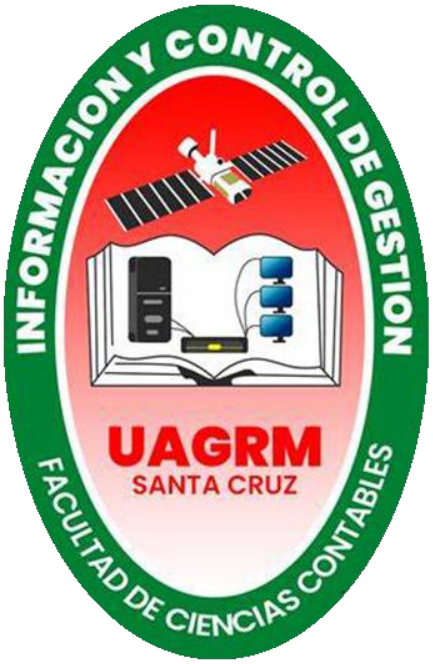
UNIVERSIDAD AUTÓNOMA GABRIEL RENE MORENO

FACULTAD DE CIENCIAS CONTABLES, AUDITORÍA, SISTEMAS DE CONTROL DE GESTIÓN Y FINANZAS



TÍTULO: DESARROLLO DE UN SISTEMA WEB MVC CON INTEGRACIÓN DE INTELIGENCIA ARTIFICIAL PARA LA AUTOMATIZACIÓN DE ATENCIÓN AL CLIENTE Y GESTIÓN DE PEDIDOS EN EL SECTOR RESTAURANTE

INTEGRANTES:

- Valdez Terrezas Luis Darwin
- Dalmanza Chavez Lider Nicolás
- Flores Alba Dayana

Fernández Heredia Rosa Laura

Agreda Daza Jose Daniel

ASIGNATURA: SISTEMAS OPERATIVOS

DOCENTE: PEINADO PEREIRA JUAN CARLOS

ÍNDICE

PORTADA

* Identificación Institucional: Universidad Autónoma Gabriel René Moreno (UAGRM) - Facultad de Ciencias Contables, Auditoría, Sistemas de Control de Gestión y Finanzas

* Logotipo de la carrera de Información y Control de Gestión

* Título del proyecto

* Integrantes

* Asignatura y Docente

* **1.1. INTRODUCCIÓN**

* **1.2. PLANTEAMIENTO DEL PROBLEMA**

* **1.2.1. Identificación del Problema**

* El Fenómeno de la "Inmediatez Insatisfecha"

* Limitaciones de la Capacidad Humana (El Factor 24/7)

* Descentralización de los Datos y Falta de Estructura

* 1.2.2. Formulación del Problema

****1.3. OBJETIVOS****

* 1.3.1. Objetivo General

* 1.3.2. Objetivos Específicos

* **1.4. JUSTIFICACIÓN**

* 1.4.1. Justificación Técnica

* 1.4.2. Justificación Académica

* 1.4.3. Justificación Social y Económica

* **1.5. ALCANCES**

* 1.5.1. Alcance Temático

* 1.5.2. Alcance Geográfico

* 1.5.3. Alcance Temporal

* ***CAPÍTULO II. MARCO TEÓRICO****

* 2.1. DESARROLLO DEL MARCO TEÓRICO

* 2.1.1. Metodologías de Desarrollo Ágil

* 2.1.2. Patrón de Diseño Arquitectónico: Modelo-Vista-Controlador (MVC)

* *Diagrama estructural del flujo MVC*

* Definiciones de Componentes (Modelo, Vista, Controlador)

* *Infografía: Patrones de Arquitectura MVC*

* 2.1.3. Ecosistema de Desarrollo: Paradigma de Lenguaje Unificado

* El Enfoque Full-Stack JavaScript

* Express.JS (Node.Js)

* Ventajas y Desventajas de Express.js

* *Logotipo de Express JS*

* Babel (Transcompilador de JavaScript)

* Ventajas Académicas y Técnicas de Babel

* Express.js (Framework Backend para MVC - Continuación)

* MySQL (Base de Datos Relacional)

* 2.1.4. Inteligencia Artificial y Procesamiento de Lenguaje Natural (PLN)

* **7. PROPUESTA****

* Módulo de Inteligencia Artificial (Chatbot PLN)

* Módulo Administrativo (Back-Office Comercial)

