



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN**

Aplicación Web para gestionar registros de comandas para alimentos y bebidas en el restaurante Bonny utilizando la metodología Ágil Lean

Trabajo de Titulación para optar al título de Ingeniero en Tecnologías de la Información

Autor:

Caranqui Caranqui, Jefferson Rolando

Tutor:

Ing. Pamela Alexandra Buñay Guisñan, Mgs.

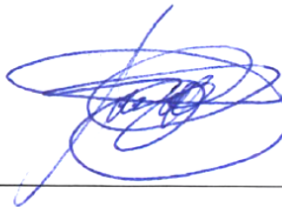
Riobamba, Ecuador. 2025

DECLARATORIA DE AUTORÍA

Yo, Jefferson Rolando Caranqui Caranqui, con cédula de ciudadanía 0605260975, autor del trabajo de investigación titulado: Aplicación web para gestionar registros de comandas para alimentos y bebidas en el Restaurante Bonny utilizando la metodología ágil LEAN, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, a los 2 días del mes de abril del 2025.



Jefferson Rolando Caranqui Caranqui

C.I: 0605260975



ACTA FAVORABLE - INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN

En la Ciudad de Riobamba, a los 14 días del mes de abril de 2025, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por el estudiante **JEFFERSON ROLANDO CARANQUI CARANQUI** con CC: **0605260975**, de la carrera **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado "**APLICACIÓN WEB PARA GESTIONAR REGISTROS DE COMANDAS PARA ALIMENTOS Y BEBIDAS EN EL RESTAURANTE BONNY UTILIZANDO LA METODOLOGÍA ÁGIL LEAN**", por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



Firmado electrónicamente por:
PAMELA ALEXANDRA
BUNAY GUISNAN

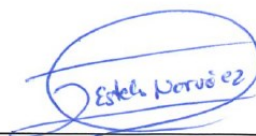
Mgs. Pamela Buñay
TUTORA

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

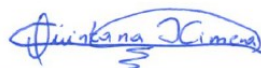
Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación: Aplicación web para gestionar registros de comandas para alimentos y bebidas en el Restaurante Bonny utilizando la metodología ágil LEAN, presentado por Jefferson Rolando Caranqui Caranqui, con cédula de identidad número 0605260975, bajo la tutoría de Mgs. Pamela Alexandra Buñay Guisñan; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba a los 21 días del mes de mayo de 2025.

Miryan Narváez, PhD.
PRESIDENTE DEL TRIBUNAL DE GRADO



Ximena Quintana, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO



Danny Velasco, Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO





CERTIFICACIÓN

Que, **JEFFERSON ROLANDO CARANQUI CARANQUI** con CC: **0605260975**, estudiante de la Carrera **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**APLICACIÓN WEB PARA GESTIONAR REGISTROS DE COMANDAS PARA ALIMENTOS Y BEBIDAS EN EL RESTAURANTE BONNY UTILIZANDO LA METODOLOGÍA ÁGIL LEAN**", cumple con el **2%**, de acuerdo al reporte del sistema Anti plagio **COMPILATIO**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 12 de mayo de 2025



Mgs. Pamela Buñay
TUTORA

DEDICATORIA

Con el corazón lleno de gratitud, dedico este logro a quienes han sido pilares fundamentales en mi camino universitario.

A mi madre, **Ines Caranqui**, fuente inagotable de inspiración y fortaleza. Transformar al ser que soy hoy ha sido una labor que has realizado con devoción y sabiduría. Contempla ahora el resultado de tu entrega. Te prometo corresponder no sólo con afecto, sino convirtiéndome en tu respaldo y motivo de orgullo en cada nueva etapa. Este triunfo te pertenece más a ti que a mí. Has sido mi guía ejemplar; sembraste en mí principios inquebrantables y el valor para perseguir mis anhelos, apoyando mis elecciones aun en circunstancias adversas. Por ti, hoy no sólo experimento felicidad, sino también claridad en mis propósitos. Cada conquista mía es y será eternamente un tributo a tu nombre. Infinitas gracias, mamá.

A mi padre, **Raúl**, por tu respaldo constante y tu influencia formativa. Has constituido una parte esencial de nuestra familia, contribuyendo significativamente a nuestro desarrollo y bienestar. Tu esfuerzo y compromiso son invaluable y mi agradecimiento hacia ti será eterno. A mis hermanos **Anderson Paul** y **Mijahel Soveroth**, por su aliento continuo y su compañía en momentos cruciales. Valoro profundamente su cariño y su presencia, representando mi soporte cuando más lo he requerido y compartiendo cada victoria y reto.

A mi mejor amigo, Jimson, por su lealtad es un tipazo de persona. Agradezco cada instante compartido, las alegrías vividas y tu presencia tanto en celebraciones como en dificultades.

A todos quienes han sumado de alguna forma a la consecución de esta meta, mi sincero agradecimiento. Este recorrido ha sido posible gracias a cada uno de ustedes, dejando una marca profunda en mi historia personal.

AGRADECIMIENTO

En primer lugar, deseo expresar mi más profunda gratitud a Dios, quien me ha concedido la fortaleza, sabiduría y salud necesarias para llevar a cabo y culminar exitosamente este proyecto de investigación.

Mi reconocimiento más sincero y emotivo va dirigido a mi madre, Inés Caranqui, cuyo amor incondicional, sacrificio constante y apoyo inquebrantable han sido el pilar fundamental en cada etapa de este camino académico. Mamá, agradezco a Dios por permitirme tenerte cada día de mi vida, pues sin tu presencia y aliento este logro no habría sido posible. A mis hermanos, les agradezco profundamente por su constante motivación y por acompañarme en cada momento de este proceso, siendo siempre una fuente de inspiración y fortaleza.

Extiendo mi gratitud a mis amigos, quienes durante toda mi trayectoria universitaria me brindaron su invaluable compañerismo, lealtad y amistad sincera. Juntos creamos momentos inolvidables que iluminaron mis días e hicieron más llevadero este recorrido académico. Su presencia ha sido fundamental para mi crecimiento personal y profesional.

Reconozco también la importancia del trabajo que desempeñé como mesero durante mis años universitarios, el cual representó un apoyo económico esencial que me permitió sostener mis estudios y avanzar hacia la consecución de esta meta. Esta experiencia laboral no solo fue un sustento financiero, sino también una valiosa escuela de vida que fortaleció mi carácter y determinación.

De manera especial, expreso mi más sincero agradecimiento a la Ingeniera Pamela Buñay, por su orientación profesional, dedicación y contribuciones significativas a este trabajo de investigación. Su guía experta y apoyo constante fueron elementos clave para el desarrollo y culminación exitosa de este proyecto.

Finalmente, agradezco a todas las personas que, de una u otra manera, contribuyeron a hacer realidad este sueño académico. Cada gesto de apoyo, cada palabra de aliento y cada muestra de confianza han sido fundamentales en este proceso de crecimiento y aprendizaje.

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA	
DICTAMEN FAVORABLE DEL PROFESOR TUTOR	
CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL	
CERTIFICADO ANTIPLAGIO	
DEDICATORIA	
AGRADECIMIENTO	
ÍNDICE GENERAL	
ÍNDICE DE TABLAS	
ÍNDICE DE FIGURAS	
RESUMEN	
ABSTRACT	

CAPÍTULO I. INTRODUCCIÓN.....	14
1.1 Planteamiento del Problema	15
1.2 Justificación	15
1.3 Formulación del problema	16
1.4 Objetivos.....	16
CAPÍTULO II. MARCO TEÓRICO	17
2.1 Metodología ágil	17
2.2 Principios fundamentales.....	17
2.3 Metodologías tradicionales vs. metodologías ágiles	17
2.4 Metodología ágil LEAN	18
2.4.1 Origen y fundamentación.....	18
2.4.2 Principios y prácticas	19
2.4.3 Fases de la metodología LEAN.....	20
2.4.4 Metodología LEAN para el desarrollo de software.....	21
2.4.5 Comparación de LEAN con otras metodologías ágiles de desarrollo	22
2.5 Frameworks y lenguajes para desarrollo web.....	23
2.5.1 Django.....	23
2.5.2 Comparativa con otras tecnologías (PHP, Ruby on Rails)	24
2.6 Bases de datos y su relevancia en la gestión de datos de negocios	25
2.6.1 Bases de datos relacionales: MySQL	25
2.6.2 Escalabilidad y rendimiento de bases de datos	25
2.7 Calidad del software	26

2.7.1	Modelo FURPS	27
2.8	JMeter	27
CAPÍTULO III. METODOLOGÍA		28
3.1	Tipo de investigación	28
3.1.1	Según la fuente de la investigación	28
3.1.2	Según el objeto de estudio	28
3.2	Diseño de la investigación	28
3.3	Población de estudio y tamaño de la muestra	28
3.4	Técnicas de recolección de datos	28
3.5	Identificación de variables	29
3.5.1	Variable independiente	29
3.5.2	Variable dependiente	29
3.6	Operacionalización de variables	29
3.7	Metodología de desarrollo	31
3.7.1	Definición del Valor	31
3.7.2	Mapeo de la Cadena de Valor	34
3.7.3	Creación de Flujo.....	35
3.7.4	Producción bajo demanda (Pull)	41
3.7.5	Perfeccionamiento continuo (Kaizen)	45
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....		48
4.1	Resultados.....	48
4.2	Discusión.....	54
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....		55
5.1	Conclusiones	55
5.2	Recomendaciones.....	56
BIBLIOGRAFÍA		57
ANEXOS		59

ÍNDICE DE TABLAS

Tabla 1: Principios de LEAN en Desarrollo de Software.....	21
Tabla 2: Comparación de LEAN.....	22
Tabla 3: Métricas de Rendimiento del Modelo FURPS.....	27
Tabla 4: Operacionalización de variables	30
Tabla 5: Requisitos funcionales de la aplicación web	31
Tabla 6: Requisitos no funcionales de la aplicación web	32
Tabla 7: Roles de la aplicación	33
Tabla 8: Planificación de actividades	35
Tabla 9: Tabla auth_user.....	38
Tabla 10: Tabla auth_role	38
Tabla 11: Tabla auth_user_groups	38
Tabla 12: Tabla gestion_mesa	39
Tabla 13: Tabla gestion_producto.....	39
Tabla 14: Tabla gestion_pedido	40
Tabla 15: Tabla gestion_itempedido	40
Tabla 16: Mediciones de JMeter	48
Tabla 17: Requerimientos solicitados para eficacia.....	51
Tabla 18: Valores obtenidos modelo FURPS	53

ÍNDICE DE FIGURAS

Figura 1: Principios LEAN	19
Figura 2: Prácticas LEAN	19
Figura 3: Fases LEAN	21
Figura 4: Framework Django	24
Figura 5: Comparativos de Frameworks.....	25
Figura 6: Flujo actual de comandas restaurante Bonny.....	34
Figura 7: Flujo optimizado.....	34
Figura 8: Diagrama de componentes	35
Figura 9: Casos de uso	36
Figura 10: Diagrama de clase.....	37
Figura 11: Diagrama de base de datos.....	37
Figura 12: Modelado de interfaz aplicación web	41
Figura 13: Modelado de interfaz aplicación móvil.....	41
Figura 14: Arquitectura aplicación web.....	42
Figura 15: Arquitectura aplicación móvil para meseros	42
Figura 16: Interfaz gráfica de la aplicación móvil.....	43
Figura 17: Gestión de caja aplicación web	43
Figura 18: Pantalla de hacer pedido	43
Figura 19: Código aplicación web.....	44
Figura 20: Gestión de mesas	44
Figura 21: Gestión de categorías de productos	44
Figura 22: Pedido por mesa.....	45
Figura 23: Información del cliente	45
Figura 24: Transferir pedido a otra mesa.....	46
Figura 25: Generar factura	46
Figura 26: Pruebas realizadas al sistema	47
Figura 27: Plan de pruebas.....	49
Figura 28: View Results in Table	49
Figura 29: Thread Group	50
Figura 30: Tiempos de respuesta.....	51
Figura 31: Rendimiento del CPU	52
Figura 32: Utilización de la memoria	52
Figura 33: Métricas de lectura del disco	52
Figura 34: Velocidad de la red	53

RESUMEN

En la actualidad, el desarrollo e implementación de aplicaciones web en el ámbito gastronómico se ha convertido en una necesidad, las herramientas tecnológicas como: Python, Django y MySQL optimizan los procesos operativos, mejoran la gestión y elevan la calidad del servicio al cliente. El restaurante Bonny enfrenta limitaciones debido a un sistema de comandas obsoleto, ocasionando problemas de compatibilidad, fallos frecuentes y tiempos prolongados en la atención. Para abordar estas deficiencias, se desarrolló una aplicación web que automatiza la gestión de comandas, reduce errores y optimiza los recursos disponibles.

El objetivo central de esta investigación fue el desarrollo de una aplicación web destinada a la gestión de comandas, implementada mediante el framework Django y guiada por los principios de la metodología ágil LEAN. Para evaluar el rendimiento del sistema, se adoptó el modelo de calidad FURPS, el cual permitió analizar dimensiones clave como la funcionalidad, usabilidad, confiabilidad, desempeño y soporte técnico.

Las pruebas de rendimiento se llevaron a cabo utilizando Apache JMeter, simulando distintos escenarios de carga para analizar métricas clave como la eficacia, el tiempo de respuesta y el consumo de recursos. Los resultados obtenidos evidenciaron una eficacia del 99%, con un tiempo de respuesta promedio de 4600 ms y un uso de recursos del 25%. Estos datos sugieren que el sistema no solo cumple con los estándares definidos por el modelo FURPS, sino que también presenta un rendimiento robusto y una base sólida para futuras mejoras y ampliaciones.

Palabras claves: LEAN, comandas, FURPS, Django, Jmeter.

ABSTRACT

Developing and implementing web applications in the gastronomy sector has become a necessity. Technological tools like Python, Django, and MySQL optimize operational processes, improve management, and enhance customer service quality. The Bonny restaurant faces limitations due to an obsolete ordering system, which causes compatibility problems, frequent failures, and prolonged service times. To address these deficiencies, developing a web application that automates order management, reduces errors, and optimizes available resources was necessary. The central objective of this research was the development of a web application for order management, implemented using the Django framework and guided by the principles of the LEAN agile methodology. The researcher adopted the FURPS quality model to evaluate system performance, which allowed for the analysis of key dimensions such as functionality, usability, reliability, performance, and technical support. It was vital to conduct the performance testing using Apache JMeter, simulating different load scenarios to analyze key metrics such as efficiency, response time, and resource consumption. The results showed an efficiency of 99%, with an average response time of 4,600 ms and a resource usage of 25%. This data suggests that the system not only meets the standards defined by the FURPS model but also presents robust performance and a solid foundation for future improvements and expansions.

Keywords: LEAN, Order, FURPS, Django, JMeter.



Reviewed by:

Mgs. Jessica María Guaranga Lema

ENGLISH PROFESSOR

C.C. 0606012607

CAPÍTULO I. INTRODUCCIÓN

Actualmente, las aplicaciones web se utilizan ampliamente en todos los campos laborales debido a su accesibilidad. Estas facilitan a empresas grandes y pequeñas optimizar el desarrollo de productos y mejorar su funcionamiento. Por ello, muchas organizaciones implementan aplicaciones web para asegurar la calidad y eficiencia de sus operaciones [1]. Este avance está ligado a la aparición de nuevas tecnologías y frameworks. Además, la adopción de metodologías ágiles para el desarrollo web ha sido fundamental en este contexto, debido a que permitió a las empresas responder rápidamente a las demandas del mercado y mejorar la eficiencia en sus procesos de desarrollo [2].

La metodología LEAN ha contribuido significativamente al desarrollo de aplicaciones más rápidas y eficientes, enfocándose en maximizar el valor para el cliente mediante la eliminación de desperdicios y la optimización de todos los procesos de producción [3]. La metodología LEAN, originada en el sistema de producción de Toyota, pone énfasis en la mejora continua y en la optimización integral de los procesos con el fin de incrementar la eficiencia y el valor del producto final. En el ámbito del desarrollo de software, su integración con enfoques ágiles como Kanban permite una entrega más ágil y eficiente de aplicaciones web, facilitando a las organizaciones responder con rapidez a los cambios y demandas del mercado [4].

La presente investigación responde a la necesidad del restaurante Bonny de actualizar su sistema de registro de comandas, el cual se encuentra obsoleto y ha dejado de utilizarse. Para abordar esta problemática, se desarrolló una aplicación web basada en el framework Django. Asimismo, se implementó la metodología ágil LEAN para guiar el proceso de desarrollo y se aplicó el modelo de calidad FURPS como herramienta de evaluación del rendimiento del sistema.

La estructura del presente trabajo de titulación se organiza en cinco capítulos. El Capítulo I presenta el problema de investigación. En el Capítulo II se exponen los fundamentos teóricos. El Capítulo III describe la metodología y herramientas aplicadas en el desarrollo. En el Capítulo IV se presentan los resultados obtenidos en la evaluación del sistema, con los criterios del modelo FURPS. Finalmente, el Capítulo V presenta las conclusiones y recomendaciones del trabajo realizado.

1.1 Planteamiento del Problema

El sistema de para registro de comandas del restaurante Bonny en la ciudad de Riobamba actualmente, se encuentra retrasado tecnológicamente. Se creó en base a la tecnología .NET Framework 4.6, actualmente esta plataforma ha quedado obsoleta por los avances tecnológicos y lo más importante es que no hay soporte adecuado. Motivo por el que el sistema ha presentado problemas de compatibilidad, dificultando su integración con tecnologías más recientes, provocando disminución de su rendimiento.

Otro inconveniente es que el sistema tiene muchos fallos que interrumpen el funcionamiento, obligando al personal a reiniciar los equipos y volver a ejecutar la aplicación de manera constante. Esta problemática genera pérdida de tiempo al momento de dar el servicio, afectando la eficiencia operativa del restaurante y causando mucha demora en la atención al cliente. Además, el personal se ve en la necesidad de realizar acciones adicionales de mantenimiento y soporte técnico, lo que no está dentro de sus responsabilidades principales.

El elevado consumo de recursos del sistema actual es otro de los problemas, esto provoca sobrecargas en los equipos y reduce su vida útil. Estas inconvenientes afectan directamente en la experiencia del cliente, por los incrementos de tiempos de espera, reduciendo la calidad del servicio. Estos problemas al presentarse de manera frecuente orillan al restaurante a la necesidad urgente de usar un sistema moderno y optimizado que contribuya con mejoras en las actividades del restaurante Bonny.

1.2 Justificación

Con el objetivo de solucionar las problemáticas antes mencionadas, se desarrolló un nuevo sistema para registro de comandas usando tecnologías actuales y con buen rendimiento. Para hacerlo posible, se usó el framework Django en su versión última versión, se eligió esta framework gracias a su solidez, seguridad y escalabilidad. Estas cualidades, además a su capacidad para facilitar el desarrollo de aplicaciones web de buena calidad, lo convierten en una buena opción para cumplir con los objetivos.

El uso del framework Django se complementó con el gestor de bases de datos MySQL, lo que hizo posible un manejo eficiente y seguro de la información. Con estas herramientas tecnológicas no solo se optimizó el rendimiento general del sistema, sino que también se garantizó la protección de los datos en contexto restaurante y clientes. MySQL, al ser estable y compatible con Django, hizo más sencilla la integración garantizando mantenimiento sostenible a largo plazo.

1.3 Formulación del problema

¿La utilización de la metodología ágil LEAN incidirá en el rendimiento de la aplicación web para la gestión de comandas en el restaurante Bonny?

1.4 Objetivos

Objetivo General

Implementar una aplicación web para gestionar registros de comandas para alimentos y bebidas en el Restaurante Bonny utilizando la metodología ágil LEAN.

