



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA INGENIERÍA DE SOFTWARE

**DESARROLLO DE UNA APLICACIÓN WEB PROGRESIVA
PARA LA GESTIÓN DE RESERVA Y PEDIDOS DEL
RESTAURANTE “EL FOGÓN DE COZ” IMPLEMENTANDO
SSR**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar el grado académico de:

INGENIERO EN SOFTWARE

AUTORES: EMILY MICAELA CHIMBO ENCALADA
ANGELLO RAFAEL ORDOÑEZ ZAPATA

DIRECTORA: Ing. GLORIA DE LOURDES ARCOS MEDINA

Riobamba – Ecuador

2022

© 2022, Emily Micaela Chimbo Encalada, Angello Rafael Ordoñez Zapata

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Nosotros, Emily Micaela Chimbo Encalada y Angello Rafael Ordoñez Zapata, declaramos que el presente Trabajo de Integración Curricular es de nuestra autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 14 de diciembre de 2022



Emily Micaela Chimbo Encalada
060475148-7



Angello Rafael Ordoñez Zapata
230036866-5

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA INGENIERÍA DE SOFTWARE

El Tribunal del Trabajo de Integración Curricular certifica que: El trabajo de Integración Curricular; Tipo: Proyecto Técnico, “**DESARROLLO DE UNA APLICACIÓN WEB PROGRESIVA PARA LA GESTIÓN DE RESERVA Y PEDIDOS DEL RESTAURANTE “EL FOGÓN DE COZ” IMPLEMENTANDO SSR**”, realizado por los señores: **CHIMBO ENCALADA EMILY MICAELA, ORDOÑEZ ZAPATA ANGELLO RAFAEL**, ha sido minuciosamente revisado por los Miembros del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
DR. DANILO MAURICIO PASTOR RAMIREZ PRESIDENTE DEL TRIBUNAL		22-12-14
ING. GLORIA DE LOURDES ARCOS MEDINA DIRECTOR DE TRABAJO DE INTEGRACIÓN CURRICULAR		22-12-14
ING. MIGUEL ÁNGEL DUQUE VACA ASESOR DEL TRABAJO DE INTEGRACION CURRICULAR		22-12-14

DEDICATORIA

El presente trabajo me lo dedico principalmente a mí misma, por dar lo mejor de mi cada día, por nunca darme por vencida, por mantenerme responsable y perseverante para que cada sacrificio desde que era niña me haya llevado a este punto de mi vida, sabiendo que me espera un largo camino aún para alcanzar todos mis sueños. A mi padre Silvio, por enseñarme que nada es imposible en esta vida si te lo propones, por todo su amor, por creer en mí y trabajar cada día para que no me falte nada. Y a mi compañero Angello, que no solo me dio su apoyo incondicional en las aulas, sino también fuera de ellas y me enseñó el verdadero significado de la amistad. Espero que nuestros caminos siempre estén enlazados y te agradezco por haber confiado en mí.

Emily

Todo el esfuerzo y desempeño quiero dedicarlo a todas las personas, amigos y familiares, que me han apoyado durante el transcurso de mis estudios y especialmente a mis padres Angel Ordoñez y Marisol Zapata, a mi tía Edith Zapata, que han sido las personas que me han apoyado incondicionalmente, siempre me han dado palabras de aliento, especialmente cuando lo necesitaba, también ellos me han demostrado que con esfuerzo y dedicación todas las metas pueden cumplirse, a mi compañera Emily Chimbo, que ha sido un apoyo durante toda la carrera y que me ha demostrado que es una amiga en quien se puede confiar y espero tener el honor de volver a trabajar con ella. Y especialmente quiero dedicar este trabajo a mi querida Jessica Ballesteros, porque es un ejemplo para seguir, por qué siempre ha creído en mí, incluso cuando dudaba, por su apoyo y ánimo que me brinda para alcanzar nuevas metas, por sus sabios consejos que me han ayudado con mi formación profesional y personal.