* Módulo Operativo (Monitores de Cocina y Caja)

* **8. CONCLUSIONES**

* **9. RECOMENDACIONES**

* **ANEXOS / CAPTURAS DE PANTALLA**

* **BIBLIOGRAFÍA**

* Sobre Algoritmos de Planificación y Sistemas Operativos (Base Técnica)

* Sobre Inteligencia Artificial y Procesamiento de Lenguaje Natural (PLN)

* Sobre la Adopción Tecnológica y Canales Digitales en PyMEs

* Sobre Gestión de Procesos y Control de Gestión

1.1. INTRODUCCIÓN

En el dinámico ecosistema empresarial actual, la inmediatez se ha transformado de una ventaja competitiva en una exigencia fundamental del consumidor moderno. En el sector gastronómico y de restaurantes de Santa Cruz, las interacciones fluidas y la rapidez en el procesamiento de pedidos dictan el éxito o fracaso comercial. Con más de 2,000 millones de usuarios activos a nivel global, WhatsApp se ha consolidado como el canal de comunicación predilecto entre las marcas y los clientes. Sin embargo, la gestión exclusivamente humana de este volumen de mensajes presenta desafíos críticos en términos de escalabilidad, tiempos de respuesta y costos operativos para los restaurantes.

La implementación de un sistema informático basado en la arquitectura **Modelo-Vista-Controlador (MVC)** utilizando el entorno de ejecución asíncrono **Node.js** e integrado con una herramienta de **Inteligencia Artificial (IA)** surge como la solución estratégica a estos desafíos. A diferencia de los sistemas de respuesta automática tradicionales basados en menús rígidos y numéricos, la integración de IA —específicamente mediante el Procesamiento de Lenguaje Natural (PLN)— permite comprender la intención real del comensal, automatizar la toma de pedidos, gestionar reservas y resolver consultas de forma fluida y conversacional. El presente documento detalla el uso, alcance y desarrollo de esta herramienta, analizando cómo su despliegue optimiza la eficiencia operativa interna del restaurante utilizando un único lenguaje de programación en todo el ecosistema de desarrollo.

1.2. PLANTEAMIENTO DEL PROBLEMA

1.2.1. Identificación del Problema

Las pequeñas y medianas empresas (PyMEs) del sector gastronómico en Santa Cruz enfrentan una paradoja digital: aunque disponen de acceso directo a sus clientes a través de canales digitales como WhatsApp, carecen de la infraestructura y la capacidad operativa para responder con eficiencia al volumen de demanda generado. El problema central radica en las siguientes deficiencias operativas identificadas:

- **El Fenómeno de la "Inmediatez Insatisfecha":** El cliente moderno espera recibir atención o confirmar un pedido en menos de 5 minutos; cualquier tiempo de espera superior se traduce directamente en una pérdida de interés y en el posible abandono del cliente hacia la competencia gastronómica.
- **Limitaciones de la Capacidad Humana (El Factor 24/7):** Los restaurantes dependen frecuentemente de un equipo reducido (o un solo cajero/administrador) para contestar mensajes, procesar pedidos y coordinar con la cocina de forma manual. Esto genera horarios restringidos donde las consultas nocturnas o fuera de hora quedan desatendidas perdiendo ventas potenciales, así como saturación, fatiga humana y errores en la transcripción de las comandas hacia la cocina.
- **Descentralización de los Datos y Falta de Estructura:** Al recibir pedidos por chats de texto informales, la información no se almacena de forma estructurada. No se cuenta con una base de datos automatizada que asocie de inmediato el pedido a una mesa, un cliente o un estado específico de preparación (pendiente, en cocina, listo), retrasando los flujos de trabajo internos del negocio.

1.2.2. Formulación del Problema

A partir de la situación descrita, se establece la siguiente pregunta de investigación:

¿Cómo optimizar el proceso actual de atención al cliente, toma de pedidos y gestión interna en el sector restaurante en la ciudad de Santa Cruz mediante el uso de tecnologías de desarrollo web asíncronas basadas en JavaScript y automatización conversacional inteligente?

1.3. OBJETIVOS

1.3.1. Objetivo General

Desarrollar un sistema web de gestión interna bajo el entorno Node.js basado en el patrón arquitectónico Modelo-Vista-Controlador (MVC), con integración de un Chatbot inteligente basado en Procesamiento de Lenguaje Natural (PLN) para la automatización de la atención al cliente y optimización del flujo de pedidos en un restaurante de Santa Cruz.