Objetivos Específicos

- Investigar la metodología ágil LEAN, en la gestión y seguimiento de tareas durante la fase de desarrollo.
- Desarrollar la aplicación web para gestionar registros de comandas para alimentos y bebidas utilizando la metodología ágil LEAN.
- Evaluar el rendimiento de la aplicación web en el modelo de calidad FURPS.

CAPÍTULO II. MARCO TEÓRICO

2.1 Metodología ágil

El concepto de desarrollo ágil surge como un movimiento disruptivo en la industria del software, cuando 17 expertos en desarrollo se reunieron en 2001 para formular el *Manifiesto Ágil*. Este documento reflejaba la necesidad de enfrentar los desafíos del desarrollo tradicional: extensos plazos de entrega, altos costos, y productos finales que no cumplían con las expectativas del cliente [1].

Los modelos de desarrollo como el de cascada, caen dentro de los modelos tradicionales y se caracterizan por tener una secuencia lineal y estricta de fases, donde cada etapa debía concluirse en su totalidad para poder avanzar con la siguiente. Este esquema solo limitaba la posibilidad de hacer cambios en los requerimientos o en las condiciones del entorno. En cambio, la implementación de las metodologías ágiles de desarrollo propone un enfoque más flexible, porque se basan en la iteración constante y la entrega continua, esto permite adaptar el desarrollo a necesidades emergentes y garantizando satisfacción del cliente [2].

Algunos ejemplos de empresas que usan metodologías ágiles son Spotify y Microsoft, que han incorporado estos principios para la optimización de los procesos, reducir tiempos de desarrollo y mejorar la calidad de sus productos [3].

2.2 Principios fundamentales

Estos principios son:

- **Satisfacción del cliente:** Entrega temprana de software en funcionamiento.
- **Aceptación de cambios:** Se pueden hacer cambios en los requisitos, aunque esté en las etapas finales del desarrollo.
- **Entrega frecuente:** Entrega en tiempos cortos.
- **Colaboración constante:** Se valora la cooperación entre desarrolladores y clientes.
- **Enfoque en las personas:** Soporte a todos los que estén dentro del proyecto.
- **Simplicidad:** Tener ciclos de trabajo más productivos, al eliminar acciones innecesarias [1][2].

2.3 Metodologías tradicionales vs. metodologías ágiles

En el desarrollo de software, la comparación entre metodologías tradicionales y metodologías ágiles revela diferencias significativas respecto a la flexibilidad, enfoque y resultados. Mientras que los enfoques tradicionales, como el modelo en cascada, se caracterizan por una estructura rígida y secuencial donde cada fase debe completarse antes de avanzar a la siguiente, las metodologías ágiles destacan por su capacidad de adaptación al cambio y su enfoque iterativo. Esto permite que los equipos entreguen incrementos funcionales del producto en ciclos cortos, con retroalimentación constante del cliente. Por ejemplo, en proyectos dinámicos como un sistema de gestión de comandas para un restaurante, los enfoques tradicionales podrían enfrentar dificultades para ajustarse a cambios inesperados en los flujos de trabajo.

En contraste, las metodologías ágiles, con su énfasis en la entrega continua y la participación activa del cliente, aseguran que el producto final responda mejor a las necesidades reales y en evolución del negocio, minimizando riesgos y optimizando recursos.

2.4 Metodología ágil LEAN

Esta metodología es una fusión entre los principios del desarrollo ágil, centrada en maximizar el valor entregado al cliente a través de la reducción de desperdicios y la mejora constante de los procesos. Tiene un enfoque en la eficiencia operativa y se destaca por su capacidad de adaptación ante cambios en los requerimientos [24].

Koskela identificó dos principios fundamentales de LEAN:

- **Justo a Tiempo (JIT):** Enfocado en producir y entregar los elementos correctos en el momento exacto y en la cantidad precisa, para eliminar desperdicios por sobreproducción y tiempos de espera.
- **Control Total de Calidad (TQC):** Establece que la calidad del producto desarrollado debe ser responsabilidad de todos los miembros proyecto que han de ser parte durante todo el proceso.

El término "LEAN" se emplea para referirse al enfoque de producción ordenada en su conjunto. Diversos autores lo consideran un concepto acertado para representar esta filosofía de forma integral, no se interpreta como un elemento aislado dentro de un marco más amplio. Sin embargo, también existen posturas que difieren respecto a su alcance y definición.

Esta tecnología tiene origen en los principios del sistema de producción de Toyota, pero adaptados al contexto del desarrollo de software. Se conoce por ser considerada una de las metodologías más eficaces dentro del enfoque ágil, gracias a su énfasis en la eliminación de desperdicios, la generación de productos de valor y la mejora continua. Más allá de acelerar los procesos de desarrollo, LEAN busca optimizar el uso de los recursos disponibles y asegurar que cada etapa del proceso contribuya de manera significativa al logro de los objetivos del proyecto.

2.4.1 Origen y fundamentación

El sistema LEAN tiene sus raíces en la industria manufacturera Japonesa, específicamente en el Toyota Production System (TPS) en la década de 1950. Este sistema revolucionó la manera de producir enfocándose en la colaboración y empoderamiento de los equipos.

Durante la década de 1990, Mary y Tom Poppendieck adaptaron los principios de esta filosofía al ámbito del desarrollo de software, al notar similitudes entre la producción de productos físicos y la implementación de soluciones digitales. Así surgió el enfoque conocido como LEAN Software Development (LSD), que incorpora los fundamentos del pensamiento LEAN para enfrentar los desafíos normales del desarrollo ágil en entornos marcados por la complejidad de desarrollo [1][2].

2.4.2 Principios y prácticas

La metodología ágil LEAN se fundamenta en varios principios clave Figura 1:

- **Eliminación de desperdicios:** Identificar y eliminar todas las actividades que no aportan valor al producto final [24].
- **Mejora continua:** Revisiones constantes de procesos para ofrecer mejoras a favor de la calidad y eficiencia [24].
- **Entrega rápida de valor:** Entrega frecuente de funcionalidades al cliente, para recibir retroalimentación a tiempo y ajustarse a las necesidades reales [24].
- **Respeto por las personas:** Aprecia los aportes de cada miembro del equipo, apoyando un ambiente colaborativo y de confianza [24].
- **Optimización del flujo de trabajo:** El objetivo es mantener un flujo de trabajo constante y equilibrado, que evita sobrecargas y cuellos de botella [24].

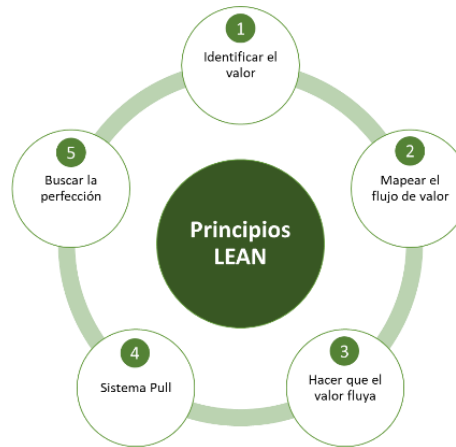


Figura 1: Principios LEAN
Fuente: [24]

Entre las prácticas comunes en ágil LEAN se incluyen, Figura 2:

- **Kanban:** Utilización de tableros con elementos visuales para gestionar el flujo de trabajo.
- **Integración continua:** Agregación de cambios en el código para detectar problemas tempranos.
- **Reuniones diarias:** Reuniones cortas para coordinar actividades y resolver problemas.

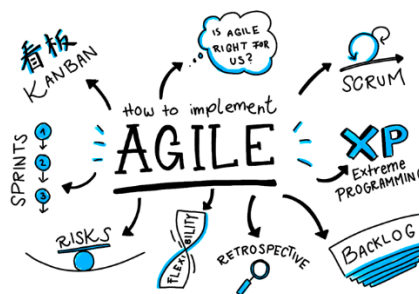


Figura 2: Prácticas LEAN
Fuente: [13]

2.4.3 Fases de la metodología LEAN

1. **Definición del valor:** Esta fase inicial se enfoca en identificar y definir los elementos que representan valor desde la perspectiva del cliente. Para el desarrollo de software, implica una comprensión profunda de los requisitos y expectativas del usuario final, de modo que las características y funcionalidades que se prioricen en el proyecto estén alineadas con sus necesidades. La identificación precisa del valor busca asegurar que el producto final logre satisfacer los objetivos del cliente y maximizar su utilidad.
2. **Mapeo de la cadena de valor:** En esta etapa se lleva a cabo un análisis detallado del proceso de desarrollo, que abarca desde la conceptualización del producto hasta su entrega final. A través del mapeo de la cadena de valor, el equipo puede identificar claramente las actividades que aportan valor y aquellas que no, con el propósito de reducir al mínimo las fuentes de desperdicio. Esta herramienta resulta fundamental para optimizar el flujo de trabajo y construir una base operativa que favorezca la eficiencia y el éxito del proyecto.
3. **Creación de flujo:** Esta fase tiene como propósito asegurar la continuidad del proceso de desarrollo, evitando interrupciones o tiempos de espera innecesarios. Para ello, se diseña un flujo de trabajo ágil en el que cada actividad esté debidamente organizada y optimizada, reduciendo al mínimo las ineficiencias, como los cuellos de botella. En el contexto del desarrollo de software, esto implica establecer una secuencia lógica de tareas que facilite la transición fluida entre etapas y prevenga la acumulación de actividades pendientes en el flujo operativo.
4. **Producción bajo demanda (Pull):** En lugar de anticiparse de forma generalizada a las posibles necesidades del cliente, el enfoque LEAN promueve una producción ajustada a la demanda real, lo que permite evitar tanto la sobreproducción como el desarrollo de características que no aporten valor inmediato. Aplicado al desarrollo de software, este principio implica crear funcionalidades específicas únicamente cuando se presenta una necesidad concreta. De este modo, se garantiza un enfoque verdaderamente centrado en el usuario final y se optimiza el uso de recursos al evitar el diseño de funciones innecesarias.
5. **Perfeccionamiento continuo (Kaizen):** Esta fase busca la mejora continua del proceso de desarrollo mediante la revisión constante y el ajuste de cada una de sus etapas. El perfeccionamiento continuo fomenta un ambiente de retroalimentación y aprendizaje, tanto del equipo de desarrollo como del cliente. En el contexto de software, el Kaizen implica la implementación de mejoras graduales que contribuyan a la eficiencia general del proyecto y aseguren que el producto final sea altamente satisfactorio y relevante para el usuario final. Las fases se ilustran en la Figura 3.



Figura 3: Fases LEAN
Fuente: [13]

2.4.4 Metodología LEAN para el desarrollo de software

El enfoque de LEAN Software Development ofrece un conjunto de herramientas y principios que permiten medir de manera eficiente el progreso y los logros de un proyecto. Esta elección responde a la necesidad de fomentar una colaboración activa dentro del equipo, logrando entregables más rápidos y alineados con los plazos establecidos, en línea con los objetivos planteados.

Para este trabajo de investigación, la selección de la metodología LEAN se direcciona hacia la entrega ágil de productos al cliente, garantizando que los resultados sean positivos y dentro de los tiempos planteados. LEAN a diferencia de otras metodologías, no establece una definición rigurosa de roles dentro del equipo, siendo esto apropiado porque considera las restricciones que pueden existir por el número de desarrolladores que formen parte del proyecto.

El desarrollo de software dentro de este contexto se conforma de cuatro fases principales: análisis, diseño, implementación y pruebas. Cada una de estas etapas cumple con un rol en el desarrollo y optimización del proceso, para asegurar que cada avance sea en pro de contribuir de forma directa al cumplimiento de los objetivos del proyecto, tal como se detalla en la Tabla 1.

Tabla 1. Principios de LEAN en Desarrollo de Software.

Fases de Desarrollo de Software	Principio(s) LSD Aplicado(s)	Aplicación en el Proyecto	Artefactos
Análisis	Potenciar al equipo Entregar lo más rápido posible Eliminar desperdicios Posponer decisiones	Iniciar con una visión completa del proyecto y definir qué se desarrollará y cómo. Definir los objetivos del proyecto y las herramientas a utilizar. Definir los roles del equipo para el desarrollo Reuniones con el cliente para obtener las	Definir los objetivos de proyecto. Definir lenguaje de programación y herramientas de desarrollo de software. Definición de roles de equipo. Historias de Usuario. Cronograma de actividades.

		historias de usuario. Diseñar el cronograma del desarrollo del sistema.	
Diseño	Ampliar aprendizaje Construir con integridad	Modelar los Diagramas de Casos de Uso. Diseño de la Base de Datos. Definir la arquitectura de software y de despliegue. Preparar el prototipo del sistema.	Diagrama de Casos de Uso Modelo Entidad Relación de la Base de Datos. Diagrama de arquitectura de software. Diagrama de despliegue. Diseño de Prototipo
Implementación	Entregar lo más rápido posible Construir con integridad Eliminar desperdicios	Documentar código que sea relevante para el sistema. Documentar la guía de estilo usado en el sistema. Mostar las interfaces de usuario funcionales. Realizar el manual de usuarios del sistema para los interesados.	Documentación de código. Guías de estilo Interfaces de usuario desarrolladas
Pruebas	Entregar lo más rápido posible. Construir con integridad	Definir el plan de pruebas para comprobar el correcto funcionamiento del sistema.	Plan de pruebas LEAN Manual de usuario

Fuente: [2]

2.4.5 Comparación de LEAN con otras metodologías ágiles de desarrollo

LEAN, como metodología ágil, se distingue por su enfoque en la eliminación de desperdicios, la optimización del flujo de trabajo y la entrega de valor continuo al cliente. Sin embargo, existen otras metodologías ágiles, como Scrum, Kanban y XP (Extreme Programming), que tienen diferentes áreas de especialización y aplicación. Comparar estas metodologías permite comprender mejor las fortalezas y debilidades de cada una, así como sus aplicaciones específicas en proyectos de desarrollo de software. Como se observa en la Tabla 2.

Tabla 2: Comparación de LEAN

Aspecto	LEAN	Scrum	Kanban	Extreme Programming (XP)
Enfoque principal	Eliminación de desperdicios, maximización del valor y mejora continua.	Trabajo en sprints con roles definidos (Scrum Master, PO, equipo).	Ver el flujo de trabajo para eliminar cuellos de botella.	Mejora del código a través de técnicas específicas.

Estructura	Basada en principios, no se limita a roles específicos.	Estructurada con eventos clave (sprints, daily, review, retro).	No tiene roles ni eventos definidos, se maneja a través de WIP.	Ayuda a la planificación para programar en parejas.
Iteración	Tiene un flujo continuo dependiendo del proyecto.	Los sprints van desde 1 a 4 semanas.	No tiene iteraciones definidas de forma estricta.	Las iteraciones son cortas y con enfoque en la calidad.
Medición del progreso	Centrado en las métricas de evaluación procurando eliminar desperdicio.	Se basa en cumplir con los objetivos de cada sprint.	Se basa en seguir el flujo de trabajo.	Valora la satisfacción del cliente.
Adaptabilidad	Es muy adaptable porque se acopla a entornos cambiantes.	Es adaptable, pero se mantiene muy estructurado.	Su adaptabilidad es buena para acciones repetitivas.	No es tan flexible porque se enfoca en pasos específicos.
Rol del cliente	Forma parte de todo el proceso,	El rol de Product Owner representa al cliente.	Los clientes no se involucran en todo el proceso necesariamente.	El cliente es clave para definir prioridades de desarrollo.

Fuente: [2]

2.5 Frameworks y lenguajes para desarrollo web

El uso de frameworks y lenguajes de programación ha evolucionado el desarrollo de aplicaciones porque están diseñados para que el proceso de creación, mantenimiento y escalabilidad de sistemas complejos sea más ágil. Estos frameworks, aparte de proporcionar una estructura más estandarizada, incluyen herramientas y bibliotecas para facilitar el desarrollo de aplicaciones seguras y estables [1]. En el contexto de este proyecto de investigación, el uso de tecnologías adecuadas es el primer paso para construir aplicaciones que ofrezcan una experiencia de usuario rápida y confiable. En estos momentos, Python, PHP y Ruby on Rails se encuentran entre las opciones más escogidas para desarrollo de aplicaciones de comandas [2].

2.5.1 Django

Es un framework basado en Python, ha ganado popularidad gracias a su arquitectura robusta y a su principio de "no te repitas" (DRY, Don't Repeat Yourself), facilitando el desarrollo de aplicaciones eficientes [4]. Django emplea el patrón de arquitectura Model-View-Template (MVT), que permite a los desarrolladores separar claramente la lógica de negocio de la interfaz de usuario, obteniendo en un código más ordenado y mantenible [5]. Como se observa en la Figura 4.

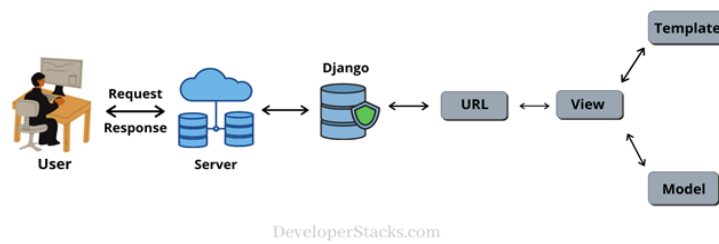


Figura 4: Framework Django
Fuente: [3]

Además de su arquitectura en base a módulos, Django autentificación de usuarios, administración de contenido y protección contra amenazas, tales como inyecciones SQL y ataques XSS (Cross-Site Scripting) [6]. La integración de Django gestores de bases de datos como MySQL permite una gestión eficiente de datos, esto es adecuado para aplicaciones con datos que deben mantenerse seguros en las integraciones [7]. Gracias a estas características, Django se propone como una alternativa sólida para el desarrollo de sistemas de gestión en restaurantes, al ofrecer un equilibrio óptimo entre rendimiento, escalabilidad y flexibilidad [8].

2.5.2 Comparativa con otras tecnologías (PHP, Ruby on Rails)

Para el desarrollo de aplicaciones web, también existen otras alternativas como PHP y Ruby on Rails, cada una con sus propias ventajas. PHP, muy utilizado en el desarrollo web, destaca por su integración sencilla con servidores y bases de datos relacionales, lo que ha impulsado su uso en plataformas populares de desarrollo web como WordPress y Joomla [9]. Si bien es eficaz para proyectos de pequeña y mediana escala, es necesario el mayor consumo de recursos si se desea alcanzar rendimiento y seguridad comparables a los que Django ofrece [10].

Por otro lado, Ruby on Rails, un framework basado en el lenguaje Ruby, es popular en el desarrollo de aplicaciones web debido a su enfoque en la simplicidad y la rapidez de desarrollo [11]. Rails, al igual que Django, emplea una arquitectura similar al MVT y sigue el principio de "convención sobre configuración", para reducir la necesidad de configuración adicional y permite un desarrollo más rápido [12]. Sin embargo, Django suele ofrecer un rendimiento superior en aplicaciones de gran escala, y Python se considera un lenguaje más versátil y rápido en comparación con Ruby [13]. La comparativa se observa en la Figura 5.

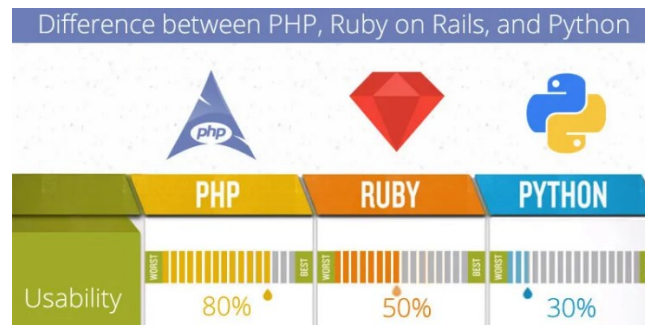


Figura 5: Comparativos de Frameworks
Fuente: [13]

2.6 Bases de datos y su relevancia en la gestión de datos de negocios

En el contexto de la gestión empresarial moderna, las bases de datos desempeñan un papel esencial al facilitar el almacenamiento, la organización y el análisis eficiente de la información. Su implementación permite a las organizaciones acceder con rapidez a datos clave y utilizarlos como soporte para la toma de decisiones estratégicas, lo cual es especialmente relevante en sectores de alta rotación como el de la restauración. Una base de datos bien diseñada no solo garantiza la consistencia y precisión de la información, sino que también ofrece mecanismos para protegerla y respaldarla adecuadamente [1].