Angello

AGRADECIMIENTO

Quiero agradecer a mis tías Janeth y Miriam por enseñarme a ser una mujer independiente y que siempre tengo que pensar en grande, espero que siempre se sientan orgullosas de mí como yo de ellas. A mi madrina Vicky, que me inspiró con toda su sabiduría y me enseñó a ser valiente para ser la mejor versión de mí. A mi mejor amiga María José, que siempre me supo dar su apoyo incluso en los momentos más difíciles para mí. A cada uno de mis profesores que supieron guiar mi camino para amar la carrera más que a nada en este mundo. Y finalmente, quiero agradecer a mi madre, por darme la vida, por ser mi amiga y sobre todo por ser la mejor maestra que he podido tener.

Emily.

Quiero agradecer en primer lugar a Dios por permitirme llegar hasta este punto de mi vida, también quiero agradecer a mis padres porque siempre me apoyaron, en todas las metas que me propongo, a mi tía, por siempre estar pendiente de mí y mi familia, por su apoyo incondicional, de igual forma a mis amigos Emily, Jimmy, Sebastián y Dennys, porque siempre estuvieron presente durante el semestre dando ánimos para acabar este proyecto, a Jessica, por siempre estar conmigo, por darme su amor y atención de forma incondicional, también quiero agradecerme a mi persona, por no rendirme, por todo el esfuerzo que realice para acabar el proyecto, por ultimo agradezco a Jorge Menéndez, Gloria Arcos y Miguel Duque por todas las correcciones y consejos que nos han brindado durante el desarrollo de este trabajo.

Angello.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	x
ÍNDICE DE GRÁFICOS	xi
ÍNDICE DE ANEXOS	xii
RESUMEN	xiii
SUMMARY	xiv
INTRODUCCIÓN	1

CAPITULO I

1	DIAGNÓSTICO DEL PROBLEMA	2
1.1	Antecedentes	2
1.2	Formulación del Problema	3
1.3	Sistematización del Problema	3
1.4	Justificación del Trabajo De Integración Curricular	3
1.4.1	Justificación Teórica	3
1.4.2	Justificación Aplicativa	4
1.5	Objetivos	6
1.5.1	Objetivo General	6
1.5.2	Objetivos Específicos	6

CAPITULO II

2	FUNDAMENTOS TEÓRICOS	8
2.1	Proceso de Sistema de Reservas	8
2.2	Proceso de Sistema de Pedidos	8
2.3	Sistema Informático de Reservas	8
2.3.1	Características de un sistema de reservas informático	9
2.4	Sistema Informático de Pedidos	9
2.4.1	Sistemas de pedidos internos	9
2.4.2	Sistemas de pedidos en línea	10
2.5	Aplicaciones Web	10
2.5.1	Características de las aplicaciones web	10

2.5.2	Clasificación de las aplicaciones web	11
2.6	Aplicaciones Web Progresivas.....	12
2.6.1	Características	12
2.6.2	Componentes.....	12
2.6.3	Web App Manifest.....	13
2.6.4	Service Worker	13
2.7	Técnicas de Renderización en la Web.....	15
2.7.1	Client Side Rendering	15
2.7.2	Server Side Rendering.....	16
2.8	Calidad del Software	18
2.8.1	Modelos de Calidad del Software.....	18
2.8.2	Características y Subcaracterísticas del modelo de calidad 25010	20
2.8.3	Eficiencia	22
2.8.4	Disponibilidad	22
2.9	Trabajos Relacionados	22

CAPITULO III

3	MARCO METODOLÓGICO	25
3.1	Tipo de Estudio	25
3.2	Métodos, Técnicas y Fuentes del Trabajo de Integración Curricular	26
3.2.1	Objetivo 1: Analizar las características del desarrollo de una aplicación web progresiva con SSR.....	27
3.2.2	Objetivo 2: Determinar el proceso actual para la gestión de reservas y pedidos en el restaurante “El fogón de COZ”	29
3.2.3	Objetivo 3: Desarrollar los módulos de pedidos, reservación, menú, autenticación, pagos e inicio que formarán parte de la aplicación web progresiva. mediante la implementación de server side rendering.....	30
3.2.4	Objetivo 4: Evaluar la eficiencia y disponibilidad que tiene la gestión de reservas y pedidos de la aplicación web progresiva a través del modelo ISO/IEC 25000	32
3.3	Aplicación de la metodología Scrumban en el desarrollo del software RESPEDAPP.....	36
3.3.1	Fase 1: Análisis preliminar	37
3.3.2	Fase 2: Planificación.....	45
3.3.3	Fase 3: Implementación	49
3.3.4	Fase 4: Revisión y Retrospectiva	53