1.3.2. Objetivos Específicos

- Identificar los requerimientos y procesos actuales en la recepción de pedidos y flujos de atención al cliente dentro del establecimiento gastronómico.
- Configurar un entorno de desarrollo unificado con JavaScript utilizando Node.js y el transcompilador Babel para el uso de sintaxis moderna y eficiente en el backend.
- Diseñar la arquitectura del sistema mediante diagramas de software (Casos de Uso, Clases, Entidad-Relación) y maquetar las interfaces visuales (vistas de administración y cocina).
- Programar la lógica de negocio del sistema implementando el patrón MVC con Express.js e integrando el componente de IA para el procesamiento automático de mensajes provenientes del cliente.

- Ejecutar pruebas de funcionalidad del sistema web para asegurar la correcta persistencia de los datos, consistencia en los pedidos y la adecuada experiencia de usuario.

1.4. JUSTIFICACIÓN

1.4.1. Justificación Técnica

Desde el punto de vista del desarrollo de software, el proyecto implementa el patrón arquitectónico MVC en un entorno Node.js, lo que garantiza una separación clara de responsabilidades entre el manejo de los datos (Modelos), las interfaces de usuario (Vistas) y las operaciones lógicas de control (Controladores). Técnicamente, el uso de JavaScript unificado tanto en el frontend como en el backend optimiza los recursos de desarrollo. Además, el uso de Babel permite emplear características de JavaScript moderno (como ES6+ Modules), asegurando que el código sea altamente escalable, mantenible y modular mientras mantiene compatibilidad absoluta en producción.

1.4.2. Justificación Académica

El proyecto pone en práctica y consolar de manera empírica conceptos avanzados de la ciencia de la computación. Al desarrollarse sobre Node.js bajo la materia de Sistemas Operativos, se explota de forma directa el modelo de I/O asíncrono y no bloqueante guiado por eventos, permitiendo comprender a nivel práctico el manejo de la concurrencia, hilos de ejecución únicos (Event Loop) y gestión de colas de procesos en tiempo real al ingresar múltiples comandas simultáneas.

1.4.3. Justificación Social y Económica

A nivel socioeconómico, la automatización del proceso de pedidos tiene un impacto directo en las PyMEs gastronómicas de Santa Cruz de la Sierra. Permite reducir los costos operativos al optimizar los recursos humanos disponibles, incrementa las ventas al asegurar una atención continua las 24 horas del día, los 7 días de la semana, y minimiza drásticamente los errores de entrega y los tiempos de espera, aumentando la satisfacción general de la población consumidora.

1.5. ALCANCES

1.5.1. Alcance Temático

El proyecto abarca el análisis, diseño, modelado e implementación de una aplicación web orientada a la gestión de restaurantes. Comprende de forma estricta los siguientes módulos de software:

1. **Módulo de Inteligencia Artificial (Chatbot PLN):** Recepción de mensajes del cliente, interpretación de intenciones (saludo, ver menú, realizar pedido, cancelar), y extracción de entidades (platillos, cantidades, dirección o mesa).
2. **Módulo Administrativo (Back-Office en Node.js/MVC):** Panel de control para el administrador donde se gestionan los usuarios del sistema, se editan los platillos del menú (precios, stock, categorías) y se visualizan los reportes de ventas.
3. **Módulo Operativo (Cocina/Caja):** Interfaz en tiempo real que lista de forma ordenada los pedidos pendientes con sus respectivos estados de preparación, actualizados dinámicamente.

1.5.2. Alcance Geográfico

El estudio, levantamiento de requerimientos y despliegue inicial de la aplicación piloto se limita al municipio de Santa Cruz de la Sierra, Bolivia.

1.5.3. Alcance Temporal

El diseño, desarrollo, pruebas funcionales y documentación del sistema web se ejecutará durante la gestión académica correspondiente al año 2026.

CAPÍTULO II. MARCO TEÓRICO

2.1. DESARROLLO DEL MARCO TEÓRICO

2.1.1. Metodologías de Desarrollo Ágil

Para la gestión del flujo de trabajo de programación se seleccionan los fundamentos de metodologías ágiles como **Kanban**. Kanban se traduce como "tablero visual" y consiste en una herramienta que permite administrar el avance de las tareas mediante tarjetas organizadas en columnas específicas (Por Hacer, En Proceso, Completado). Esta metodología maximiza la eficiencia del desarrollo al identificar cuellos de botella de manera temprana y flexibilizar los cambios según las necesidades del software.