2.6.1 Bases de datos relacionales: MySQL

Son un tipo de sistema de gestión de bases de datos (DBMS) donde los datos se organizan en tablas que se relacionan a través de claves primarias y foráneas, para hacer consultas de los datos. MySQL, es uno de lo más utilizados por su alto rendimiento y habilidad para manejar grandes volúmenes de datos y por ser compatibles con varios lenguajes de programación [2].

MySQL ofrece grandes beneficios en aplicaciones empresariales, entre estos el soporte para transacciones y el ajuste con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), lo que garantiza la seguridad de los datos, incluso durante operaciones complejas [3]. Esta característica convierte a MySQL en una opción confiable para la gestión de comandas, donde es fundamental asegurar tanto la integridad como la disponibilidad continua de los datos [4].

2.6.2 Escalabilidad y rendimiento de bases de datos

La escalabilidad y el rendimiento de una base de datos son factores importantes para aplicaciones empresariales [5]. La escalabilidad permite que los negocios, como los restaurantes, gestionen un volumen creciente de datos sin comprometer el rendimiento del sistema.

Para mejorar el rendimiento, MySQL tiene varias opciones de configuración, como la creación de índices, particionar tablas y poder escoger motores de almacenamiento. Entre estos, InnoDB, que es compatible con transacciones y su capacidad de bloqueo a nivel de registro. Además, MySQL tiene la función de replicar para permitir la sincronización de

datos entre múltiples servidores, garantizando alta disponibilidad y recuperación ante fallos. Estas características resaltan valor en sistemas de gestión que trabajan a tiempo real [6].

El rendimiento en MySQL también se puede mejorar mediante el uso de técnicas de almacenamiento en caché y optimización de consultas, así se reduce el tiempo de respuesta de las consultas y mejora la experiencia del usuario. Estas características hacen que MySQL sea una base de datos adecuada para aplicaciones que requieren alta velocidad de procesamiento y confiabilidad en la manipulación de datos [7].

2.7 Calidad del software

Investigadores españoles y colombianos [22] subrayan que el software debe poseer ciertas características que aseguren a los usuarios un producto seguro y confiable en términos de funcionalidad y eficiencia. Estas características mejoran el desempeño del software a lo largo de su ciclo de vida, contribuyendo a la excelencia del software. Debe ser garantizada mediante un modelo de calidad que se aplique durante todo el proceso de desarrollo, gestionando las características del software y asegurando que coincida con los requisitos iniciales [22].

El IEEE define una métrica de calidad de software como una función que toma datos del software y produce un valor numérico que representa el grado en que el software posee un atributo específico que afecta su calidad [23]. Esta definición se puede aplicar tanto a productos como a servicios de software e infraestructura. El uso adecuado de estas métricas puede reducir el número de fallos en el software.

Las métricas de calidad del software se agrupan en tres categorías principales: métricas de proyectos, métricas de procesos y métricas de productos. Las métricas de productos son las más comunes en la evaluación del software, porque se centran en el desarrollo del producto y se utilizan en la fase final del proceso. Según algunos autores [24], [23], las métricas deben ser validadas teóricamente para asegurar que miden correctamente los atributos, validadas empíricamente para evaluar su utilidad en la medición de atributos externos, y deben ser fáciles de realizar, preferiblemente de forma automatizada.

Para medir la calidad del software, el sistema de medición debe seleccionarse considerando estándares que aseguren resultados verificables, reales y no subjetivos, con precisión y significancia [4]. Los modelos de calidad de software siguen una estructura [25], [24], y contienen diversos atributos de calidad evaluables mediante métricas, facilitando una evaluación objetiva al asignar valores cualitativos o cuantitativos a los atributos analizados.

Las métricas de calidad en productos de software comprenden diversas dimensiones que permiten evaluar su desempeño y robustez. Entre ellas se encuentran el tamaño del software, medido a través de las métricas de Halstead [25], y la complejidad del programa, determinada por la complejidad ciclomática [25]. También se consideran aspectos como el rendimiento, la accesibilidad y la seguridad del sistema. Por otro lado, el proceso de pruebas

se evalúa mediante indicadores como el esfuerzo destinado a su ejecución y la eficiencia obtenida durante su desarrollo.

2.7.1 Modelo FURPS

Propuesto por Robert Grady [26], fue extendido a FURPS+ por IBM Rational Software [26][27], abarcando requisitos como necesidades físicas, requisitos de interfaz, requisitos de implementación y restricciones de diseño. El modelo clasifica las características en requisitos funcionales y no funcionales. Los requisitos funcionales se definen por las entradas y salidas esperadas o la funcionalidad (F). Por otro lado, los requisitos no funcionales se agrupan en usabilidad (U), fiabilidad (R), rendimiento (P) y soporte del producto (S) [26].

Para evaluar el rendimiento de la aplicación, se establecieron métricas específicas basadas en el modelo FURPS. Este modelo permite evaluar de forma integral la aplicación, porque considera a los atributos funcionales y no funcionales [27]. En la Tabla 3 se detallan los parámetros considerados y los valores obtenidos. [27].

Tabla 3: Métricas de Rendimiento del Modelo FURPS	
Parámetros	Valores
Tiempo de respuesta	5s
Eficacia	95%
Uso de recursos	25%

Fuente: [27]

2.8 JMeter

Es una herramienta de código abierto desarrollada por Apache, diseñada principalmente para realizar pruebas de rendimiento y carga en aplicaciones web. Su objetivo principal es simular múltiples usuarios concurrentes accediendo a una aplicación, para evaluar el comportamiento del sistema bajo condiciones de carga [28]. JMeter puede generar una gran cantidad de tráfico de usuarios y medir varios aspectos del rendimiento, como tiempos de respuesta, ancho de banda, uso de recursos del servidor y capacidad de procesamiento. Utilizando JMeter, es posible identificar cuellos de botella y problemas de rendimiento en la aplicación antes de que esta se despliegue en un entorno de producción [28].

Para evaluar usando JMeter, hay que configurar los "Thread Groups", que permiten simular el uso de múltiples usuarios a la vez, y los "Samplers", que representan las solicitudes HTTP [28]. JMeter permite personalizar estos elementos para reproducir escenarios. Los resultados obtenidos se observan en los "Listeners", que presentan la información en tablas y gráficos detallados. Por estas funcionalidades se escogió JMeter [29].

CAPÍTULO III. METODOLOGÍA

El enfoque de la investigación fue cuantitativo, centrado en evaluar el rendimiento de la aplicación web desarrollada para la gestión de comandas en el restaurante Bonny.

3.1 Tipo de investigación

3.1.1 Según la fuente de la investigación

Investigación bibliográfica: La base teórica de este proyecto se fundamentó en una revisión bibliográfica de artículos sobre la metodología LEAN en el desarrollo de una aplicación web para la gestión de comandas del restaurante Bonny.

3.1.2 Según el objeto de estudio

Investigación aplicada: Tiene un enfoque aplicado, porque el objetivo principal fue el desarrollo de una aplicación web para la gestión de registros de comandas utilizando la metodología ágil LEAN.

Investigación descriptiva: La investigación adoptó un enfoque descriptivo, orientado a analizar y caracterizar los resultados obtenidos durante la evaluación del rendimiento del sistema, empleando para ello las métricas establecidas por el modelo FURPS. Este enfoque permitió documentar de manera detallada las propiedades funcionales y no funcionales del software desarrollado.

3.2 Diseño de la investigación

No experimental, con enfoque cuantitativo. Se eligió un diseño de investigación no experimental con un enfoque cuantitativo, porque no se intervinieron ni manipularon las variables independientes.

3.3 Población de estudio y tamaño de la muestra

La evaluación del rendimiento de la aplicación web se realizó con la herramienta JMeter, considerando una población infinita.

3.4 Técnicas de recolección de datos

El rendimiento de la aplicación web se evaluó utilizando la herramienta JMeter, obteniendo porcentajes de eficiencia, tiempos de respuesta y consumo de recursos.

Eficiencia: refleja la cantidad de solicitudes que el sistema procesa exitosamente en relación con el total de peticiones realizadas.

Tiempo de respuesta: es el intervalo que transcurre desde que un usuario envía una solicitud al servidor hasta que se produzca la primera respuesta. Se expresa generalmente en milisegundos (ms).

Consumo de recursos: es el uso de CPU y memoria mientras se ejecutan las pruebas de carga.

3.5 Identificación de variables

3.5.1 Variable independiente

Aplicación web

3.5.2 Variable dependiente

Rendimiento de la aplicación web para la gestión de comandas del restaurante Bonny.

3.6 Operacionalización de variables

A continuación, en la Tabla 4, se describe la operacionalización de variables.

Tabla 4: Operacionalización de variables

PROBLEMA	TEMA	OBJETIVOS	VARIABLES	CONCEPTUALIZACIÓN	DIMENSIÓN	INDICADORES
¿La implementación de la metodología ágil LEAN incidirá en el rendimiento de la aplicación web para la gestión de comandas en el restaurante Bonny?	Aplicación web para gestionar registros de comandas para alimentos y bebidas en el Restaurante Bonny utilizando la metodología ágil LEAN.	General	Independiente	Herramientas utilizadas por usuarios para acceder a un servidor web mediante internet y en dispositivos móviles.	Software	- Tiempo de Desarrollo - Número de Líneas de Código - Número de módulos.
		Específicos	Dependiente	Se determina al conjunto de aspectos que deben ser considerados para el desarrollo de la gestión de comandas del restaurante Bonny dentro de la aplicación web.	Rendimiento	- % Eficacia. - % Tiempo de respuesta. - % Utilización de recursos.
		- Investigar la metodología ágil LEAN, en la gestión y seguimiento de tareas durante la fase de desarrollo. - Desarrollar la aplicación web para gestionar registros de comandas para alimentos y bebidas utilizando la metodología ágil LEAN. - Evaluar el rendimiento de la aplicación web en el modelo de calidad furps.	Rendimiento de la aplicación web para la gestión de comandas del restaurante Bonny			

3.7 Metodología de desarrollo

Para desarrollar la aplicación web de gestión de comandas en el restaurante Bonny, se implementó la metodología LEAN para tener un proceso ágil y eficiente en todas las etapas del proyecto. El que se estructuró en cuatro fases: análisis, diseño, implementación y pruebas, cada etapa alineada con el cumplimiento de los objetivos, optimizando recursos y teniendo como prioridad las necesidades del cliente. Estas fases se realizaron con los principios de LEAN, que fueron detallados en actividades específicas para el desarrollo: identificación de valor, mapeo del flujo de valor, creación de flujo continuo, establecimiento de un sistema pull, mejora continua (Kaizen).

Para el desarrollo del sistema, se utilizó el framework Django, que permitió una integración adecuada a los principios de la metodología LEAN, y así se obtuvo una solución que no solo optimizó la gestión de comandas, sino que también mejoró la experiencia del usuario, redujo errores y aumentó la eficiencia operativa del restaurante.

3.7.1 Definición del Valor

Situación actual y problemas identificados

La metodología LEAN en la etapa de análisis, el primer paso es identificar los requerimientos desde el punto de vista del cliente, también definir el flujo de valor de la operación actual. En este proceso hay que identificar actividades innecesarias, demoras y errores que perjudican el proceso. Al mapear y analizar el flujo operativo, se pueden identificar puntos críticos, con la finalidad de establecer estrategias para eliminar desperdicios, optimizar las tareas y mejorar la calidad del servicio.

Requisitos funcionales

Los requisitos funcionales de detallan en la Tabla 5.

Tabla 5: Requisitos funcionales de la aplicación web

ID	Prioridad	Funcionalidad	Descripción	Observaciones
RF01	Alta	Registro de comandas	Permitir a los meseros registrar nuevas comandas en menos de 30 segundos.	La interfaz debe ser intuitiva para minimizar el tiempo de registro.
RF02	Alta	Notificaciones automáticas	Notificar a la cocina cada vez que se registre una nueva comanda.	La notificación debe ser en tiempo real y confiable.
RF03	Media	Modificación de comandas	Permitir a los meseros editar una comanda antes de que sea procesada por la cocina.	El sistema debe registrar los cambios realizados.
RF04	Alta	Visualización del estado de comandas	Mostrar en tiempo real el estado de cada comanda (pendiente, en preparación, listo).	La actualización debe ser automática y sin necesidad de recargar la página.

RF05	Alta	Actualización de estado de pedidos por chefs	Permitir que los chefs actualicen el estado de las comandas.	Debe ser un proceso sencillo para evitar demoras.
RF06	Alta	Generación de reportes	Los administradores generan reportes de ventas en formato PDF y Excel.	Reportes de número de comandas e ingresos.
RF07	Media	Gestión de usuarios	Los administradores pueden agregar, editar y eliminar usuarios con diferentes roles.	Los roles deben incluir mesero, chef y administrador.
RF08	Alta	Registro de métricas	Tener métricas de desempeño como tiempos de atención promedio y comandas despachadas.	Las métricas se deben visualizar en un tablero de administración.
RF09	Baja	Consulta del menú digital	Los usuarios pueden consultar el menú desde cualquier dispositivo.	Esta funcionalidad es opcional para clientes o usuarios finales.
RF10	Media	Notificaciones para clientes	Enviar estado de los pedidos a los clientes.	Se puede desarrollar en fases posteriores del proyecto.

Requisitos no funcionales

Los requisitos no funcionales de detallan en la Tabla 6.

Tabla 6: Requisitos no funcionales de la aplicación web

ID	Prioridad	Funcionalidad	Descripción	Observaciones
RNF01	Alta	Disponibilidad	El sistema debe estar disponible al menos el 99.5% del tiempo.	Se deben implementar medidas de alta disponibilidad en el servidor.
RNF02	Alta	Tiempo de respuesta	El tiempo de respuesta para registrar una comanda debe ser inferior a 2 segundos.	Monitorear rendimiento.
RNF03	Alta	Compatibilidad con navegadores	Aplicación web compatible con todos los navegadores.	Probar en varios navegadores.
RNF04	Alta	Accesibilidad multiplataforma	Debe poder funcionar en dispositivos móviles y computadores.	Interfaces adaptables, es decir responsivas.
RNF05	Alta	Seguridad	Proteger los datos con encriptación.	Implementar protocolos HTTPS.
RNF06	Media	Usabilidad	Interfaz de usuario intuitiva y de fácil uso	Realizar pruebas con usuarios
RNF07	Alta	Escalabilidad	El sistema debe poder se manejado aun cuando hay muchos usuarios.	Arquitectura que permita usarse cuando hayan muchos requerimientos.
RNF08	Media	Integración futura	El sistema debe permitir integración con sistemas de inventario y facturación.	Planificar una API que facilite la integración.

RNF09	Media	Descarga rápida de reportes	Los reportes generados deben ser descargables en menos de 5 segundos.	Optimizar consultas a la base de datos para reportes complejos.
RNF10	Alta	Auditoría de actividades	El sistema debe mantener un registro de actividad para auditoría y control.	Los registros deben incluir fecha, hora y usuario de cada operación.

Roles de la aplicación

Los roles de la aplicación web, se detallan en la Tabla 7.

Tabla 7: Roles de la aplicación

Rol	Descripción	Responsabilidades
Administrador	Persona encargada de la gestión y configuración del sistema.	Gestionar usuarios: agregar, editar y eliminar cuentas del sistema. Monitorear el estado y desempeño de la aplicación. Configurar accesos y permisos según roles del personal.
Usuario (Mesero)	Persona encargada de registrar y gestionar las comandas de los clientes en el sistema.	Registrar nuevas comandas en el sistema de forma rápida y eficiente. Consultar el estado de las comandas para informar a los clientes. Modificar comandas si el cliente cambia su pedido antes de que sea procesado.
Usuario (Chef)	Persona encargada de recibir y procesar las comandas en la cocina.	Visualizar las comandas en tiempo real desde la pantalla de la cocina. Actualizar el estado de las comandas (pendiente, en preparación, listo). Priorizar comandas según el tiempo transcurrido desde su registro.
Usuario (Cliente)	(Opcional) Persona que utiliza el sistema para consultar el menú o el estado de su pedido (si aplica).	Consultar el menú digital desde un dispositivo personal. Recibir notificaciones del estado de su pedido hasta su finalización.

3.7.2 Mapeo de la Cadena de Valor

Mapa de procesos actual

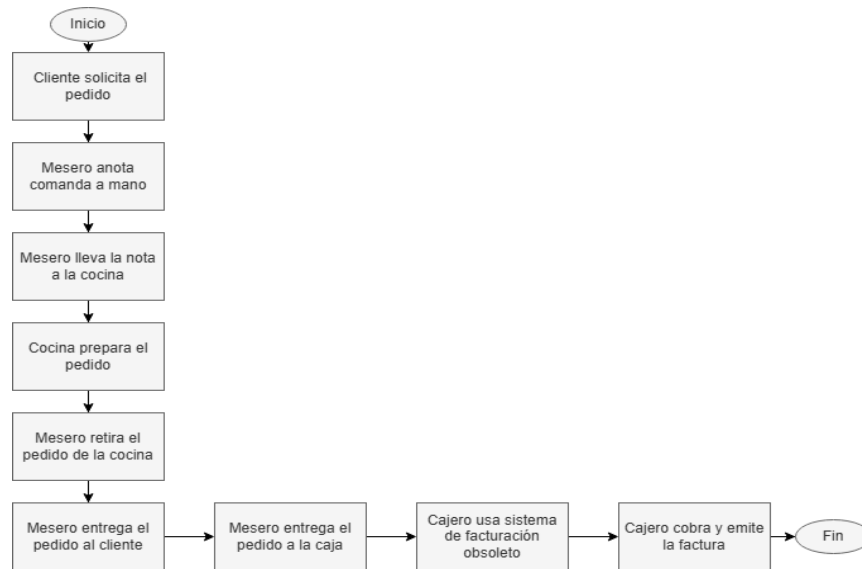


Figura 6: Flujo actual de comandas restaurante Bonny

Propuesta de flujo optimizado

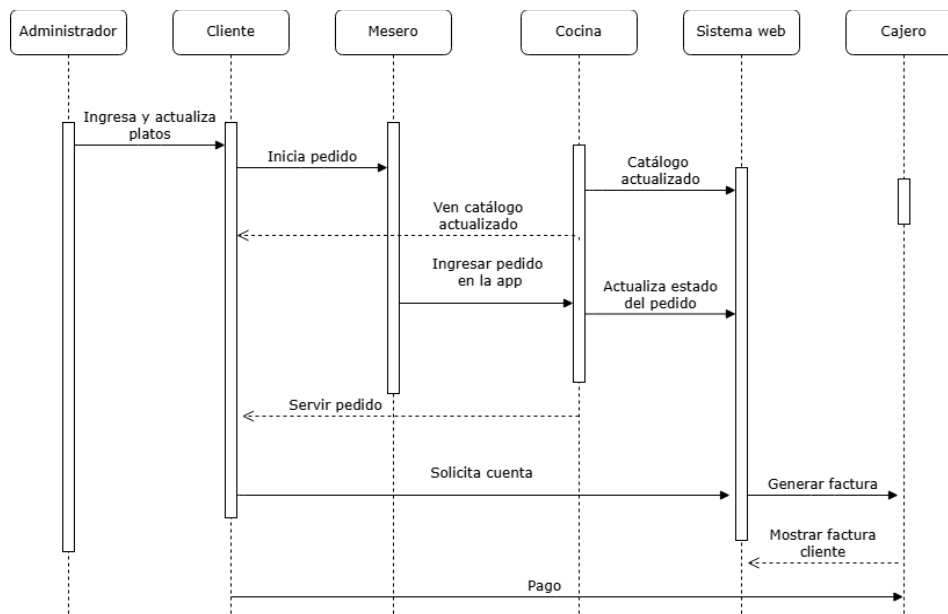


Figura 7: Flujo optimizado

Diagrama de componentes

El diagrama de componentes Figura 8, muestra una arquitectura en tres capas: Frontend, servidor de aplicación y base de datos. El Frontend se presenta como un componente que incluye módulos dedicados a la gestión del menú, las comandas y la facturación, y se conecta con el servidor de aplicación a través de interfaces. El servidor de aplicación, a su vez, agrupa

subcomponentes especializados en la lógica de las comandas, la gestión del inventario de menú, el seguimiento del estado de las órdenes y la generación de facturas. Finalmente, la base de datos actúa como un repositorio central que almacena la información del menú, las comandas, la facturación y los roles de los usuarios.