3.3.5	Fase 5: Lanzamiento y Cierre	54
-------	------------------------------------	----

CAPITULO IV

4	MARCO DE RESULTADOS Y DISCUSIÓN	56
4.1	Resultado de métricas de eficiencia.....	56
4.1.1	Tiempo de respuesta	59
4.1.2	Rendimiento	59
4.1.3	Matriz de medición del comportamiento temporal	60
4.2	Resultado de métricas de disponibilidad	61
4.2.1	Tiempo de Servicio.....	62
4.2.2	Tiempo medio de inactividad.....	62
4.2.3	Matriz de medición de la disponibilidad.....	63
4.3	Resultados de la evaluación de la calidad de la aplicación RespedApp	64
	CONCLUSIONES	65
	RECOMENDACIONES	66
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1:	Módulos que integran la aplicación web progresiva RESPEDAPP con sus funcionalidades respectivas.	5
Tabla 2-3:	Tabla de descripción del marco metodológico del presente trabajo de integración curricular	26
Tabla 3-3:	Instrumentos que han sido utilizados en el desarrollo e implementación de la aplicación RespedApp	31
Tabla 4-3:	Niveles de puntuación final de calidad para la aplicación RespedApp.	32
Tabla 5-3:	Ponderación de las características de calidad de la aplicación RespedApp	33
Tabla 6-3:	Roles de la Metodología Scrumban para el proyecto RespedApp.....	37
Tabla 7-3:	Roles del aplicativo RespedApp.....	37
Tabla 8-3:	Historias de Usuario requeridas para la aplicación web progresiva RespedApp..	40
Tabla 9-3:	Matriz de riesgos y proceso de mitigación para el proyecto RespedApp	43
Tabla 10-3:	Método T-Shirt para la aplicación web progresiva RespedApp	45
Tabla 11-3:	Estimación de historias de usuario para la aplicación web progresiva RespedApp.	47
Tabla 12-4:	Resultados de toma de tiempos de respuesta con y sin la aplicación RespedApp.	56
Tabla 13-4:	Matriz de medición de calidad de la subcaracterística eficiencia según la ISO/IEC 25000 en la aplicación RespedApp	60
Tabla 14-4:	Matriz de medición de calidad de la subcaracterística disponibilidad según la ISO/IEC 25000 en la aplicación RespedApp	63
Tabla 15-4:	Matriz de resultados de calidad de la aplicación RespedApp.....	64

ÍNDICE DE FIGURAS

Figura 1-2:	Ciclo de vida del <i>service worker</i> de una aplicación web progresiva.....	14
Figura 2-2:	Secuencia o flujo del proceso de renderizado por lado del cliente.....	15
Figura 3-2:	Secuencia o flujo del proceso de renderizado por lado del servidor	16
Figura 4-3:	Proceso actual de reservas del restaurante “El fogón de Coz”	39
Figura 5-3:	Proceso actual de pedidos del restaurante “El fogón de Coz”	40
Figura 6-3:	Estructura del tablero Scrumban utilizado para la gestión de historias de usuario	49
Figura 7-3:	Diagrama jerárquico de módulos de la aplicación Respedapp.....	50
Figura 8-3:	Diagrama de componentes del aplicativo RespedApp	51
Figura 9-3:	Diagrama físico de la base de datos del aplicativo RespedApp	52
Figura 10-3:	Bosquejo final de la sección de la página principal de RespedApp	53
Figura 11-3:	Bosquejo final del menú de RespedApp	53
Figura 12-3:	Diagrama de despliegue de la aplicación RespedApp.....	54
Figura 13-3:	Diagrama de quemado de la aplicación RespedApp	55
Figura 14-4:	Prueba Shapiro-Wilk en R de los datos de tiempo de respuesta con la aplicación RespedApp.....	58
Figura 15-4:	Prueba Wilcoxon en R de los datos de tiempo de respuesta con y sin la aplicación RespedApp	58
Figura 16-4:	Resultado de tiempo de servicio de la aplicación RespedApp en Azure Application Insights.....	62
Figura 17-4:	Resultado de tiempo medio de inactividad de la aplicación RespedApp en Azure Application Insights.....	63