2.1.2. Patrón de Diseño Arquitectónico: Modelo-Vista-Controlador (MVC)

El patrón MVC propone la división estructural de una aplicación de software en tres componentes fuertemente distinguidos, aislando la lógica de negocio de la interfaz de presentación.

[Interfaz de Usuario / Vista]

^ |

Muestra | | Peticiones del

Datos | | Usuario

| v

[Controlador]

^ |

Devuelve | | Manipula /

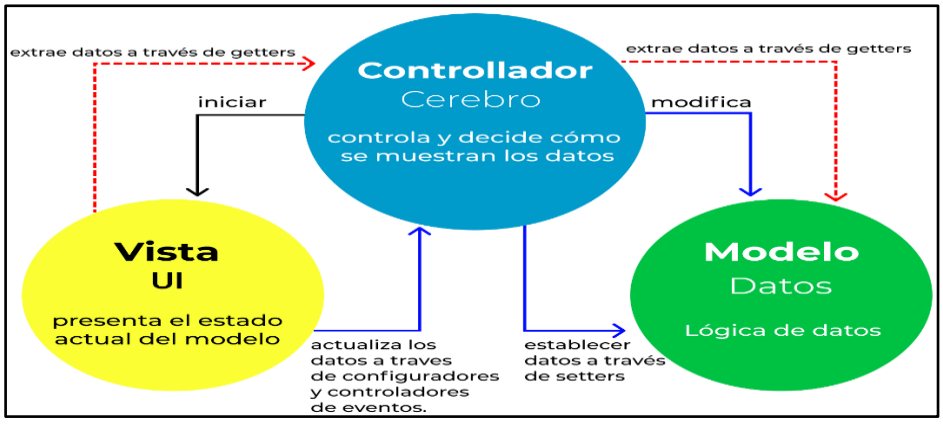
Datos | | Solicita

| v

[Modelo (Datos/Lógica)]

- **Modelo:** Se refiere de manera estricta a la estructura lógica de los datos y las reglas del negocio de la aplicación. Es el encargado de interactuar directamente con el sistema de gestión de bases de datos para consultar, insertar o actualizar la información de los pedidos y el menú, permaneciendo de forma abstracta e independiente de las interfaces de usuario.
- **Vista:** Representa la interfaz gráfica y la presentación visual de los datos procesados al usuario final. En nuestro sistema, corresponde a las interfaces de administración y pantallas de la cocina que renderizan la información de las comandas de manera legible.
- **Controlador:** Actúa como el intermediario directo y coordinador entre la Vista y el Modelo. Es el componente que intercepta las peticiones (por ejemplo, cuando el webhook de la IA procesa un pedido o el chef marca un plato como completado),

solicita la persistencia de datos pertinentes al Modelo y finalmente despacha o redirecciona la respuesta hacia la Vista adecuada.



2.1.3. Ecosistema de Desarrollo: Paradigma de Lenguaje Unificado

- **El Enfoque Full-Stack JavaScript:** Una de las decisiones arquitectónicas principales es el uso de JavaScript como lenguaje único para todo el proyecto. Esto elimina la fricción de cambiar de contexto mental o sintáctico entre el frontend y el backend, permitiendo reutilizar lógica, estructuras de validación de datos (JSON) y agilizando la integración de módulos de manera homogénea.
- **Express.JS (Node.Js)** Express.js, a veces también llamado «Express», es un framework de Backend Node.js minimalista, rápido y similar a Sinatra, que proporciona características y herramientas robustas para desarrollar aplicaciones de Backend escalables. Te ofrece el sistema de enrutamiento y características simplificadas

para ampliar el framework con componentes y partes más potentes en función de los casos de uso de tu aplicación. El framework proporciona un conjunto de herramientas para aplicaciones web, peticiones y respuestas HTTP, enrutamiento y middleware para construir y desplegar aplicaciones a gran escala y preparadas para la empresa. También proporciona una herramienta de interfaz de línea de comandos (CLI) llamada Node Package Manager (NPM), donde los desarrolladores pueden obtener paquetes desarrollados. También obliga a los desarrolladores a seguir el principio de No te repitas (DRY). El principio DRY pretende reducir la repetición de patrones de software, sustituyéndolos por abstracciones, o utilizando normalizaciones de datos para evitar la redundancia. Express.js se utiliza para una amplia gama de cosas en el ecosistema JavaScript/Node.js — puedes desarrollar aplicaciones, endpoints de API, sistemas de enrutamiento y frameworks con él. (KINSTA, 2022)

Ventajas

- Flexible y rápido.
- Escalabilidad.
- Compatible con el motor V8 de Google.
- Apoyo de la comunidad.
- Potente sistema de enrutamiento.