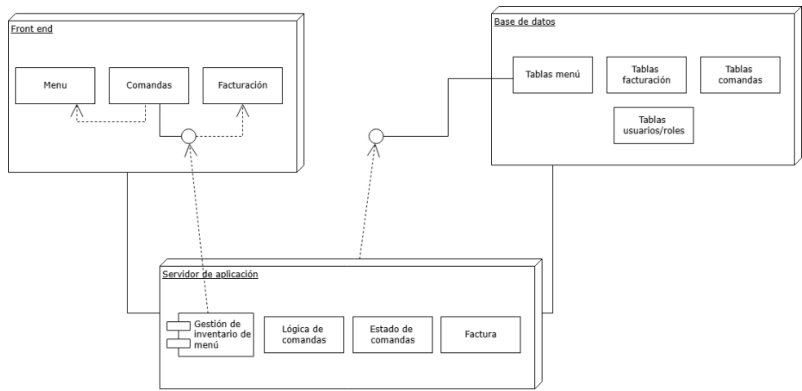


Figura 8: Diagrama de componentes

3.7.3 Creación de Flujo

En la etapa de diseño, bajo el enfoque LEAN, se busca crear un flujo continuo y eliminar todo aquel que no genera valor, priorizando la agilidad y la simplicidad del sistema. Para ello, se desarrollarán diagramas de caso de uso que reflejen la interacción entre los distintos actores y las funcionalidades críticas, se implementará un diseño modular que permita incorporar mejoras sin afectar el resto del sistema, y se elaborará un esquema de base de datos optimizado que facilite la búsqueda y actualización de la información en tiempo real. Estas acciones aseguran que cada decisión tomada durante el diseño sea en pro al desarrollo de la solución.

Planificación de actividades

Para apoyar el desarrollo se tomó en cuenta el ciclo: Planificar-Hacer-Verificar-Actuar que fue clave, pues se brindó una estructura metodológica para la mejora continua. Como se detalla en la Tabla 8.

Tabla 8: Planificación de actividades

	Actividad	Realizado
Plan	Definir objetivos.	✓
	Requerimientos funcionales y no funcionales.	✓
	Mapeo del proceso actual y análisis de problemas.	✓
	Mapeo de la solución.	✓
	Levantar casos de uso.	✓
	Diseño de base de datos.	✓
	Definir diccionario de base de datos.	✓

	Definir herramientas a usar.	✓
Do (Hacer)	Desarrollo de las funcionalidades siguiendo los casos de uso. Implementación del modelo de datos descrito en el diccionario. Desarrollo de las pantallas y módulos planteados en el diseño.	
Check (Verificación)	Analizar los resultados obtenidos Comparar tiempos iniciales Medir rendimiento	
Act (Acción)	Ajustes necesarios Corrección de posibles errores	

Diagramas de caso de uso

Actores:

- Mesero: Usuario que toma la orden del cliente y la ingresa al sistema.
- Cocina: Personal encargado de preparar las comandas.
- Cajero: Usuario que genera la factura y gestiona el pago.
- Cliente: Consumidor final.

Al modelar de manera clara y concisa los casos de uso, se facilita la identificación de oportunidades para aplicar los principios LEAN, orientando el desarrollo hacia la eliminación de desperdicios, la optimización de los procesos y la entrega de valor al cliente final. Los casos de uso se detallan en la Figura 9.

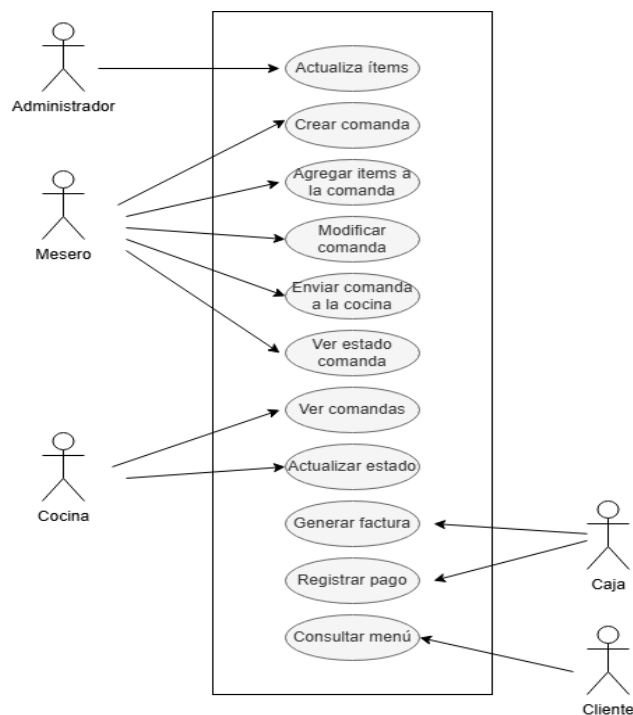


Figura 9: Casos de uso

Diagrama de clases

El diagrama de clases del sistema se representa en la Figura 10.

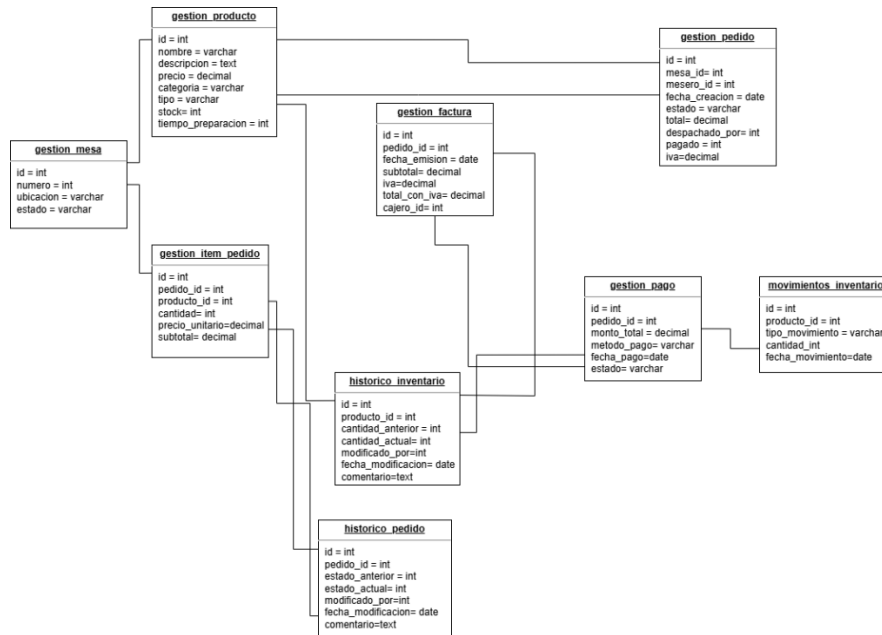


Figura 10: Diagrama de clase

Esquema de base de datos

La Figura 11, muestra el diagrama de la base de datos que sustenta el sistema desarrollado en esta tesis. Este diagrama es una referencia para entender la estructura interna del sistema.

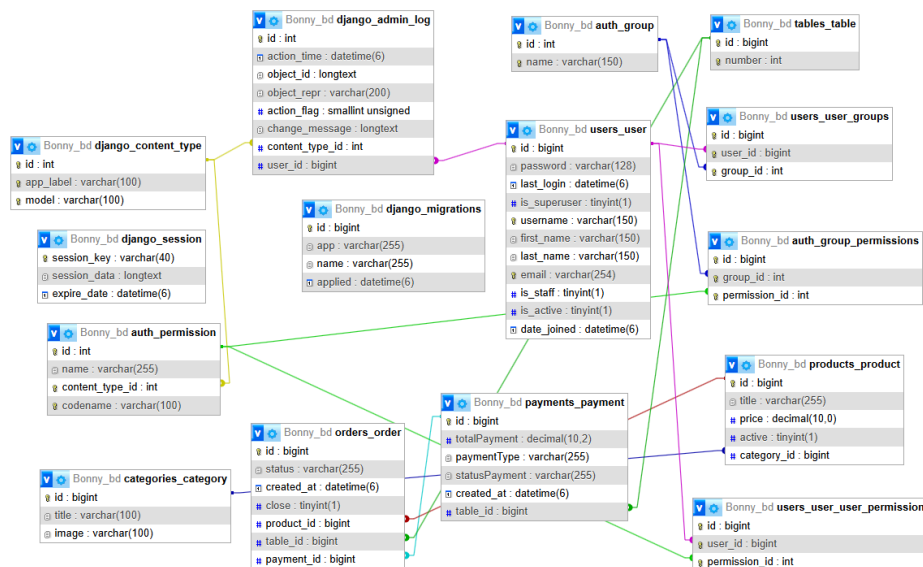


Figura 11: Diagrama de base de datos

Diccionario de datos

Tabla: auth_user

Descripción: Almacena la información de los usuarios del sistema. Tabla 9.

Tabla 9: Tabla auth_user			
Campo	Tipo	Descripción	Restricciones/Relaciones
id	bigint	Identificador único del usuario	PK, autoincremental
username	varchar(150)	Nombre de usuario para login	Único, no nulo
email	varchar(254)	Correo electrónico del usuario	Puede ser nulo o único según políticas
password	varchar(128)	Hash de la contraseña del usuario	No nulo
first_name	varchar(150)	Nombre del usuario	Opcional
last_name	varchar(150)	Apellido del usuario	Opcional
role	int	Rol asignado al usuario (referencia a auth_role)	FK a auth_role(id)
is_staff	tinyint	Indica si es staff (1=Sí, 0=No)	No nulo, booLEANo
is_superuser	tinyint	Indica si es superusuario (1=Sí, 0=No)	No nulo, booLEANo
is_active	tinyint	Indica si la cuenta está activa (1=Sí, 0=No)	No nulo, booLEANo
created_at	datetime	Fecha y hora de creación del registro	No nulo, valor por defecto current timestamp
updated_at	datetime	Fecha y hora de última actualización	Se actualiza en cada modificación

Tabla: auth_role

Descripción: Almacena los roles posibles para los usuarios. Tabla 10.

Tabla 10: Tabla auth_role			
Campo	Tipo	Descripción	Restricciones/Relaciones
id	int	Identificador único del rol	PK
name	varchar(50)	Nombre del rol	Único, no nulo
description	text	Descripción del rol	Opcional

Tabla: auth_user_groups

Descripción: Asociación entre usuarios y grupos (si se manejan grupos). Tabla 11.

Tabla 11: Tabla auth_user_groups			
Campo	Tipo	Descripción	Restricciones/Relaciones
id	bigint	Identificador único	PK
user_id	bigint	Identificador del usuario	FK a auth_user(id), no nulo
group_id	bigint	Identificador del grupo	Referencia a un grupo

Tabla: gestion_mesa

Descripción: Define las mesas del restaurante. Tabla 12.

Tabla 12: Tabla gestion_mesa

Campo	Tipo	Descripción	Restricciones/Relaciones
id	bigint	Identificador único de la mesa	PK
numero	int	Número asignado a la mesa	Único, no nulo
ubicacion	varchar(100)	Ubicación física de la mesa en el restaurante	Opcional
estado	varchar(50)	Estado de la mesa (ej. "disponible", "ocupada")	No nulo
created_at	datetime	Fecha de creación	No nulo
updated_at	datetime	Fecha de última actualización	-

Tabla: gestion_producto

Descripción: Almacena el catálogo de productos (comidas, bebidas). Tabla 13.

Tabla 13: Tabla gestion_producto

Campo	Tipo	Descripción	Restricciones/Relaciones
id	bigint	Identificador único del producto	PK
nombre	varchar(100)	Nombre del producto	No nulo, único
descripcion	text	Descripción del producto	Opcional
precio	decimal(10,2)	Precio unitario del producto	No nulo
categoria	varchar(50)	Categoría del producto (ej. "bebida", "entrada")	No nulo
iva	decimal(5,2)	Impuesto aplicado (si aplica)	No nulo (0 si no aplica)
stock	int	Cantidad disponible en inventario	No nulo
tiempo_preparacion	int	Tiempo estimado de preparación (en minutos)	Opcional
created_at	datetime	Fecha de creación del registro	No nulo
updated_at	datetime	Fecha de última actualización	-

Tabla: gestion_pedido

Descripción: Representa una orden realizada por un cliente. Tabla 14.

Tabla 14: Tabla gestion_pedido

Campo	Tipo	Descripción	Restricciones/Relaciones
id	bigint	Identificador único del pedido	PK
mesa_id	bigint	Mesa asociada al pedido	FK a gestion_mesa(id), no nulo
mesero_id	bigint	Mesero que atendió el pedido	FK a auth_user(id)
fecha_creacion	datetime	Fecha y hora en que se inició el pedido	No nulo
estado	varchar(20)	Estado del pedido (ej. "pendiente", "en preparación", "listo", "entregado")	No nulo
total	decimal(10,2)	Monto total del pedido (sin IVA)	No nulo
despachado_por	bigint	Usuario que despachó el pedido (opcional)	FK a auth_user(id) opcional
pago	tinyint	Indica si fue pagado (1=Sí, 0=No)	No nulo, booleano
iva	decimal(5,2)	IVA aplicado al pedido	No nulo (0 si no aplica)

Tabla: gestion_itempedido

Descripción: Detalle de los productos solicitados en un pedido. Tabla 15.

Tabla 15: Tabla gestion_itempedido

Campo	Tipo	Descripción	Restricciones/Relaciones
pedido_id	bigint	Identificador del pedido	FK a gestion_pedido(id), no nulo
producto_id	bigint	Identificador del producto	FK a gestion_producto(id), no nulo
cantidad	int	Cantidad solicitada de ese producto	No nulo
precio_unitario	decimal(10,2)	Precio unitario del producto en el momento del pedido	No nulo
subtotal	decimal(10,2)	Subtotal (cantidad * precio_unitario)	No nulo

3.7.4 Producción bajo demanda (Pull)

Modelado de interfaces

En la Figura 12, se muestra la interfaz del sistema web para comandas del restaurante Bonny.

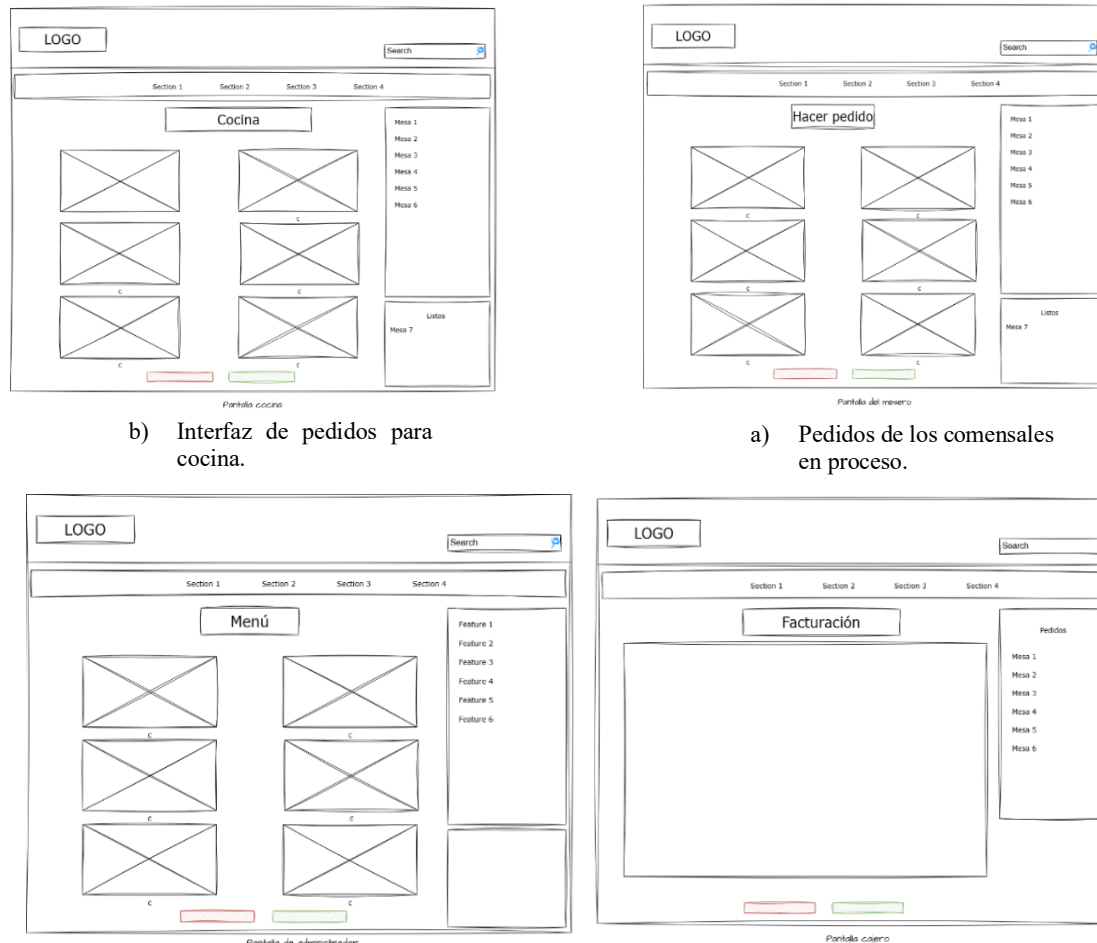


Figura 12: Modelado de interfaz aplicación web

En la Figura 13, se muestra la interfaz de la aplicación móvil del restaurante Bonny.

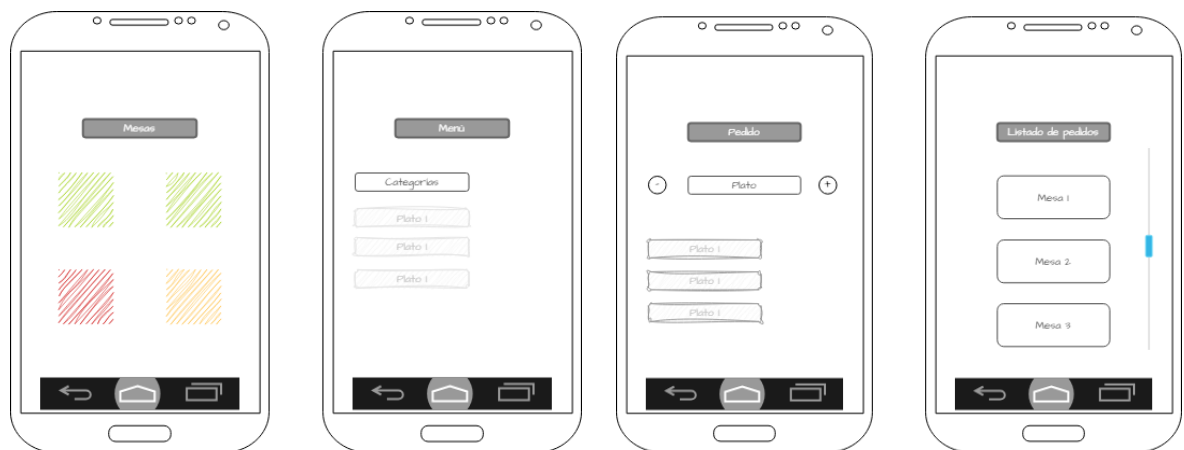


Figura 13: Modelado de interfaz aplicación móvil

Arquitectura de solución

En la aplicación web desarrollada con Django, se empleó la arquitectura basada en el patrón Modelo-Vista-Plantilla (MVT), característica de este framework, figura 14. Este enfoque permite generar dinámicamente las páginas HTML desde el servidor, integrando lógica de negocio y datos en la vista, para luego presentarlos mediante plantillas que simplifican la composición de la interfaz. Por el contrario, en la aplicación móvil se optó por una arquitectura cliente-servidor más tradicional, en la que el dispositivo solicita información a un servidor, figura 15.