ÍNDICE DE GRÁFICOS

Gráfico 1-4:	Distribución de tiempos de respuesta de la aplicación RespedApp.....	58
Gráfico 2-4:	Diagrama de caja y bigotes con los datos de tiempo de respuesta con y sin la aplicación RespedApp.....	59
Gráfico 3-4:	Comparación entre tiempos de respuesta con y sin la aplicación RespedApp	59
Gráfico 4-4:	Comparación entre el rendimiento con y sin la aplicación RespedApp	60

ÍNDICE DE ANEXOS

ANEXO A: ENTREVISTA

ANEXO B: MANUAL TÉCNICO DE RESPEDAPP

ANEXO C: ACUERDO DE SERVICIO DE RESPEDAPP

ANEXO D: ACTA DE CIERRE DE PROYECTO

ANEXO E: MANUAL DE USUARIO DE RESPEDAPP

RESUMEN

El presente trabajo de titulación tuvo como objetivo el desarrollo de una aplicación web progresiva con la técnica de renderizado por parte del servidor que mejoró la calidad de la gestión de reservas y pedidos del restaurante “El fogón de Coz”. Se realizó una búsqueda bibliográfica en relación con las aplicaciones web progresivas y la técnica de renderizado desde el servidor. Además, se identificó los procesos del restaurante. Posteriormente, se propuso el desarrollo del software con metodología Scrumban, en planificación y estimación se definió 20 historias de usuario y 2 historias técnicas. La aplicación se desarrolló en 4 *sprints* de aproximadamente 4 semanas cada uno, con 8 horas diarias; se estableció utilizar el lenguaje de programación JavaScript con el *framework* NextJS, la arquitectura API REST y el gestor de base de datos MySQL. Después se procedió a evaluar la calidad del sistema utilizando los criterios de eficiencia y disponibilidad según la norma ISO/IEC 25000. Para la eficiencia se evaluó el comportamiento temporal utilizando la técnica de perfilado de velocidad de Google Chrome junto a las pruebas estadísticas: Shapiro-Wilk y Wilcoxon a 41 tiempos de registro de pedidos y reservas que representan el promedio diario de registros del restaurante. Para evaluar la disponibilidad se utilizó la herramienta Azure Application Insights para obtener el tiempo de servicio del sistema. Se obtuvo una aplicación de calidad con un puntaje de 9.99 sobre 10 satisfaciendo las necesidades del usuario y la norma ISO/IEC 25000, además de ser funcional debido a que cumple con todos los requerimientos. Como conclusión se determinó que el uso de PWA con SSR ayuda a mejorar este tipo de procesos. Se recomienda utilizar las diferentes subcaracterísticas de la norma y el uso de frameworks para obtener una mejor evaluación de calidad y realizar el despliegue en nube para reducir costos.

Palabras Clave: <INGENIERÍA DE SOFTWARE>, <APLICACIÓN WEB PROGRESIVA>, <RENDERIZACION DEL LADO DEL SERVIDOR>, <SOFTWARE DE GESTIÓN DE RESTAURANTES>, <MODELO DE CALIDAD ISO/IEC 25000>


Ing. Cristian Castillo



2352-DBRA-UPT-2022

SUMMARY

The objective of the current thesis was the development of a progressive web application with the technique of rendering by the server that improved the quality of the management of bookings and orders of the restaurant "El fogón de Coz". We conducted a literature search regarding progressive web applications and the server-side rendering technique. In addition, we identified the processes of the restaurant. Subsequently, we proposed the development of the software with the Scrum methodology, in planning and estimation 20 user stories and 2 technical stories were defined. We developed the application in 4 sprints of approximately 4 weeks each, with 8 hours a day; It was established to use the JavaScript programming language with the NextJS framework, the REST API architecture and the MySQL database manager. Afterwards, we evaluated the quality of the system using the criteria of efficiency and availability according to the ISO/IEC 25000 standard. For efficiency, the temporary behavior was evaluated using the Google Chrome speed profiling technique together with the statistical tests: Shapiro- Wilk and Wilcoxon to 41 order and booking record times that represent the restaurant's average daily record times. The Azure Application Insights tool was utilized to evaluate the availability in order to obtain the service time of the system. A quality application was obtained with a score of 9.99 out of 10, satisfying the user's needs and the ISO/IEC 25000 standard, as well as being functional because it meets all the requirements. In conclusion, it was determined that the use of PWA with SSR helps to improve this type of process. It is recommended to use the different sub characteristics of the standard and the use of frameworks to obtain a better-quality evaluation and to carry out the deployment in the cloud to reduce costs.