Desventajas

- Difícil manejo de operaciones complejas en un solo hilo.
- Menor robustez en comparación con frameworks como laravel.

- Las dependencias pueden tener problemas de seguridad si no se gestionan bien.
- Necesita librerías externas para funciones básicas.

- **Babel (Transcompilador de JavaScript):** Babel es una herramienta esencial que actúa como un transcompilador de código. Permite escribir el backend utilizando las últimas especificaciones de ECMAScript (sintaxis moderna como import/export, *async/await* avanzados y *arrow functions*), traduciendo este código de vanguardia a una versión de JavaScript compatible hacia atrás (ES5/ES6 estándar) que Node.js ejecuta de manera completamente estable y óptima.

A diferencia de un compilador tradicional (como gcc para C) que toma código fuente en lenguaje de alto nivel y lo traduce a código de máquina binario de bajo nivel ejecutable por la CPU, Babel es un transcompilador o "transpiler".

Ventajas y Desventajas de Babel en la Arquitectura

Ventajas Académicas y Técnicas:

- Mantenibilidad del Código: Permite al equipo de desarrollo utilizar las abstracciones más limpias del lenguaje, reduciendo la cantidad de líneas de código escritas manualmente (principio DRY).

- Futuro Asegurado (Future-Proofing): Si el próximo año el estándar de JavaScript añade una nueva funcionalidad para optimizar el procesamiento de lenguaje natural, el código del sistema web se puede adaptar de inmediato simplemente agregando un nuevo plugin a Babel, sin tener que reescribir toda la aplicación desde cero.

- Unificación Frontend/Backend: Al usar Babel en el backend, tanto las vistas del navegador como el controlador de Node.js comparten exactamente la misma sintaxis de módulos (import/export), eliminando la confusión mental del programador full-stack.

- **Express.js (Framework Backend para MVC):** Express es un framework minimalista y flexible para Node.js. Proporciona un conjunto robusto de características para el enrutamiento y middleware, permitiendo estructurar el patrón de diseño MVC de forma manual y limpia organizando los directorios en controladores de rutas, modelos de persistencia y renderizado de vistas.

- **MySQL (Base de Datos Relacional):** Sistema de gestión de bases de datos relacional seleccionado por su excelente rendimiento en consultas simultáneas, integridad referencial mediante llaves foráneas y consistencia estricta en las transacciones comerciales de cobro y pedidos de comida.

2.1.4. Inteligencia Artificial y Procesamiento de Lenguaje Natural (PLN)

La Inteligencia Artificial aplicada a la atención conversacional se fundamenta en el uso de algoritmos de PLN. A diferencia de las estructuras de código lineal estándar, los modelos de PLN se encargan de tokenizar el texto plano enviado por el usuario, analizar sintácticamente las frases, clasificar la **intención** de la consulta (ej. hacer_pedido) y extraer **entidades** específicas claves (ej. cantidad: 2, producto: Silpancho). Esto permite una comunicación fluida donde el sistema interactúa de forma orgánica con el ser humano e inyecta la información ya estructurada directamente hacia el controlador en Node.js.

7. PROPUESTA

Se propone el diseño, desarrollo e implementación de un sistema informático integral basado en una arquitectura web Modelo-Vista-Controlador (MVC) utilizando el entorno de ejecución asíncrono Node.js, integrado con un módulo de Inteligencia Artificial (IA) especializado en Procesamiento de Lenguaje Natural (PLN). Esta solución está estratégicamente orientada a automatizar de forma conversacional el canal de comunicación de WhatsApp, resolviendo de manera directa el fenómeno de la "inmediatez insatisfecha" y optimizando los

flujos de trabajo operativos internos del sector gastronómico en el municipio de Santa Cruz de la Sierra.