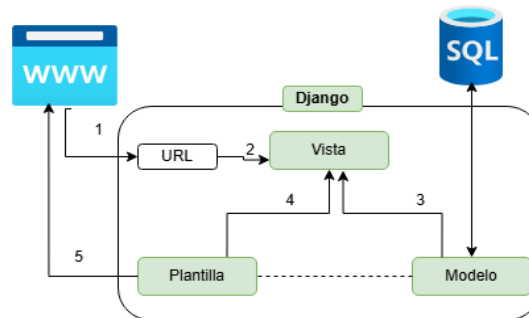


Figura 14: Arquitectura aplicación web

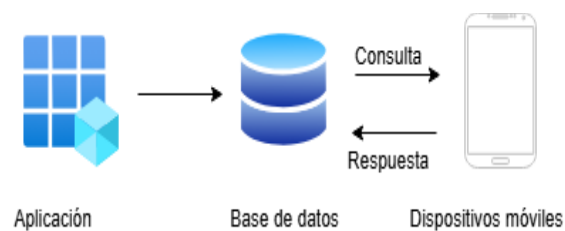
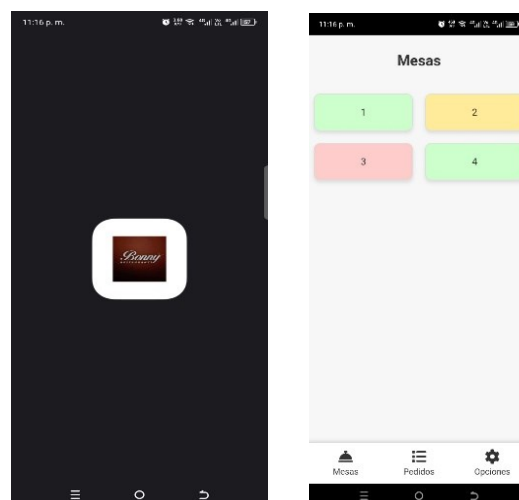


Figura 15: Arquitectura aplicación móvil para meseros

Interfaz gráfica

La Figura 16, se presenta una visión general de las interfaces gráficas implementadas para la aplicación móvil. Estas pantallas han sido concebidas con el objetivo de ofrecer una experiencia de usuario intuitiva, coherente y visualmente atractiva.



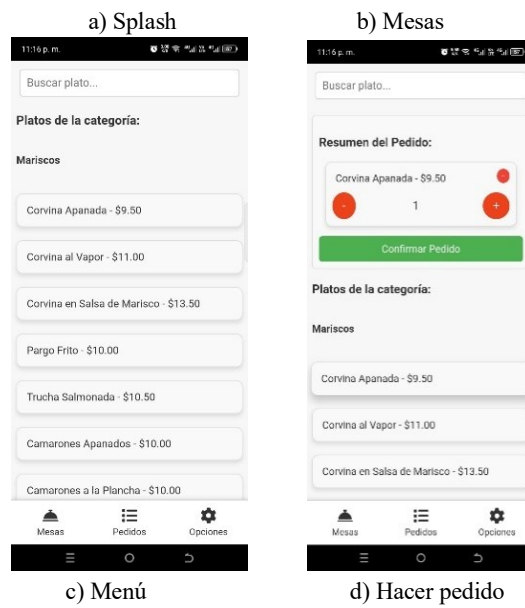


Figura 16: Interfaz gráfica de la aplicación móvil

La Figura 17, muestra la interfaz principal del módulo Gestión de Caja de un sistema desarrollado para la administración de pedidos en un restaurante. La pantalla está compuesta por una barra lateral de navegación ubicada al lado izquierdo, que incluye opciones como Pedidos en Curso e Historial de Pedidos. En la parte superior, se presenta un encabezado con el título principal.

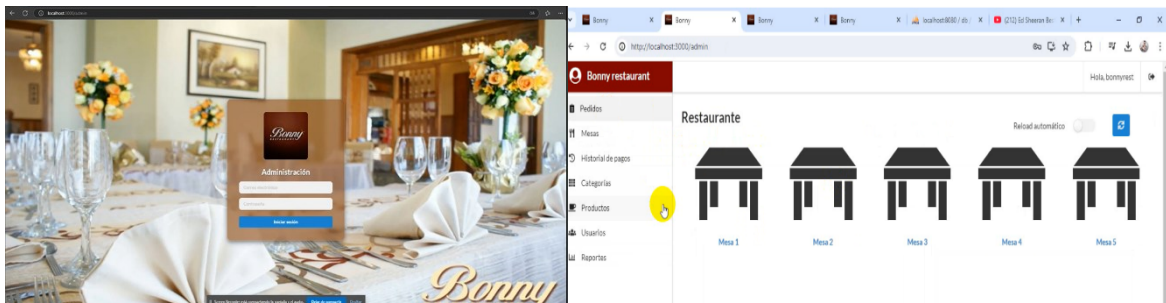


Figura 17: Gestión de caja aplicación web

La Figura 18, muestra la interfaz emergente de confirmación al intentar cancelar un pedido en el módulo Gestión de Caja del sistema.



Figura 18: Pantalla de hacer pedido

En la Figura 19, se visualiza el entorno de Visual Studio Code, donde se ha realizado la implementación del código JavaScript.

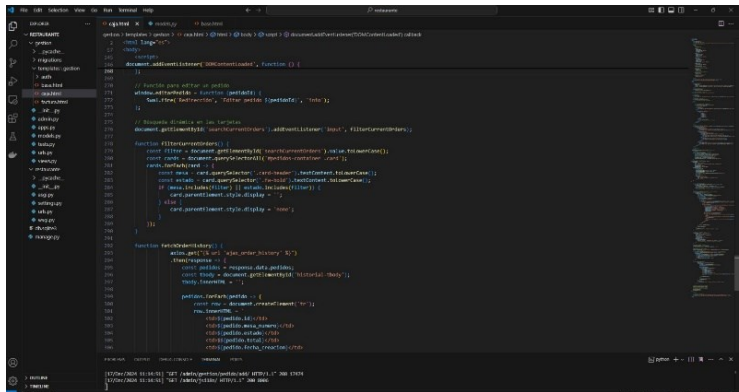


Figura 19: Código aplicación web

La Imagen 20 es el módulo de Gestión de Mesas del sistema. Esta es la pantalla para que el mesero visualice el estado actual de cada mesa.

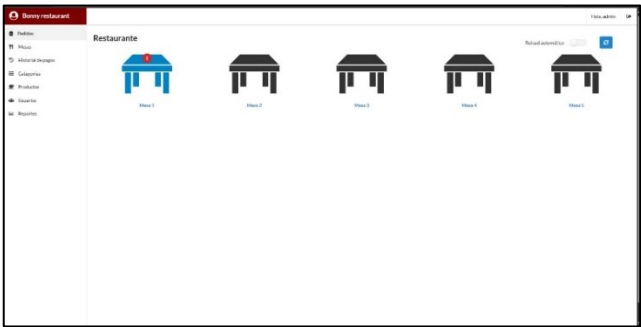


Figura 20: Gestión de mesas

La Imagen 21 ilustra el módulo de Gestión de Categorías del sistema, en este módulo, el administrador puede organizar las categorías de productos disponibles en el establecimiento.

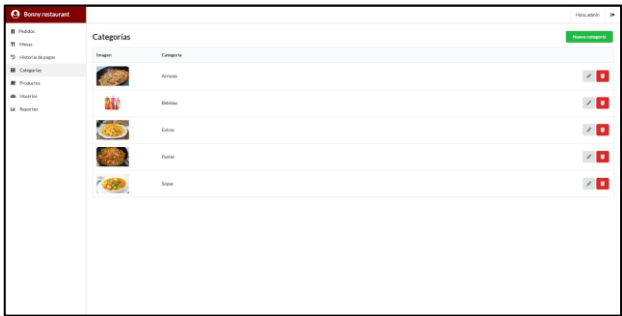


Figura 21: Gestión de categorías de productos

La imagen 22, ilustra la interfaz del carrito de compras que permite a los meseros revisar los detalles de los pedidos antes de confirmar.

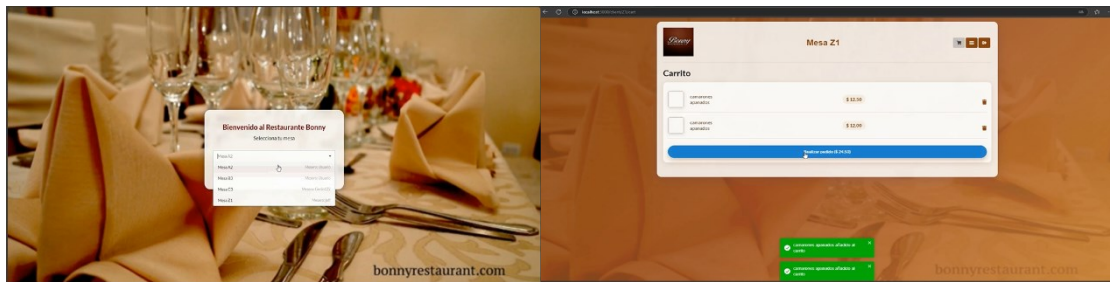


Figura 22: Pedido por mesa

3.7.5 Perfeccionamiento continuo (Kaizen)

En esta etapa se analiza el desempeño real del sistema de comandas una vez que ya está en funcionamiento y se proponen mejoras al flujo de trabajo.

Al tener retroalimentación de los usuarios, se puede identificar opciones de mejora en los procesos de cada módulo. Con cada ciclo de mejora continua, se optimizan tanto las funcionalidades existentes como el diseño de la interfaz y la estructura del sistema, reduciendo desperdicios y aumentando el valor entregado al cliente. En este sentido, la filosofía Kaizen asegura que el proyecto no se estanque tras su implementación inicial, sino que se mantenga en constante evolución para adaptarse eficazmente a las necesidades cambiantes del negocio.

Con el objetivo de optimizar la gestión de los comensales y ofrecer un servicio más personalizado, se incorporó una sección específica para el registro de información del cliente, como se muestra en la Figura 23. A través de esta pantalla el mesero ingresa los datos de los clientes.

Información del Cliente

Nombre

jeferson

Cédula

21332445455656

Correo Electrónico

jeferson.caranqui@unach.edu.ec

Confirmar y cerrar mesa

Cancelar

Figura 23: Información del cliente

Para reducir errores, el sistema puede reasignar las mesas a los clientes, tal y como se muestra en la Figura 24. Este ajuste evita confusiones y contribuye a mantener la eficiencia durante los periodos de alta demanda.

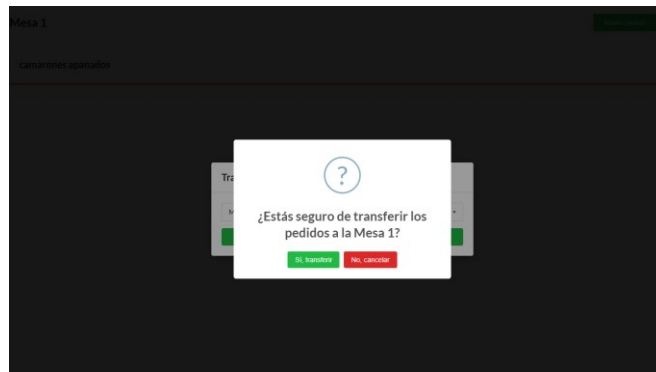


Figura 24: Transferir pedido a otra mesa

Finalmente, se perfeccionó el proceso de generar factura de manera automática, representado en la Figura 25. Esta mejora, alineada con la metodología LEAN, elimina pasos redundantes al unificar la información proveniente de los pedidos y la base de datos de facturación. El sistema consolida el consumo de los clientes, calcula los importes correspondientes y emite un documento listo para su entrega o impresión. Con ello, se reducen significativamente los errores de digitación y el tiempo invertido en la etapa de cobro, de esta forma se alivia la carga de trabajo del personal, brinda transparencia al comensal y mejora la experiencia general de pago. Además, la automatización de este proceso facilita la obtención de métricas financieras que sirven de insumo para futuras decisiones de mejora continua.



FACTURA

Restaurante Bonny
Primera Constituyente 1287-1195
Riobamba, Ecuador
Teléfono: +593998146120
Mesa: A2
Atendido por: Andres

Fecha: 20/5/2025
Factura #: 2
Ref:

DATOS DEL CLIENTE

Nombre: jefferson
Cédula/RUC: 0805260975
Email: jefferson.caranqui@unach.edu.ec

Cantidad	Descripción	Precio unit.	Subtotal
1	Pargo	\$ 10.00	\$ 10.00
1	Gordon Bleu	\$ 10.00	\$ 10.00
1	Gordon Bleu	\$ 10.00	\$ 10.00
1	Pechuga a la plancha	\$ 8.00	\$ 8.00
1	Pechuga a la plancha	\$ 8.00	\$ 8.00

Observaciones:

Subtotal: \$ 40.48

IVA 12%: \$ 5.52

Total: \$ 46.00

Figura 25: Generar factura

Planeación de pruebas

En este proyecto, el enfoque de pruebas se llevó a cabo conforme se iba avanzando en el desarrollo de cada módulo, poniendo en práctica los principios Lean de iteraciones cortas y entrega continua de valor. En la medida en que se implementaba y verificaba el código de

cada módulo, por ejemplo, la gestión de menús, la toma de pedidos o la facturación se ejecutaban pruebas que validaban tanto la funcionalidad recién incorporada como el impacto sobre los componentes ya existentes. Esta estrategia de “pruebas paralelas al desarrollo” permitió detectar y corregir errores de manera inmediata, evitando que estos se acumularan hacia las etapas finales, y asegurando que las distintas secciones de la aplicación web mantuvieran su confiabilidad y coherencia a lo largo de todo el proceso de implementación. También se procedió a realizar pruebas del sistema completo. Estas pruebas finales permitieron verificar no solo el funcionamiento adecuado de cada componente por separado, sino también la correcta interacción entre todos los elementos del sistema que contiene. La Figura 26, el Sistema de Gestión de Pedidos para Restaurante permite un flujo eficiente donde el mesero ingresa sus credenciales, selecciona el número de mesa correspondiente, registra los platillos solicitados y envía el pedido directamente a cocina con un solo clic. La pantalla muestra claramente el nombre del mesero y número de mesa para cada orden, permitiendo al personal de cocina identificar inmediatamente la procedencia de cada pedido y priorizar según orden de llegada.

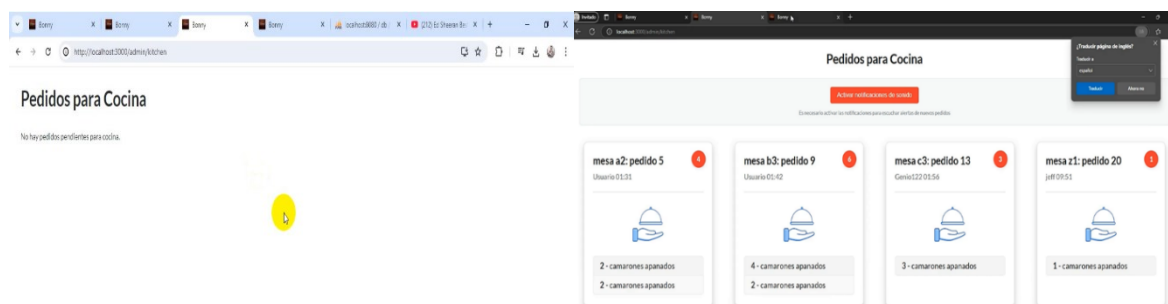


Figura 26: Pruebas realizadas al sistema

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1 Resultados

Para garantizar que la aplicación mantenga un rendimiento eficiente bajo condiciones de alta demanda, se utilizó la versión 5.5 de Apache JMeter, una herramienta de pruebas de carga y rendimiento de código abierto que ofrece funciones de prueba distribuida, amplia personalización de escenarios y generación automática de reportes detallados. Mediante la configuración de distintos niveles de concurrencia (10, 50, 100 y hasta 500 usuarios), duraciones de prueba (5, 10 y 30 minutos) y tiempos de “pensamiento” que simulan el comportamiento real, JMeter permitió medir de forma precisa indicadores como el porcentaje de eficacia, el tiempo de respuesta y la utilización de recursos del sistema. Estas funcionalidades posibilitan identificar cuellos de botella y supervisar el consumo de CPU, memoria y ancho de banda, brindando información valiosa para optimizar el flujo de la aplicación, reducir tiempos de espera y asegurar la escalabilidad del servicio, todo ello alineado con los principios LEAN de eficiencia y mejora continua.

Tabla 16: Mediciones de JMeter

Indicador	Descripción
% Eficacia	Evalúa el desempeño del sistema.
% Tiempo de respuesta	Tiempo que requiere el sistema para responder a las solicitudes.
% Utilización de Recursos	Uso eficiente del CPU, memoria y el ancho de banda.

Evaluación de rendimiento

La eficacia se determinó mediante el cálculo del porcentaje de solicitudes procesadas exitosamente, teniendo un valor de 99%. El tiempo de respuesta se midió con el tiempo requerido por el sistema para tener una respuesta, el límite establecido fue de 2 segundos en el 95% de las solicitudes. Finalmente, se evaluó la utilización de recursos, lo óptimo es que el uso de CPU y memoria no superara el 75% en la mayor carga.

La Figura 27 se ven las pruebas HTTP (GET, POST, PUT y DELETE), organizadas por operación y endpoint. En el Summary Report hay indicadores como el número de muestras, tiempos de respuesta promedio y el porcentaje de error. Estas métricas permitieron detectar cuellos de botella y verificar que el sistema sea adecuado en condiciones reales.

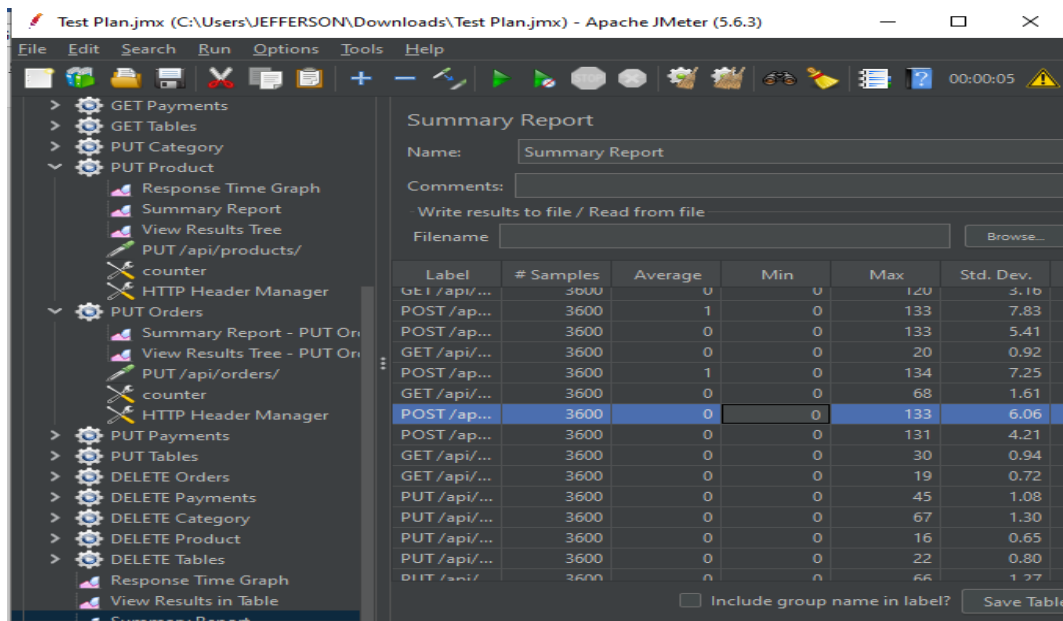


Figura 27: Plan de pruebas

En la Figura 28, el análisis de rendimiento mediante Apache Jmeter registra 8000 solicitudes POST exitosas al endpoint /api/tables con un tiempo de respuesta promedio de 159 ms ($\sigma=261$ ms). Las métricas críticas revelan latencias aceptables para procesamiento de transacciones, con tiempos de conexión TCP optimizados. La estabilidad del servicio bajo este volumen transaccional sugiere un dimensionamiento adecuado de recursos, aunque la desviación estándar indica necesidad de optimización para escenarios de mayor concurrencia. Los parámetros de throughput y bytes transferidos demuestran eficiencia en la serialización de datos y en los procesos de comunicación cliente-servidor.

