-Hoteles.html

Enterado

¿Que líneas de trufi pasan por ese lugar? o ¿Que trufi me lleva a ese destino?

La mejor aplicación recomendada para saber las rutas o líneas que te llevan a un destino en el departamento de cochabamba es la siguiente: trufi.app O si quieres saber las rutas de las distintas líneas del transporte público en Cochabamba es la siguiente: <https://viata.cochabamba.bo/event/map>

Enterado

¿Qué eventos hay esta semana en Cochabamba?

para saber sobre eventos y ferias en Cochabamba la página de: <https://viata.cochabamba.bo/> te Ayudara de mejor manera con tu consulta.

Enterado

lugares donde comer en cochabamba

Cochabamba es el departamento con la mayor cultura gastronómica del País, pero si gustas saber más sobre restaurantes mejor clasificados o donde ir a comer platos típicos de Cochabamba, te recomiendo visitar el siguiente enlace: [Restaurantes en Cochabamba](#). ¡Buen Provecho!

Enterado

Mapa

Panel de configuración de GPT

Configuración del modelo

Configurar el modelo de IA y la temperatura del bot

Configuración del modelo

Modelo GPT

GPT 4 Turbo

GPT 3.5

GPT 4.0

GPT 4 Turbo

GPT 4o

GPT 4o Mini

Actualizar

Ahorrar

1.0

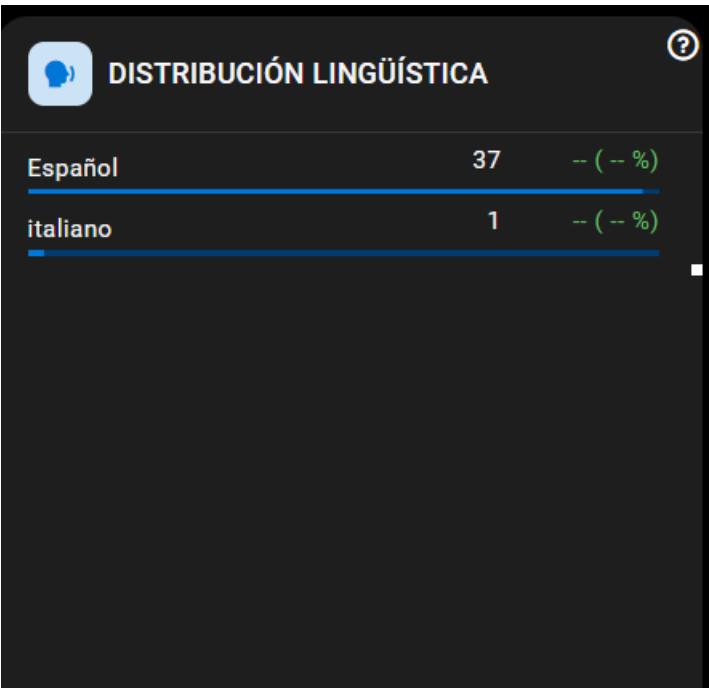
Indicaciones (personalidad)

Define la personalidad y el tono del chatbot

Panel de información del uso del chatbot



Panel de información de idiomas de coversacion



Historial de chats de todos los usuarios

Historial de chats

Oportunidades de entrenamiento de bots

Fecha: 20 de junio de 2024 - 16 de noviembre de 2024

FiltrosRefrescar

Nombre	Resumen	Idioma	Último mensaje
Zarigüya escarata encantada	Descubre la belleza de la laguna Alalay con una breve reseña.	Español	Hace un día
Guacamayo de chocolate clín...	Descubre la belleza de la Laguna Alalay en bicicleta con nuestras reseñas.	Español	Hace un día
Compañera mariposa rosa	Descubre los mejores lugares turísticos y gastronómicos en Cochabamba, como el Cristo de la C...	Español	Hace un día
ardilla indigo considerable	Descubre cómo llegar a la hermosa Laguna Corani y dónde salir a trotar en Cochabamba, incluye...	Español	Hace un día
Bata gris que se pone nervio...	Descubre los mejores lugares de Cochabamba y cómo llegar al Cristo de la Concordia. Recibe ay...	Español	Hace un día
Guacamayo de chocolate clín...	Descubre cómo llegar a la hermosa Laguna Angostura.	Español	Hace un día
lolo rosa calificado	Descubre los mejores lugares turísticos en Cochabamba.	Español	Hace un día
Alce azul analítico	Saludos, ¿en qué puedo ayudarte?	Español	Hace un día
Catálogo contemporáneo d...	Descubre cómo llegar a Cochabamba en solo unos pasos.	Español	Hace un día
Resumen de la mosca del to...	Descubre los mejores guarapos en Cliza con nuestra guía de cómo llegar.	Español	Hace un día
Guacamayo de chocolate clín...	Descubre el clima actual en Cochabamba y planifica tu día con facilidad.	Español	Hace un día
Guacamayo de chocolate clín...	Obtenga el clima actual en Cochabamba de manera rápida y precisa.	Español	Hace un día
Guacamayo de chocolate clín...	Descubre el clima en Cochabamba ahora mismo.	Español	Hace un día
Guacamayo de chocolate clín...	Descubre cómo llegar a Villa Tunari y Corani, y el clima en ambos lugares. También conoce la di...	Español	Hace un día
Guacamayo de chocolate clín...	Descubre el clima actual en Cochabamba y planifica tu día con facilidad.	Español	Hace un día