El núcleo de la propuesta se fundamenta en el paradigma de desarrollo Full-Stack JavaScript, empleando el framework Express.js y el transcompilador Babel para estructurar una plataforma robusta, modular y escalable. A nivel funcional, el sistema propuesto despliega su impacto a través de tres componentes operativos de software:

- **Módulo de Inteligencia Artificial (Chatbot PLN):** Actúa como la interfaz conversacional primaria las 24 horas del día, los 7 días de la semana. Este componente se encarga de recibir los flujos de mensajería asíncrona provenientes de los comensales, tokenizar el texto plano, interpretar de forma orgánica la intención real del usuario (saludos, solicitudes de menú, consultas frecuentes o cancelaciones) y extraer entidades estructuradas fundamentales (identificación de platillos, cantidades y asignación de mesas o direcciones de entrega).
- **Módulo Administrativo (Back-Office Comercial):** Proporciona un panel de control centralizado y seguro bajo el patrón MVC, diseñado específicamente para que los administradores y gerentes del restaurante realicen la gestión y mantenimiento del catálogo de productos (precios, stock y categorías), administren los roles de usuario y accedan a reportes analíticos de ventas integrados con un sistema de base de datos relacional MySQL.
- **Módulo Operativo (Monitores de Cocina y Caja):** Despliega una interfaz gráfica interactiva que se actualiza dinámicamente en tiempo real mediante el manejo concurrente de peticiones. Esta vista clasifica, ordena y encola los pedidos

estructurados por el chatbot según su estado de preparación (pendiente, en cocina, listo), eliminando por completo la transcripción manual de comandas y centralizando los datos operativos.

Con la implementación de esta propuesta tecnológica, se busca dotar a las pequeñas y medianas empresas (PyMEs) gastronómicas de la región de una infraestructura digital avanzada. El despliegue de esta arquitectura no solo disminuye significativamente los errores en la toma de pedidos y los costos de operación, sino que transforma la experiencia de atención al cliente, maximizando la capacidad de respuesta simultánea del negocio mediante el aprovechamiento eficiente de los hilos de ejecución y el bucle de eventos (Event Loop) provistos por el sistema operativo del servidor.

8. CONCLUSIONES

- El análisis detallado sobre los procesos de atención al cliente y gestión de pedidos en las PyMEs gastronómicas de Santa Cruz de la Sierra evidencia que los métodos tradicionales basados en interacciones puramente humanas o menús numéricos rígidos presentan limitaciones críticas de escalabilidad. Esto genera retrasos operativos severos frente a la alta demanda concurrente y produce el fenómeno de la "inmediatez insatisfecha", impactando negativamente en la retención de clientes y la rentabilidad del negocio.
- Se determinó que la implementación del patrón de diseño arquitectónico Modelo-Vista-Controlador (MVC) en un entorno unificado de desarrollo con JavaScript

(Node.js, Express.js y Babel) constituye una solución técnica altamente eficiente. Esta estructura garantiza el aislamiento completo entre las reglas de negocio de los datos (Modelos), las interfaces de presentación interactiva (Vistas) y los componentes de despacho lógico (Controladores), facilitando el mantenimiento y la escalabilidad del software en producción.

- La integración de algoritmos de Inteligencia Artificial mediante Procesamiento de Lenguaje Natural (PLN) demuestra ser una alternativa superadora frente a los sistemas lineales automatizados. La capacidad del modelo para tokenizar y extraer entidades de texto informal permite estructurar de forma automática los datos de las comandas informales directamente hacia las bases de datos de MySQL, optimizando los flujos de trabajo tradicionales de las áreas de cocina y caja.
- Desde una perspectiva de la teoría de Sistemas Operativos, la explotación del modelo de entrada/salida (I/O) asíncrono y no bloqueante de Node.js convalida de manera práctica el manejo óptimo de la concurrencia masiva. El procesamiento eficiente de múltiples peticiones simultáneas de clientes mediante un único hilo de ejecución (Event Loop) previene el colapso de los recursos del servidor informático, demostrando que la automatización conversacional inteligente es viable, robusta y económicamente sostenible para elevar la competitividad del sector gastronómico regional.