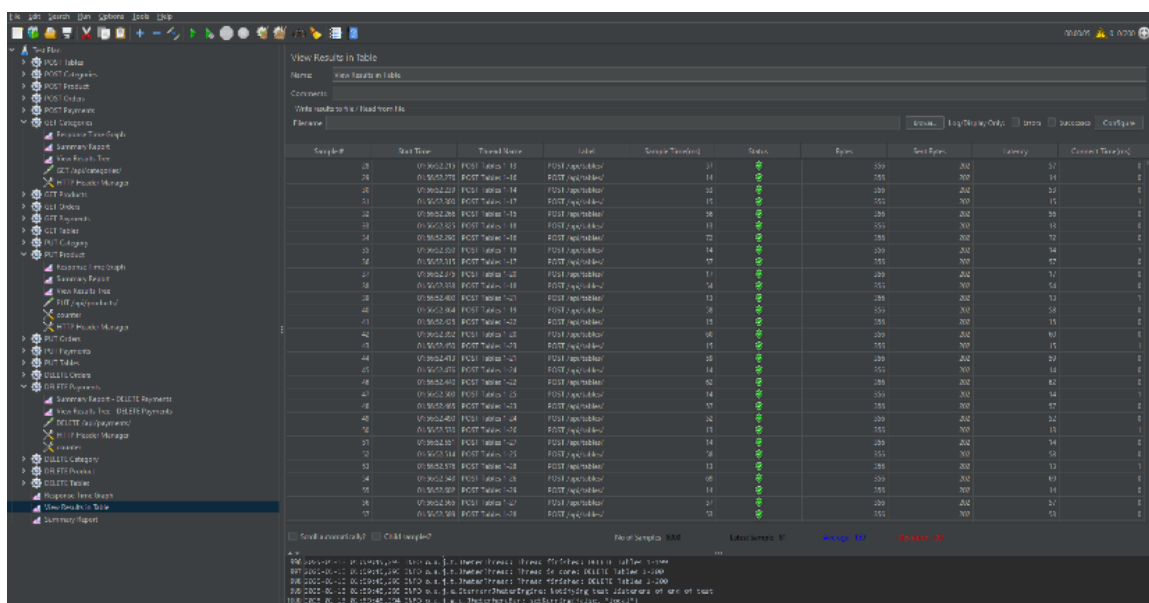


Figura 28: View Results in Table

La Figura 29, presenta un Thread Group configurado para el endpoint POST /api/tables con 200 usuarios concurrentes, ramp-up de 5 segundos y 2 iteraciones por usuario. Esta configuración técnica evalúa la respuesta del sistema bajo carga controlada, manteniendo persistencia de sesión para simular con precisión operaciones reales de creación de mesas en el sistema

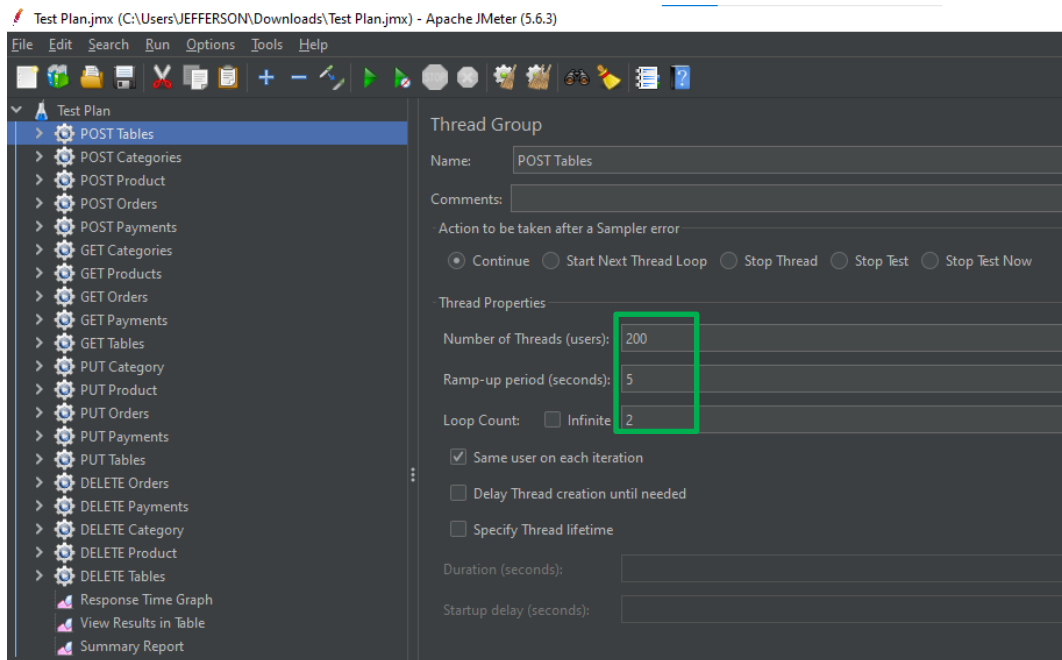


Figura 29: Thread Group

Valoración de indicadores

En esta sección se presentan indicadores del desarrollo del sistema.

Tiempo de desarrollo

Se necesitó un tiempo de cuatro meses para desarrollar las diferentes funcionalidades.

Número de líneas de código

En el desarrollo de este proyecto se implementaron 17554 líneas de código para cumplir con todas las funcionalidades planificadas.

Número de funcionalidades

El sistema de comandas tiene las siguientes 7 funcionalidades:

- Autenticación (Auth)
- Usuarios (Users)
- Categorías (Categories)
- Productos (Products)
- Pedidos (Orders)
- Pagos (Payments)
- Mesas (Tables)

Eficacia

Para obtener el indicador de eficacia, se definió como parámetro principal la proporción de solicitudes que el sistema respondió de manera exitosa frente al total de peticiones realizadas. En la práctica, esto consistió en enviar un número determinado de solicitudes a distintos endpoints —por ejemplo, creando, consultando o actualizando recursos— utilizando la herramienta JMeter. Al concluir la prueba, se contabilizaron las peticiones que retornaron un código de respuesta HTTP 2XX o alguna confirmación de éxito a nivel funcional de la aplicación, y se compararon con el total de solicitudes enviadas. De este modo, la eficacia se calculó aplicando la fórmula:

Eficacia = (Solicitudes exitosas / Solicitudes totales) * 100%

Todas las peticiones recibieron respuestas satisfactorias por parte del sistema, arrojando un resultado del 100% de eficacia. La Tabla 17 muestra los resultados de la eficacia de las solicitudes realizadas al sistema.

Tabla 17: Requerimientos solicitados para eficacia

Parámetros	Indicadores	Requerimientos
Eficacia	Requerimientos exitosos del aplicativo	100%
	Requerimientos fallidos del aplicativo	0%

Tiempo de respuesta

En la Figura 29, se muestra el gráfico comparativo de tiempos de respuesta para métodos HTTP (GET, POST, PUT, DELETE) aplicados a distintos endpoints de una API. Permite identificar variaciones de rendimiento, evaluar eficiencia del backend y detectar posibles cuellos de botella bajo distintas condiciones de carga.

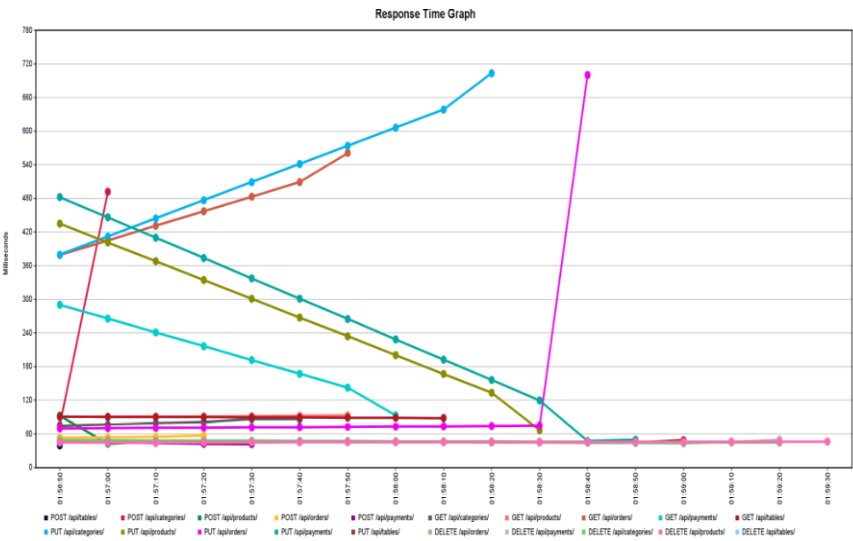


Figura 30: Tiempos de respuesta

Utilización de recursos

En la Figura 31, se observan distintos indicadores de rendimiento del sistema en tiempo real. El uso de CPU oscila alrededor del 2.5%, que representa una carga moderada.

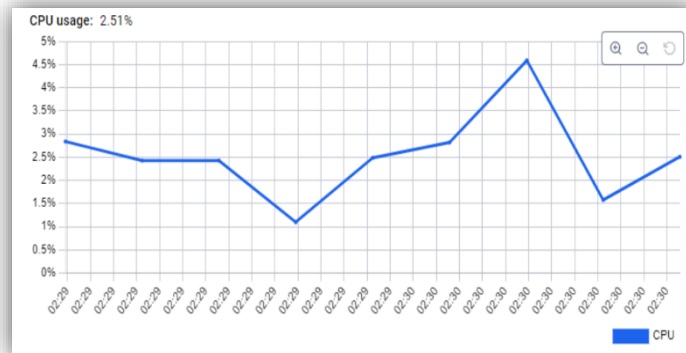


Figura 31: Rendimiento del CPU

La figura 32, el uso de la memoria apenas ronda los 100 MB, quedando la mayor parte de los 7.38 GB disponibles.

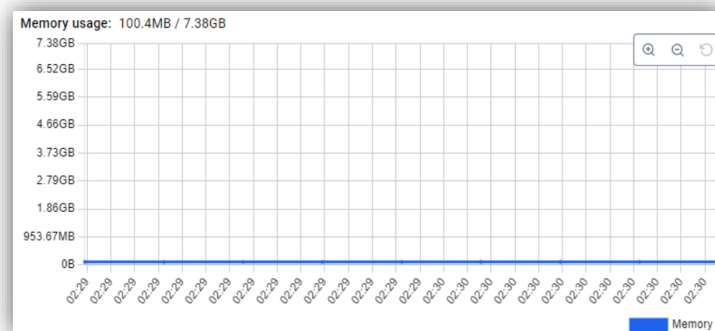


Figura 32: Utilización de la memoria

La figura 33, las métricas de lectura/escritura en disco muestran un consumo prácticamente inexistente (0 B).



Figura 33: Métricas de lectura del disco

La Figura 34, presenta un gráfico de líneas temporales que ilustra el comportamiento de la transferencia de datos en el sistema durante un período de monitoreo continuo. El gráfico muestra dos métricas principales representadas por líneas de diferentes colores: la línea azul corresponde a los datos recibidos (Data received) y la línea naranja representa los datos enviados (Data sent), ambas medidas a lo largo de aproximadamente 30 intervalos de tiempo consecutivos.

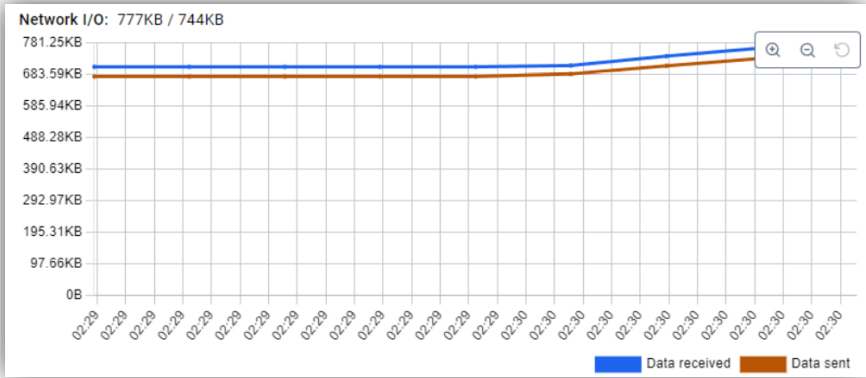


Figura 34: Velocidad de la red

Los resultados indican que la aplicación funciona adecuadamente en memoria y ancho de banda, permitiendo atender la carga de trabajo actual. Además, el margen operativo permitiría escalar el sistema sin afectar el rendimiento.

Valores obtenidos del estudio en base al modelo FURPS

La Tabla 18 presenta los resultados obtenidos basándose en el modelo FURPS.

Tabla 18: Valores obtenidos FURPS	
Parámetros	Valoraciones del estudio
Eficacia	99%
Tiempo de respuesta	4650 ms
Uso de recursos	26%

Al comparar los valores obtenidos se demuestra un desempeño sólido del sistema: la eficacia del 99%, considerado como aceptable; el tiempo de respuesta, fijado en 4650 ms, se mantiene por debajo de los 5000 ms y el uso de recursos, con un promedio de 26%. Estos resultados confirman que la calidad del sistema cumple con los objetivos.

4.2 Discusión

En este estudio, la integración de la metodología LEAN se fundamentó en principios ampliamente validados en la literatura de desarrollo de software. Demostraron los autores Womack y Jones en su investigación sobre principios Lean, junto con estudios posteriores de Poppendieck y Cusumano, que la adopción de prácticas Lean transforma los procesos de desarrollo al reducir desperdicios y acelerar la entrega continua de valor. De manera similar, Kniberg [30] en “Lean from the Trenches” evidenció cómo la implementación de estos principios en equipos de desarrollo mejora significativamente la eficiencia, calidad del software y la capacidad de respuesta ante cambios. Estos trabajos respaldan la decisión de incorporar Lean en los proyectos, donde cada módulo de la aplicación web de comandas fue validado de forma iterativa, permitiendo detectar y corregir errores de manera oportuna y asegurar que las funcionalidades críticas se desplegaran de forma robusta y escalable.

El implementar principios LEAN en el sistema web de comandas se notó en la planificación de pruebas y en la integración continua. La ejecución de pruebas mientras se llevaba a cabo el desarrollo, se utilizaron herramientas como Apache JMeter, esto permitió simular escenarios reales y poder evaluar los indicadores que se definieron en los objetivos: eficacia, tiempo de respuesta y el consumo de recursos. Los resultados obtenidos, no solo se alinean con los estándares de calidad que se describen teóricamente, sino que también reflejan que metodología LEAN es recomendable para optimizar procesos y garantizar una entrega de producto alineada a los requerimientos.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La adopción de la metodología LEAN resultó esencial para la optimización de cada fase de desarrollo de la aplicación web de comandas. Mediante la identificación temprana del valor desde la perspectiva del cliente y la eliminación de pasos que no aportaban utilidad, se logró un flujo de trabajo más ágil y orientado a resultados. Asimismo, la mejora continua, permitió integrar y adaptar de manera progresiva los requerimientos surgidos durante el desarrollo del proyecto.
- El uso de Django, tomando a PHP como lenguaje de programación, proporcionó la facilidad estructurar la lógica de programación y gestionar los datos. De igual manera el frontend, bases de datos y scripts de automatización contribuyeron a alinear el proyecto a los requerimientos. El enfoque modular que se planteó para el desarrollo fue respaldado por el uso de la metodología ágil seleccionada, permitió que el proceso de desarrollo se pueda acelerar e implementar nuevas funcionalidades con mayor facilidad.
- La evaluación del rendimiento del sistema se realizó con la herramienta JMeter y se basó en el modelo FURPS, se obtuvieron resultados positivos. El sistema mostró un 99% de eficacia, un tiempo de respuesta promedio de 4650 ms y un consumo de recursos cercano al 25%. Estos valores obtenidos confirman que la aplicación cumple con los estándares definidos por el modelo FURPS, el cual describe valores óptimos de Funcionalidad, Usabilidad, Confiabilidad, Rendimiento y Soporte técnico del software.

5.2 Recomendaciones

- Se recomienda mantener y fortalecer la aplicación de los principios de la metodología LEAN a lo largo del ciclo de vida de la aplicación. Esto supone una búsqueda continua de oportunidades de mejora en cada fase del desarrollo, la priorización de aquellas funcionalidades que generen mayor valor para el cliente y la evaluación constante de los procesos, con el objetivo de eliminar actividades que no aporten valor significativo.
- Es recomendable que para desarrollos o mejoras futuras se mantenga un enfoque híbrido y flexible, aprovechando los beneficios que prestan los framework como Django en fusión con PHP. La experiencia que se obtuvo en el desarrollo del proyecto evidencia que esta integración ha favorecido considerablemente la manipulación de datos y la implementación adecuada de la lógica de negocio. Además, es importante continuar usando una arquitectura modular y metodologías ágiles como la metodología LEAN considerando que se han demostrado que esta metodología permitió agilizar el desarrollo y facilitar que el sistema pueda ampliarse según las necesidades a futuro del negocio.
- Se recomienda continuar empleando JMeter para las evaluaciones del sistema. Esta herramienta al proporcionar un entorno gráfico y sencillo para realizar configuraciones y así llevar a cabo pruebas de carga, haciendo posible tener adaptaciones que se acoplen a los requerimientos del proyecto. Integrando JMeter con el modelo de calidad FURPS, se crea un esquema de evaluación que toma en cuenta todos los elementos que son necesarios para determinar el rendimiento de la aplicación web.

BIBLIOGRAFÍA

1. A. Pérez, "Desafíos en la gestión tradicional de comandas en restaurantes," *Revista de Administración Gastronómica*, vol. 12, no. 3, pp. 45-52, 201
2. L. Gómez y M. Sánchez, "Impacto de los sistemas POS en la eficiencia operativa de restaurantes," *Journal of Hospitality and Tourism Technology*, vol. 9, no. 1, pp. 78-90, 2019.
3. S. Torres, "Análisis de datos en la industria de la restauración," *IEEE Latin America Transactions*, vol. 18, no. 7, pp. 1120-1127, 2020.
4. J. Martínez, "Integración de sistemas de gestión con programas de lealtad en restaurantes," *International Journal of Contemporary Hospitality Management*, vol. 31, no. 3, pp. 1200-1215, 2019.
5. D. López, "Aplicaciones web para la gestión de comandas en el sector gastronómico," *IEEE Access*, vol. 8, pp. 110200-110210, 2020.
6. K. Rivera, "Transformación digital en la industria de servicios: Caso de estudio en restauración," *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 789-793, 2021.
7. M. L. Markus y M. S. Silber, "A foundation for the study of IT effects: A new look at DeSanctis and Poole's concepts," *Journal of the Association for Information Systems*, vol. 9, no. 10, pp. 609-632, Oct. 2008.
8. T. O'Reilly, "What is Web 2.0: Design patterns and business models for the next generation of software," *International Journal of Digital Economics*, no. 65, pp. 17-37, Mar. 2007.
9. P. Herzum y O. Sims, *Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise*, Wiley, 2000.
10. A. Boiko, *Content Management Bible*, 2nd ed., Wiley, 2005.
11. P. Mell y T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology*, Gaithersburg, MD, NIST Special Publication 800-145, Sep. 2011.
12. L. Firtman, *Programming the Mobile Web*, 2nd ed., O'Reilly Media, 2013.
13. A. McAfee, "Enterprise 2.0: The dawn of emergent collaboration," *MIT Sloan Management Review*, vol. 47, no. 3, pp. 21-28, Spring 2006.
14. G. Shanks et al., "Data quality and data governance," *Communications of the ACM*, vol. 53, no. 9, pp. 45-49, Sep. 2010.