KEY WORDS: <SOFTWARE ENGINEERING>, <PROGRESSIVE WEB APPLICATION>, <SERVER-SIDE RENDERING>, <RESTAURANT MANAGEMENT SOFTWARE>, <ISO/IEC 25000 QUALITY MODEL>

Revised by:



Lic. Carolina Campaña D. Mgs.

ID number: 1804191482

EFL Professor

INTRODUCCIÓN

En la actualidad la gestión de los restaurantes con respecto a cómo se reciben los pedidos y las reservaciones que realizan los clientes ha cambiado gracias al avance tecnológico, es por ello que es común ver que las personas recurran a la tecnología para realizar sus pedidos desde su teléfono o computador, todo esto mediante aplicaciones de terceros que ofrecen servicios de *delivery* como lo son PedidosYa, UberEats, Glovo, entre otros.

Para el restaurante “EL FOGÓN DE COZ” la comodidad de sus clientes son su mayor prioridad, además de brindar eficiencia al momento de que sus requerimientos sean atendidos, es por ello que como objetivo del presente trabajo de integración curricular se ha optado por diseñar e implementar una aplicación web progresiva con la técnica de renderizado por parte del servidor (SSR) que mejore calidad del proceso de pedidos y reservaciones asegurando de que las solicitudes de los clientes serán atendidas por el personal del restaurante todo el tiempo, que los pedidos no tendrán demoras y su tiempo de servicio que aumentará considerablemente.

Para cumplir con el objetivo principal, en el Capítulo 2 se realizó una búsqueda bibliográfica sobre las aplicaciones web progresivas y sus características, se analizó las ventajas que posee la técnica de renderizado por parte del servidor, y en conjunto se investigó acerca de un modelo de calidad que ayude a evaluar las condiciones de la situación previa y posterior a la construcción del aplicativo web. En el Capítulo 3 se ha realizado una recopilación de métodos y técnicas utilizadas para cumplir con cada objetivo planteado y formular así una hipótesis en relación con la calidad de la aplicación a desarrollar. Posteriormente se documenta y describe el desarrollo e implementación de la aplicación web progresiva con SSR utilizando la metodología de desarrollo de software Scrum junto al uso de algunas tecnologías populares utilizadas por varios desarrolladores, como lo es la biblioteca React para frontend, el marco de trabajo NextJS, el entorno de backend NodeJS, el lenguaje de programación JavaScript, herramientas como Git, entre otras. Finalmente, en el Capítulo 4 se exponen resultados, conclusiones y recomendaciones que se han logrado determinar de la medición de calidad según las ponderaciones recomendadas por la norma ISO/IEC 25000 y el desarrollo de la aplicación web progresiva RespedApp.

CAPITULO I

1 DIAGNÓSTICO DEL PROBLEMA

1.1 Antecedentes

El restaurante de comida a la parrilla “El fogón de COZ” ubicado en la ciudad de Baños de Agua Santa, en el sector La Cienega ha servido a la ciudadanía por más de 12 años, siendo este lugar un punto donde familias y turistas llegan a deleitar los diferentes manjares que ponen a su disposición. Sin embargo, como consecuencia de la pandemia por COVID-19 la clientela ha llegado a disminuirse y el restaurante ha optado por realizar una mayor adaptación tecnológica y de esta manera aumentar la competitividad con el mercado.