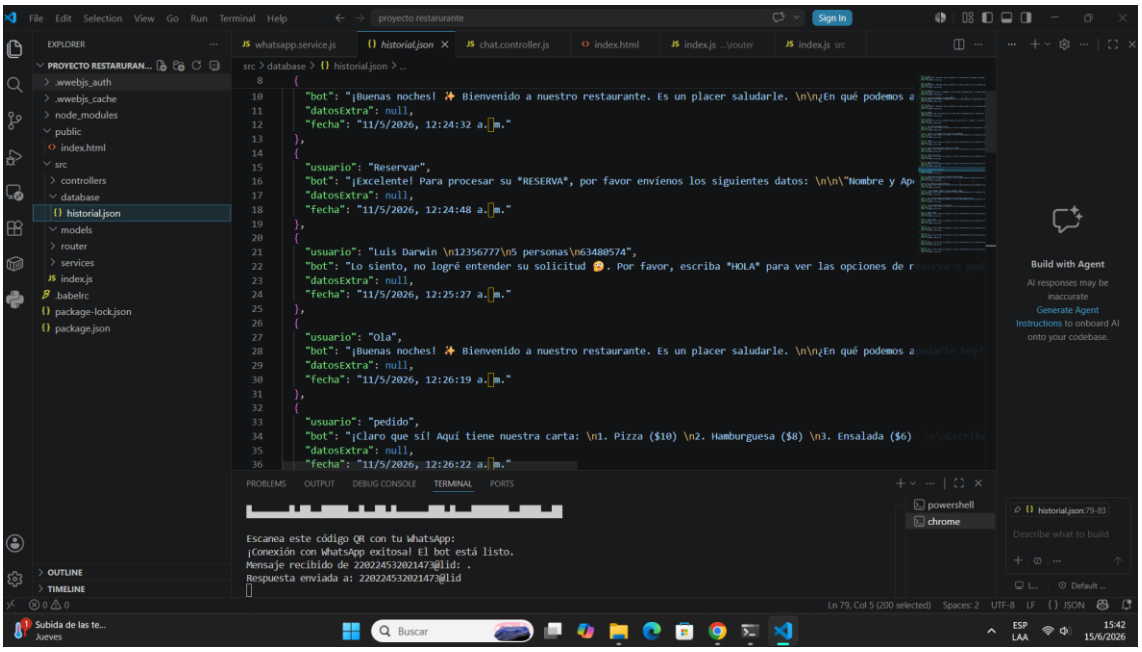
9. RECOMENDACIONES

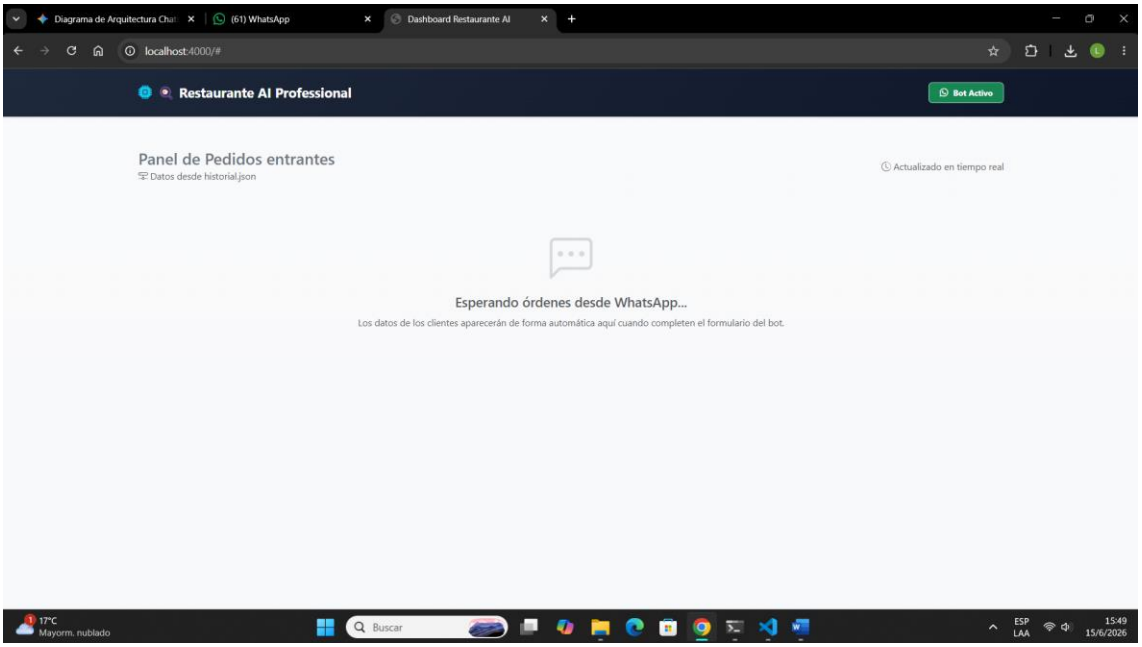
- Se recomienda al sector de restaurantes y PyMEs gastronómicas de Santa Cruz de la Sierra adoptar e incorporar sistemas web integrados con arquitecturas de automatización e Inteligencia Artificial, con el fin de migrar los canales de atención