15. K. C. Laudon y C. G. Traver, E-commerce 2018: Business, Technology, Society, 14th ed., Pearson, 2018.
16. P. Kotler, H. Kartajaya y I. Setiawan, Marketing 4.0: Moving from Traditional to Digital, Wiley, 2017.
17. S. Kimes, "The future of restaurant revenue management," Journal of Revenue and Pricing Management, vol. 10, no. 2, pp. 189-194, 2011.
18. M. Whitman y H. Mattord, Principles of Information Security, 6th ed., Cengage Learning, 2017.
19. M. L. Markus, "Technochange management: Using IT to drive organizational change," Journal of Information Technology, vol. 19, no. 1, pp. 4-20, Mar. 2004.
20. J. P. Kotter, "Leading change: Why transformation efforts fail," Harvard Business Review, vol. 85, no. 1, pp. 96-103, Jan. 2007.
21. G. Westerman, D. Bonnet y A. McAfee, Leading Digital: Turning Technology into Business Transformation, Harvard Business Review Press, 2014.
22. P. Boccia, "Restaurant Management: The Role of Technology in Streamlining Operations," Journal of Hospitality Management, vol. 7, pp. 142-150, 2021.
23. L. Rivera, "The Integration of POS Systems in Modern Restaurants," Gastronomy Tech Review, vol. 9, no. 3, pp. 52-59, 2020.
24. M. Gonzalez, "Real-Time Order Management in the Food Industry," International Journal of Food Services, vol. 15, pp. 198-206, 2019.
25. S. Patel, "Financial Oversight Through ERP in Hospitality," Journal of Business and Technology, vol. 6, pp. 32-40, 2021.
26. T. Berger, "ERP and Payment Integration: A Solution for Restaurant Management," Digital Trends in Hospitality, vol. 5, no. 2, pp. 77-85, 2022.
27. J. Thompson, "Data-Driven Decision Making in Restaurants," Hospitality Data Analytics, vol. 4, pp. 89-95, 2020.

ANEXOS

Anexo 1: Acta de entrega del sistema



CERTIFICACIÓN

Riobamba, 07 de abril de 2025

A petición de la parte interesada; yo Sr. Héctor Gonzales Velásquez, Gerente General del Restaurante Bonny ubicado en la ciudad de Riobamba.

Certifico:

Que, el señor **JEFFERSON ROLANDO CARANQUI CARANQUI**, portador del número de cédula de ciudadanía No. 0605260975, estudiante de la carrera de Ingeniería en Tecnologías de la Información de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo, ha diseñado, desarrollado e implementado con excelencia una aplicación web integral para el registro y gestión de comandas en el Restaurante Bonny, cumpliendo con todos los requerimientos técnicos y funcionales establecidos. Dicho sistema forma parte de su trabajo de Titulación denominado: **"APLICACIÓN WEB PARA GESTIONAR REGISTROS DE COMANDAS PARA ALIMENTOS Y BEBIDAS EN EL RESTAURANTE BONNY UTILIZANDO LA METODOLOGÍA ÁGIL LEAN"**, demostrando un alto nivel de competencia técnica, profesionalismo y compromiso durante todo el proceso de desarrollo.

Es todo cuanto puedo certificar en honor a la verdad.

Sr. Héctor Gonzales Velásquez
GERENTE

C.I.0600962450

Bonny
RESTAURANT
FIRMA AUTORIZADA

Anexo 2: Manual de Usuario

Sistema de Gestión de Comandas para el Restaurante “Bonny”

Manual de Usuario

VERSIÓN

1.0.0

FECHA DE CREACIÓN

26/04/2025

REALIZADO POR

Jefferson Rolando Caranqui Caranqui

Contenido

1. Introducción	62
2. Visión general del sistema.....	62
3. Administrador.....	62
4. Mesero	70
5.Cocina.....	73
6. Cliente.....	74

Ilustraciones

Ilustración 1: login	62
Ilustración 2: Mesas	63
Ilustración 3: Opciones del pedido.....	63
Ilustración 4: Ingreso de mesa	64
Ilustración 5: Gestionar mesas.....	64
Ilustración 6: Historial de pagos	65
Ilustración 7. Detalle de pago	65
Ilustración 8: Nueva categoría	66
Ilustración 9: Gestionar categorías.....	66
Ilustración 10: Ingresar un nuevo producto.....	67
Ilustración 11: Gestionar productos	67
Ilustración 12: Administrar usuarios.....	68
Ilustración 13: Agregar usuarios.....	68
Ilustración 14: Reportes.....	69
Ilustración 15: Login-móvil.....	70
Ilustración 16: Pantalla principal-móvil	71
Ilustración 17: Pedido por mesa-móvil	72
Ilustración 18: Pedido en cocina.....	73
Ilustración 19: Pantalla de inicio del cliente.....	74
Ilustración 20: Seleccionar lo que se va a consumir del menú	75
Ilustración 21: Pedido en curso.....	76
Ilustración 22: Confirmar método de pago.....	76
Ilustración 23: Pedir cuenta	77

Introducción

Este documento está destinado para el usuario final del sistema de gestión de comandas del restaurante “Bonny”, proporcionando una descripción detallada de cada módulo diseñado para facilitar su uso. Incluye instrucciones específicas sobre todas las funciones y características del sistema.

Visión general del sistema

El sistema está desarrollado utilizando el framework Python Django, elegido por sus características potentes para la creación de aplicaciones que requieren un procesamiento eficiente de datos. Este proyecto surge como respuesta a la necesidad de implementar tecnologías modernas que ofrezcan una solución sólida y confiable para la gestión de comandas en el restaurante “Bonny”.

Administrador

1 Pantalla principal de inicio de sesión.

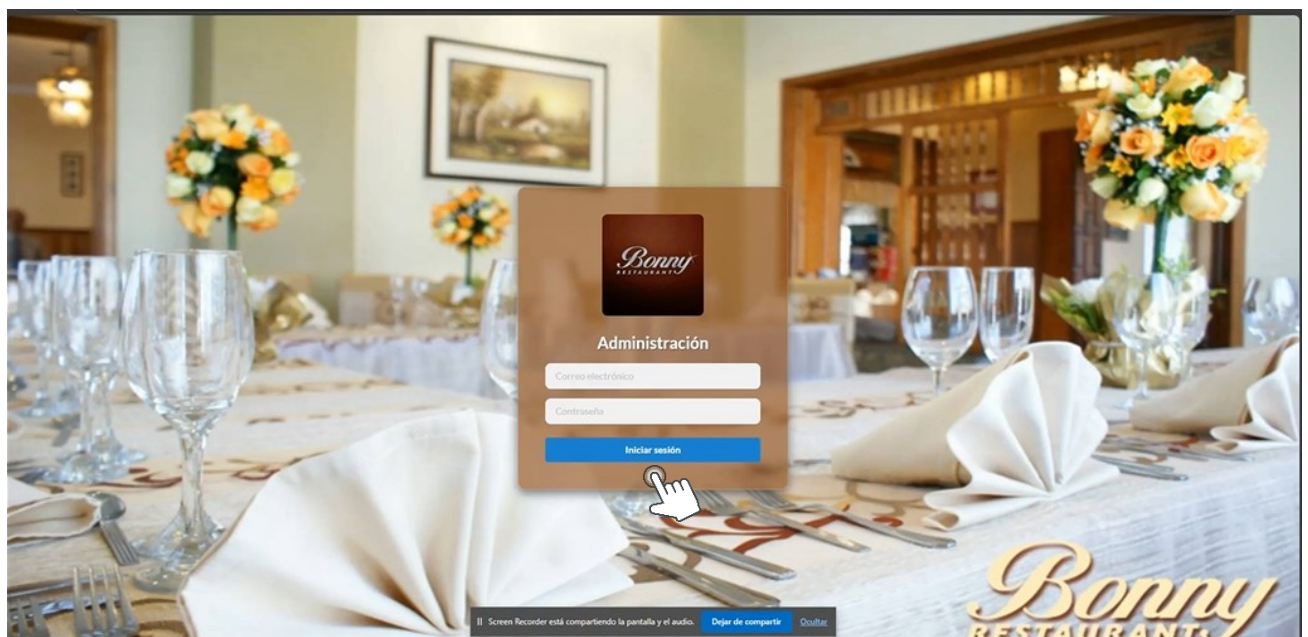


Ilustración 1: login

2

Después de iniciar sesión, el administrador accede a la pantalla principal, que muestra las mesas del restaurante de forma organizada. Desde esta interfaz, puede gestionar pedidos, generar cuentas y transferir mesas.

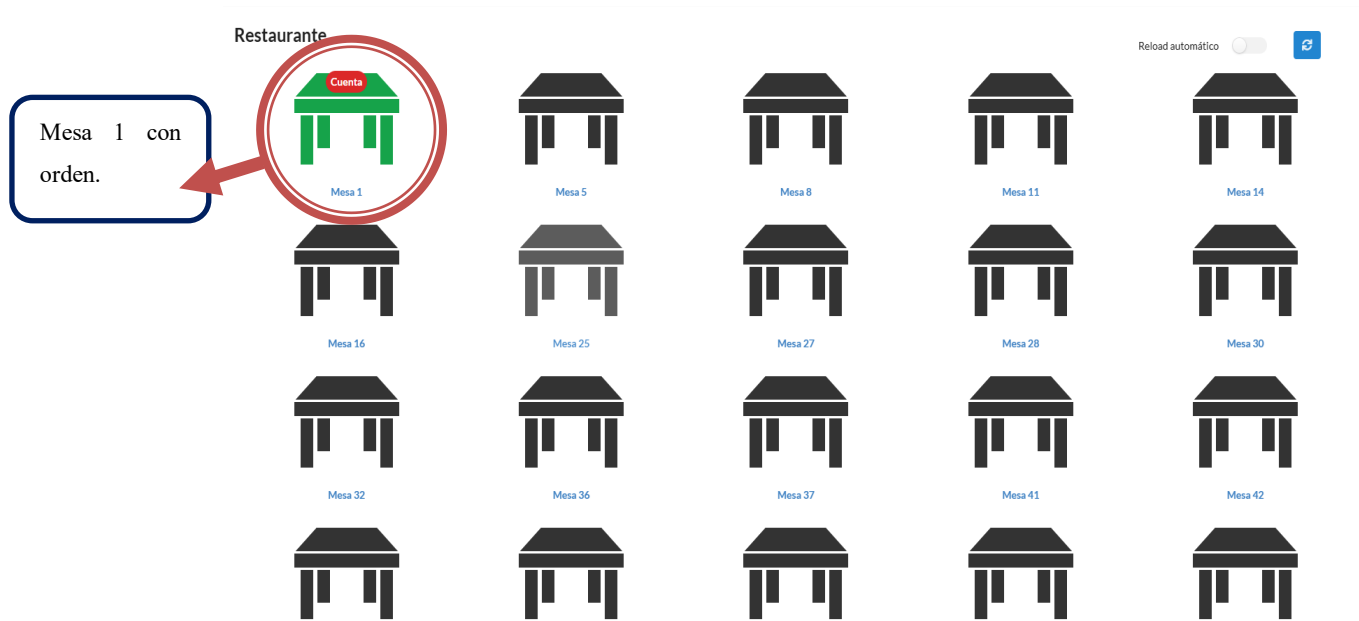


Ilustración 2: Mesas

Al acceder a la mesa, se puede: añadir pedidos, generar cuenta y transferir cuenta a otra mesa.

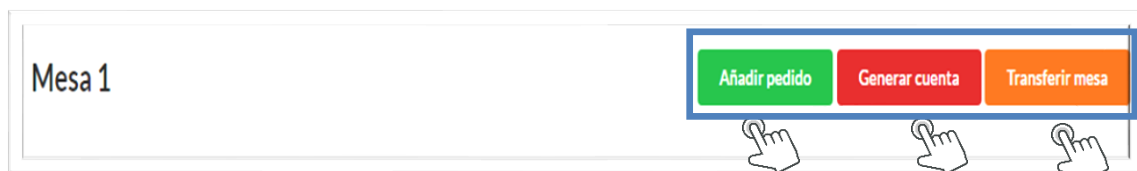


Ilustración 3: Opciones del pedido

- 3 En la sección de mesas, es posible gestionar las mesas del restaurante, permitiendo ingresar nuevas mesas, eliminarlas o editarlas según sea necesario.

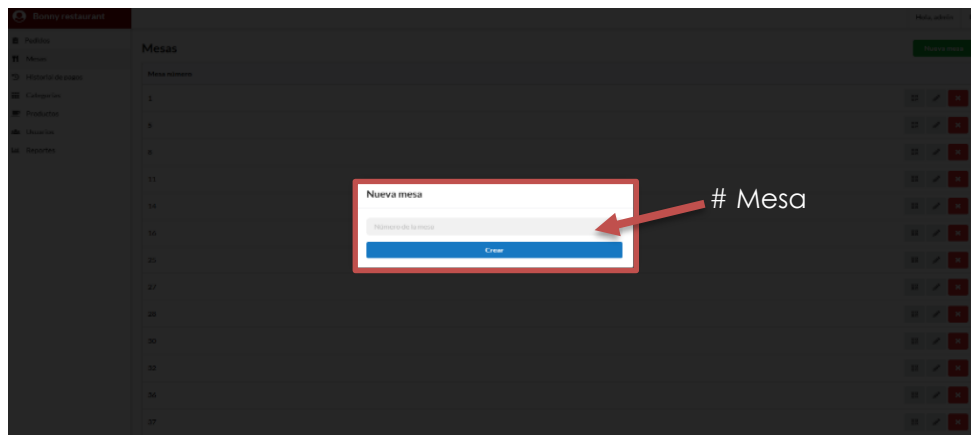


Ilustración 4: Ingreso de mesa

Muestra el listado numerado de mesas con botones para generar código QR, editar y eliminar cada mesa. Incluye botón "Nueva mesa" en la esquina superior derecha.

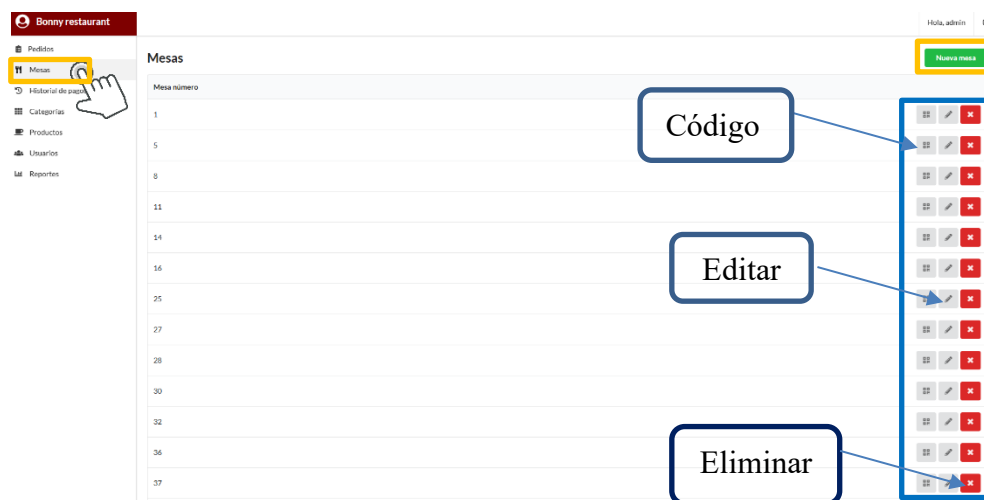


Ilustración 5: Gestionar mesas

4

En la sección de historial de pagos, se muestra una lista de todos los pagos realizados por los clientes, con la opción de visualizar el detalle de cada transacción, incluyendo los productos vendidos.

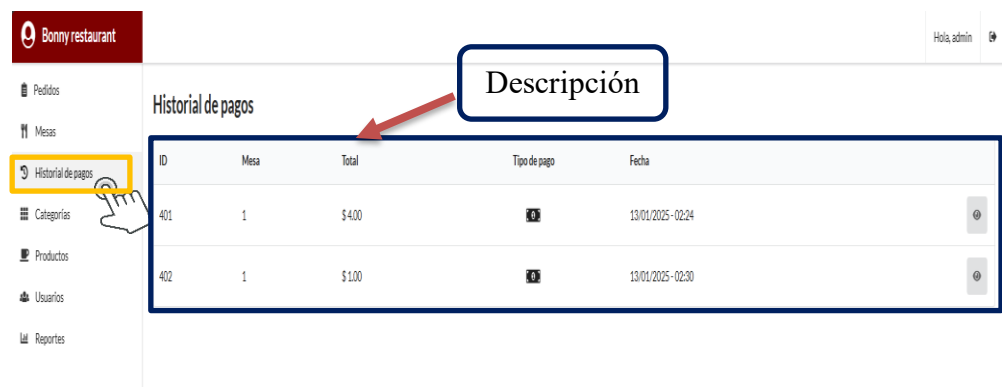


Ilustración 6: Historial de pagos

Pantalla de gestión de pedidos. Muestra el detalle de productos ordenados por mesa. Incluye precios y opción para registro de cobro.



Ilustración 7. Detalle de pago

- 5 En la sección de categorías, se pueden gestionar las categorías del restaurante, permitiendo agregar, editar o eliminar categorías, junto con la opción de asociarles una imagen representativa.

El formulario 'Nueva categoría' contiene los siguientes elementos:

- Un campo de texto etiquetado 'Nombre de la categoría'.
- Un botón etiquetado 'Subir imagen'.
- Un botón azul etiquetado 'Crear'.

Un recuadro amarillo con el texto 'Importante Completar los campos' tiene flechas que apuntan a los campos de nombre y imagen. El botón 'Crear' está rodeado por un recuadro rojo y una mano cursor apunta a él.

Ilustración 8: Nueva categoría

Muestra listados de categorías de alimentos con imágenes representativas. Incluye opciones para editar y eliminar cada categoría, y un botón para crear nueva categoría.

La interfaz de gestión de categorías de Bonny restaurant muestra:

- Un menú lateral con opciones: Pedidos, Mesas, Historial de pagos, **Categorías** (destacado), Productos, Usuarios y Reportes.
- Un encabezado con el nombre 'Bonny restaurant' y un saludo 'Hola, bonnyrest'.
- Un botón verde 'Nueva categoría' en la esquina superior derecha.
- Una tabla con las siguientes categorías:

Imagen	Categoría	Acciones
	Carnes	Editar, Eliminar
	Jugos naturales	Editar, Eliminar
	Mariscos	Editar, Eliminar
	Pastas	Editar, Eliminar
	Platos Típicos	Editar, Eliminar

Las acciones 'Editar' y 'Eliminar' están etiquetadas con recuadros azules y flechas rojas que apuntan a los iconos correspondientes en la columna de acciones.

Ilustración 9: Gestionar categorías

- 6 En la sección de productos, se pueden gestionar los productos del restaurante, permitiendo agregar, editar o eliminar productos. Cada producto debe asociarse a una categoría específica y configurarse como activo para que esté disponible en el sistema.