El problema que radica dentro del restaurante “El fogón de COZ” es la poca eficiencia y disponibilidad de las reservas y los pedidos debido a que se lo realiza de manera rudimentaria, provocando la insatisfacción de los clientes al momento de optar por los servicios que ofrece este restaurante. Actualmente, el restaurante lleva el registro de sus reservas y pedidos mediante la aplicación de WhatsApp por parte del personal, quienes posteriormente deben registrarlos en documentos físicos, los pedidos sufren retrasos o muchas veces pueden llegar a ser olvidados, debido a que el personal debe emplear un mayor esfuerzo al momento de receptar los pedidos y reservaciones, servir la comida, recoger la vajilla y mantener el lugar limpio, además, de que estos documentos físicos podrían llegar a malograrse por el ambiente en el que circulan.

Actualmente existe una porción de restaurantes que cuentan con su propio sistema de gestión de pedidos y/o gestión de reservas, por lo que la clientela se ha inclinado más a optar por la innovación que brindan estos sistemas. Para mejorar la atención a sus clientes se requiere que:

- Las reservaciones puedan realizarse con anticipación y sin necesidad de ir al lugar o contactarse directamente con el personal del restaurante.
- Los alimentos sean cocinados al momento de receptar el pedido, por lo que es necesario tener esta información lo más pronto posible.
- El cliente observe el menú, los diferentes platillos y el costo total de los mismos.
- Sea rápido el proceso de pagos, y que el cliente pueda conocer el valor de su consumo al instante.

Existen trabajos relacionado con este trabajo de Integración Curricular como el de (Santillán Marfetan, y Gusñay Cela, 2020) en el que se desarrolla una aplicación web que facilita y agiliza los procesos e incremento de las ventas en los establecimientos de comida, permitiendo superar los problemas ocasionados por falta de herramientas tecnológicas. Este proyecto consta de módulos

como usuarios, productos, ventas, pedidos, proveedores, compras, turnos, movimientos de caja y reportes, y el aporte de la herramienta en la gestión y ventas de un restaurante. Otro trabajo en el que se encuentra similitudes es el de (MONTROYA, y SANCHEZ, 2020), el cual inspiró a realizar un trabajo similar a su aplicación móvil y sistema web desarrollado con la ayuda de metodologías y herramientas de desarrollo libres y gratuitas. Logrando de esta manera automatizar los procesos para la toma de pedidos y procesos administrativos como la gestión de empleados, menú, reporte de ventas, entre otros. Se ofreció de esta manera a los clientes una nueva experiencia digital con un menú interactivo, productos personalizables y formas de entretenimiento mientras esperaban por su orden.

En base a lo anteriormente relacionado es importante que el restaurante pueda brindar un servicio eficaz y disponible al momento de gestionar las reservas y pedidos, tendrá un módulo para reservas, otro módulo para pedidos el cual incluirá la opción de pago en línea a través de pasarela de pagos. La aplicación posteriormente recibirá el nombre de RESPEDAPP (Aplicación de Reservas y Pedidos); finalmente el sistema está dirigido tanto para el personal como para los clientes que cuenten con un dispositivo inteligente.

1.2 Formulación del Problema

¿Cómo mejorar la eficiencia y disponibilidad de la gestión de reservas y pedidos del restaurante “El fogón de COZ” mediante una aplicación web progresiva implementando SSR?

1.3 Sistematización del Problema

¿Cuáles son las características que brinda una aplicación web progresiva?

¿Cuáles son las características que brinda la implementación de SSR?

¿Cuál es el proceso actual para la gestión de reservas y pedidos en el restaurante “El fogón de COZ”?

¿Cuáles son los módulos que tendrá la aplicación web progresiva?

¿Cómo se va a evaluar la eficiencia y disponibilidad de la gestión de reservas y pedidos de la aplicación web progresiva?

1.4 Justificación del Trabajo De Integración Curricular

1.4.1 Justificación Teórica

En la actualidad, son pocos los negocios que no cuentan con una página web que muestre sus productos además de una aplicación de escritorio para llevar el control de reservas y pedidos. Además, en la actualidad se lucha día a día para combatir las secuelas del COVID-19 a través de un cambio tecnológico y automatización de procesos. Las aplicaciones web a diferencia de las páginas web o las aplicaciones de escritorio, pueden brindar un vínculo mayor con los clientes. Los clientes pueden interactuar con la aplicación web, ver la información y además formar parte del ciclo de vida de su pedido o reserva. Los negocios apoyan la automatización de sus procesos, pues se mejora el tratamiento de sus productos.