tradicionales hacia un modelo conversacional asíncrono y permanente (24/7) que mitigue la pérdida de ventas potenciales fuera de los horarios comerciales.

- Se sugiere mantener una estructura rígida en la separación de responsabilidades de software mediante el patrón arquitectónico MVC. Esto permitirá realizar actualizaciones modulares e independientes en las interfaces gráficas del personal de cocina o administración, sin que estas alteraciones afecten la integridad referencial y las reglas de negocio configuradas en los modelos de base de datos.
- Se aconseja profundizar de manera continua en el entrenamiento y refinamiento de los modelos de Procesamiento de Lenguaje Natural (PLN) que alimentan al chatbot, incorporando modismos lingüísticos y variables locales propias del habla cruceña en el sector de alimentos. Esto garantizará un incremento sostenido en la precisión de la extracción de entidades y el reconocimiento de platillos singulares del menú.
- Es altamente recomendable ejecutar de manera sistemática pruebas de carga, concurrencia y persistencia de datos sobre el entorno backend en Node.js, simulando picos elevados de demanda en horas de alta concurrencia en el restaurante. De este modo se podrá auditar el comportamiento del bucle de eventos del sistema operativo del servidor y asegurar una sincronización impecable en tiempo real con las pantallas de cocina.
- Finalmente, se sugiere implementar planes de capacitación técnica orientados al personal operativo del restaurante (cajeros, administradores y personal de cocina) para facilitar la transición hacia el uso del nuevo panel de gestión digital (*Back-*

Office), minimizando la curva de aprendizaje y promoviendo una cultura de control de procesos basada en el almacenamiento estructurado de datos comerciales.





BIBLIOGRAFÍA

1. **Sobre Algoritmos de Planificación y Sistemas Operativos (Base Técnica):**

- Silberschatz, A., Galvin, P. B., y Gagne, G. (2019). *Sistemas operativos* (10a ed.). Editorial Limusa.
- Stallings, W. (2014). *Sistemas operativos: Aspectos internos y principios de diseño* (7a ed.). Pearson Educación.
- Tanenbaum, A. S., y Bos, H. (2015). *Sistemas operativos modernos* (4a ed.). Pearson Educación.

2. **Sobre Inteligencia Artificial y Procesamiento de Lenguaje Natural (PLN):**

- Russell, S., y Norvig, P. (2021). *Inteligencia artificial: Un enfoque moderno* (4a ed.). Pearson Educación.
- Jurafsky, D., y Martin, J. H. (2023). *Speech and Language Processing* (3rd ed. draft). Stanford University.

3. **Sobre la Adopción Tecnológica y Canales Digitales en PyMEs:**

- Comisión Económica para América Latina y el Caribe [CEPAL]. (2024). *El estado de la digitalización de las MIPYMES en América Latina y el Caribe: Desafíos y oportunidades de la inteligencia artificial*. Naciones Unidas.
- Microsoft Latinoamérica. (2024). *La IA y las PyMEs: Un análisis de su adopción e impacto en la región*. Microsoft News.
- Infobip. (2025). *El impacto de las plataformas de mensajería conversacional en la experiencia del cliente (CX)*. Infobip Insights.

4. **Sobre Gestión de Procesos y Control de Gestión (Área de tu Facultad):**

- Chase, R. B., Jacobs, F. R., y Aquilano, N. J. (2018). *Administración de operaciones: Producción y cadena de suministros* (14a ed.). McGraw-Hill.
- Kaplan, R. S., y Norton, D. P. (2008). *The Execution Premium: Integrando la estrategia y las operaciones para lograr ventajas competitivas*. Deusto.