Nuevo producto

Nombre del producto

Precio del producto

Categoría del producto

☐ Producto activo

Crear

Importante completar los campos

Ilustración 10: Ingresar un nuevo producto

Productos

Producto	Precio	Categoría	Activo
Churrasco	\$ 8	Carnes	
Lomo Apanado	\$ 9	Carnes	✓
Jarra de limonada	\$ 4	Jugos naturales	✓
Jarras de Naranja	\$ 4	Jugos naturales	✓
Vaso de tomate	\$ 2	Jugos naturales	✓
Camarones al Ajillo	\$ 11	Mariscos	✓
Corvina Apanada	\$ 9	Mariscos	✓
Trucha salmoneada	\$ 10	Mariscos	✓

Nuevo producto

Eliminar

Ilustración 11: Gestionar productos

7

En la sección de usuarios, se pueden gestionar los perfiles de los usuarios del sistema, permitiendo agregar, editar o eliminar usuarios. Cada usuario debe asignarse a un rol específico, ya sea administrador o mesero.



Ilustración 12: Administrar usuarios

Formulario de creación de usuario que indica completar todos los campos antes de presionar "Crear".

Nuevo usuario

admin@admin.com

Correo electrónico

Nombre

Apellidos

.....

☒ Usuario activo

☐ Usuario administrador

Importante completar los campos

Crear

Ilustración 13: Agregar usuarios

8

La sección de Reportes Estratégicos proporciona una visión general de los datos clave del restaurante. Incluye métricas como el total de pedidos, ingresos totales, promedio por pedido y el producto más vendido. Además, se presentan gráficos para visualizar la tendencia de pedidos por fecha y los productos más vendidos.



Ilustración 14: Reportes

Mesero

- 1 Al ingresar a la aplicación móvil, se muestra la pantalla de inicio de sesión, donde los meseros deben acceder utilizando sus credenciales correspondientes para poder ingresar al sistema.



Ilustración 15: Login-móvil

- 2 La pantalla principal de la aplicación móvil muestra las mesas del restaurante con íconos y colores para identificar su estado. Los meseros pueden seleccionar mesas para gestionar pedidos y realizar acciones, con opciones de actualización automática o manual.

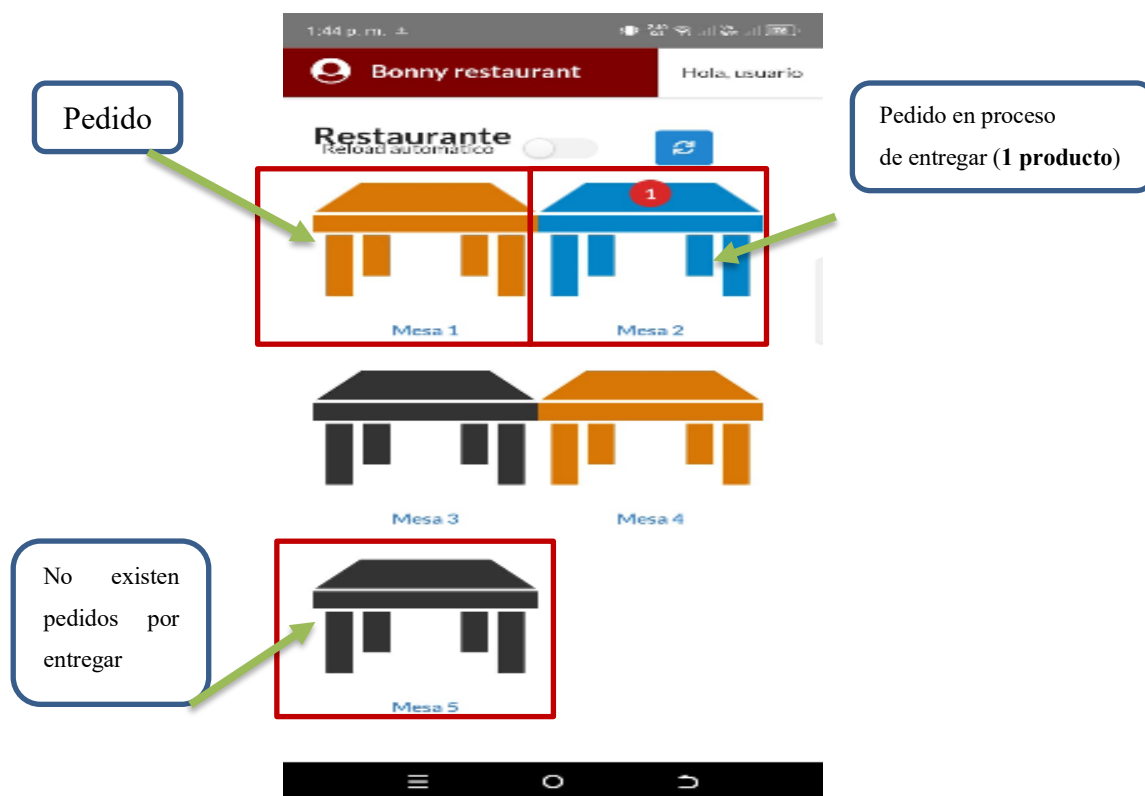


Ilustración 16: Pantalla principal-móvil

- 3 Al seleccionar una mesa, se pueden añadir productos a la mesa, gestionar su estado y marcar si ya fueron entregados, permitiendo un control completo de los pedidos.

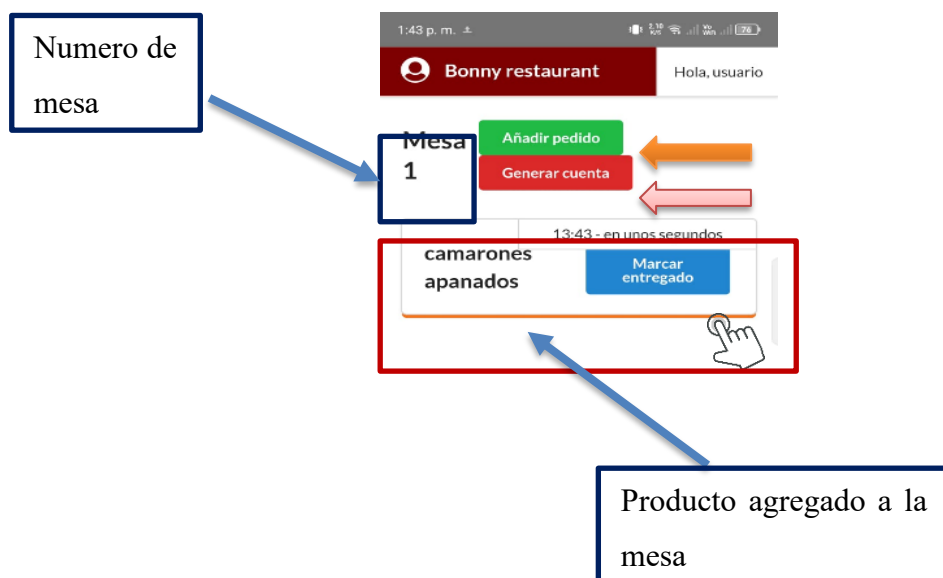


Ilustración 17: Pedido por mesa-móvil

Cocina

- 1 Los pedidos en cocina se gestionan de forma automática, añadiendo y eliminando según corresponda. Además, se muestran los detalles de cada pedido para facilitar su preparación.

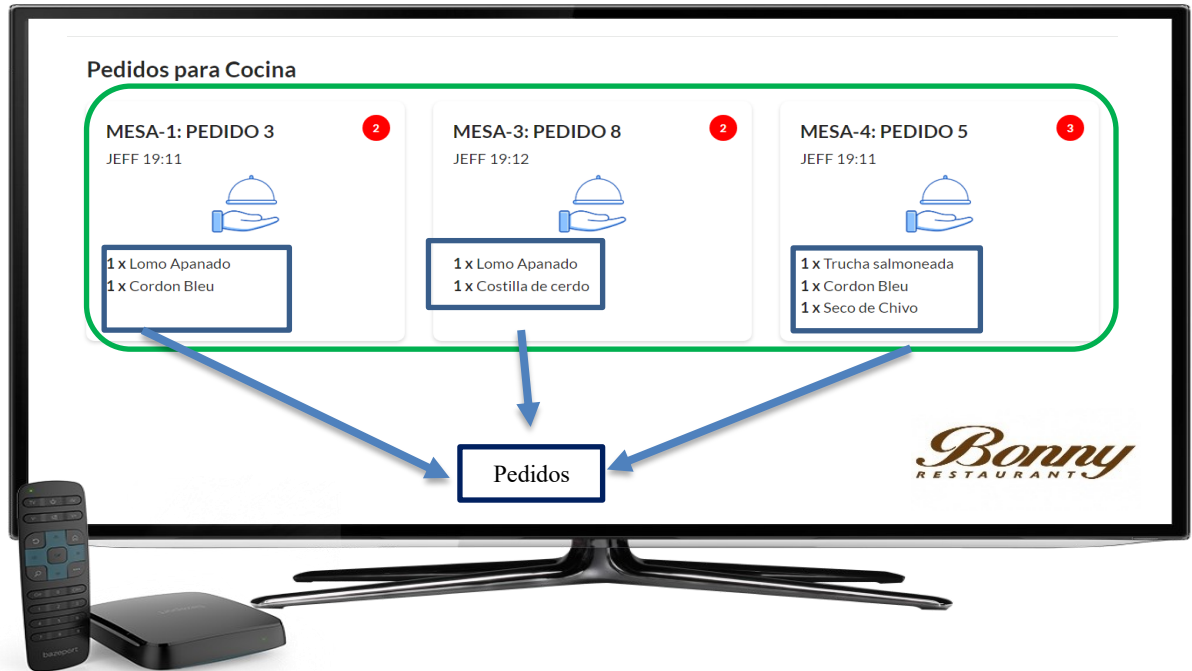


Ilustración 18: Pedido en cocina

Cliente

- 1 Pantalla de inicio del cliente, donde debe seleccionar la mesa.

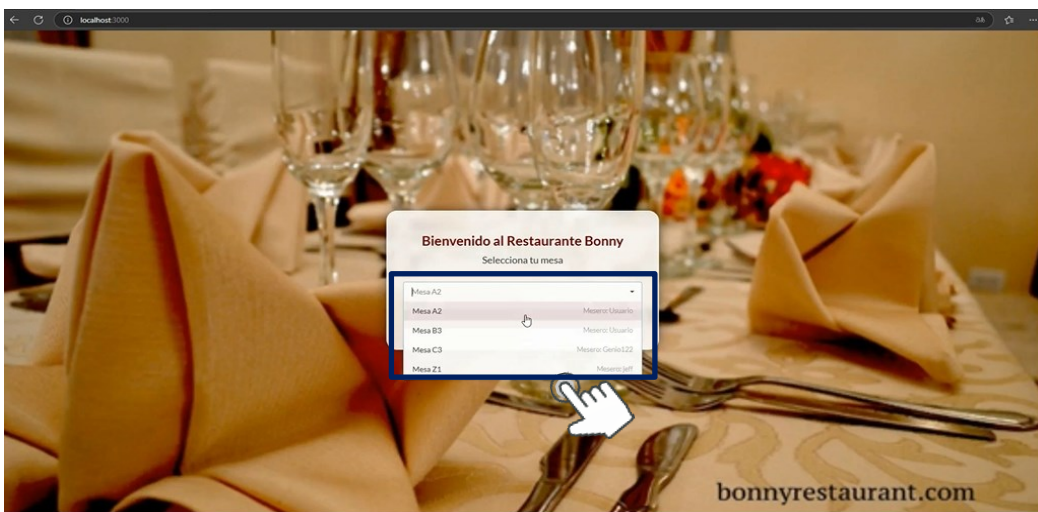
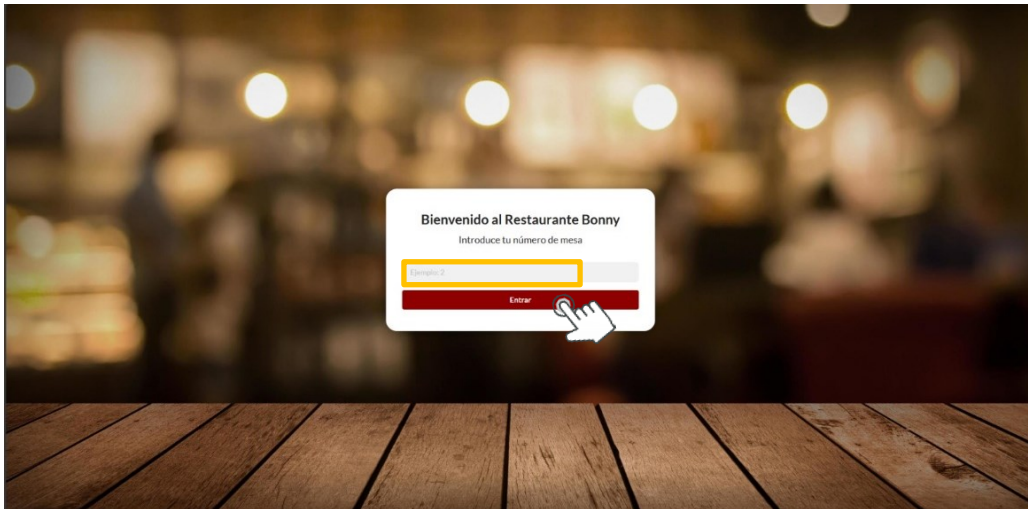


Ilustración 19: Pantalla de inicio del cliente

- 2 Pantalla del sistema de pedidos donde se visualiza el carrito de la *Mesa 2*. Se muestra un producto añadido, el total a pagar y una confirmación en verde indicando que el ítem fue agregado con éxito.

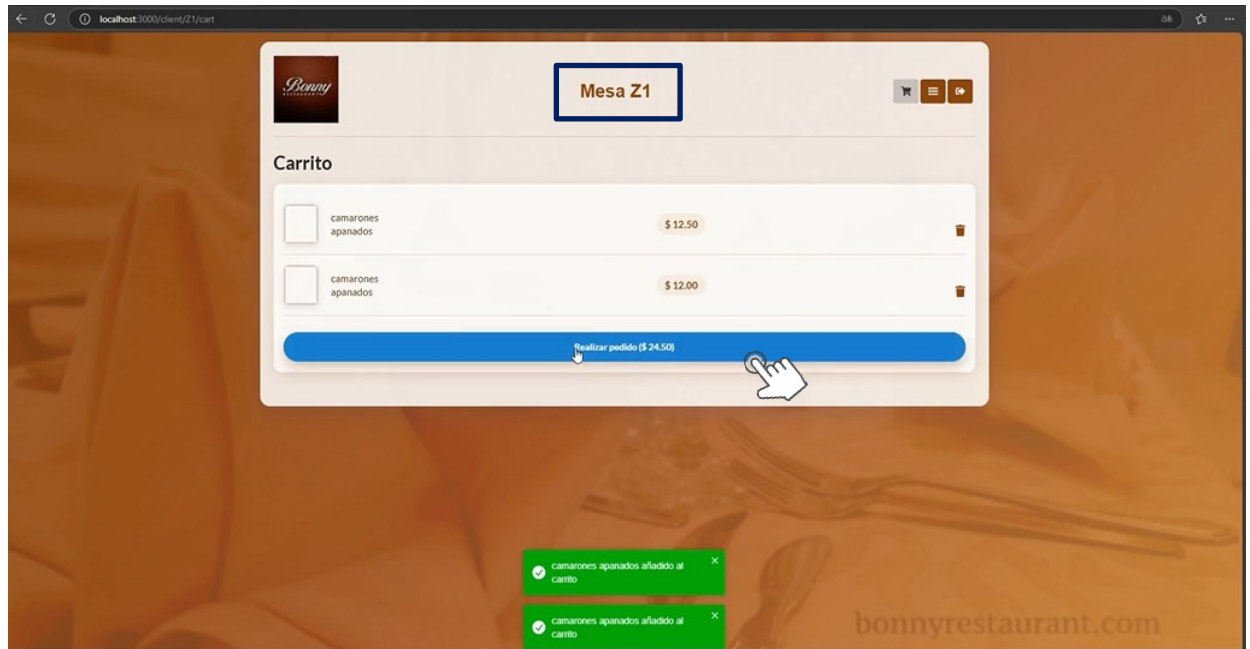


Ilustración 20: Seleccionar lo que se va a consumir del menú

- 3 Interfaz para monitoreo y gestión de pedidos que visualiza ítems solicitados en Mesa 2, con registro cronológico de solicitudes del cliente y herramientas de acción para personal de servicio.

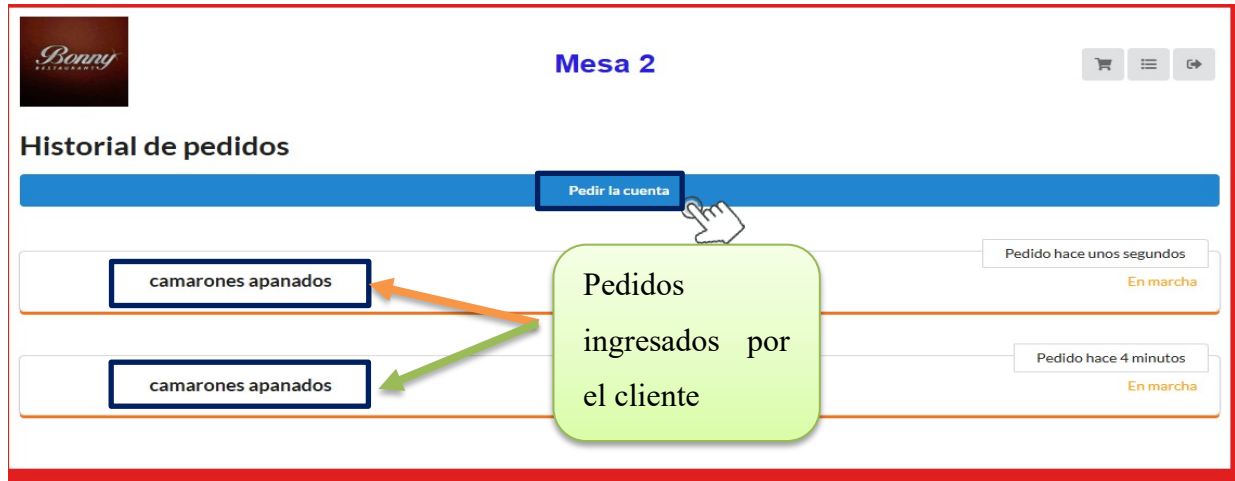


Ilustración 21: Pedido en curso

- 4 Seleccionar el método de pago antes de pedir la cuenta.

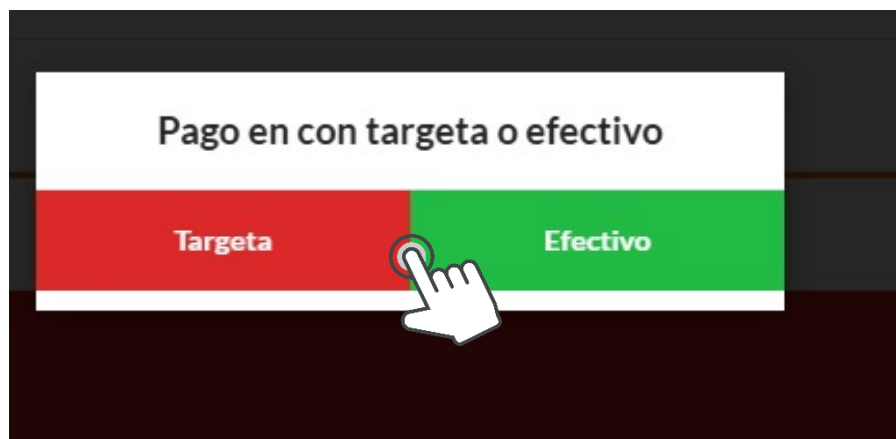


Ilustración 22: Confirmar método de pago

5 Registro de Pedidos del Cliente.



Ilustración 23: Pedir cuenta