Una aplicación web progresiva permite combinar lo mejor de las aplicaciones nativas: presencia y funcionalidad; y lo mejor de la web: eficiencia, adaptabilidad y accesibilidad. Ofrece al usuario una experiencia en pantalla completa, de forma rápida, sin necesidad de conexión, y sin la instalación (Frías, 2019). No se debe confundir los términos, una aplicación web progresiva sigue siendo una aplicación web, no se convierte mágicamente en una aplicación de escritorio o móvil.

Algunas ventajas que brinda el tener una PWA (aplicación web progresiva) en un negocio son: brindar notificaciones al usuario, ser responsive, ser utilizadas sin conexión a internet, tener mayor velocidad de carga, ser actualizadas automáticamente sin molestar al cliente, no estar alojadas en una tienda de aplicaciones, ocupar poco o nulo espacio en el dispositivo, entre otras ventajas. Se va a utilizar el renderizado del lado del servidor (SSR), que es la capacidad que tiene una aplicación para ayudar a mostrar la página web del lado del servidor en lugar de renderizarla en el lado del cliente (Nuxt, 2022). Los beneficios de SSR son una carga inicial más rápida y la mejora en el posicionamiento en gestores de búsqueda como lo es Google.

1.4.2 Justificación Aplicativa

Con RespedApp se automatizará la gestión de los pedidos donde los clientes podrán visualizar el menú de platillos y seleccionar los que desee para luego realizar su pedido, además existirá dos clases de pedidos los cuales son: dentro del local y a domicilio. Además de realizar reservas que también pueden incluir un pedido.

La estrategia de negocio más común de llevar actualmente es “negocio centrado en el cliente”, por este motivo se debe ser muy asertivo al momento de brindarles un servicio, y que mejor manera de hacerlo evitándole largas colas y haciéndole parte del proceso para la obtención de su alimento.

Para las reservaciones el cliente podrá ingresar al apartado de reservaciones y realizar su reservación para lo cual deberá ingresar los datos correspondientes, seleccionar la hora y se generará un número de reservación. Para el lado administrativo se podrá visualizar los pedidos que se van generando en tiempo real, las reservaciones existentes, así mismo administrar su menú. RespedApp, además, de ofrecer la automatización, también tiene un compromiso con el medioambiente que ofrece reducir el consumo de papel, en aspectos de administración de pedidos, reservaciones y principalmente el menú, ya que a menudo se sacan nuevos platillos, o estos se desgastan por la manipulación y esto implica volver a imprimir las cartillas del menú constantemente.

A continuación, en la **Tabla 1 - 1** se detallará los módulos y secciones con los que contará la aplicación RESPEDAPP.

Tabla 1-1: Módulos que integran la aplicación web progresiva RESPEDAPP con sus funcionalidades respectivas.

Módulo	Descripción
Autenticación de clientes	En este módulo los clientes podrán iniciar sesión en el caso de tener una cuenta o registrarse en el caso de no tener una cuenta, además, se podrá ingresar a modo de Invitado.
Autenticación administrativa	En este módulo se controlará el ingreso para el uso exclusivo de los administradores del restaurante
Gestión de Menú	En este módulo el administrador podrá listar, agregar, editar y eliminar los ítems del menú
Gestión de Usuarios	En este módulo el administrador podrá listar, agregar, editar y eliminar los usuarios (excepto cliente que solo se eliminan)
Gestión de Pedidos	En este módulo el administrador podrá visualizar los pedidos realizados por los clientes, además, podrá agregar, editar, listar y eliminar.
Visor de pedidos	En este módulo el administrador podrá visualizar los pedidos realizados por los clientes en tiempo real, además, de que podrá ir marcando el estado del pedido (recibido, en preparación, finalizado).
Gestión de Reservaciones	En este módulo el administrador podrá visualizar las reservaciones realizadas por fechas, además, podrá agregar, editar, listar y eliminar.
Pagos	Para este módulo se consumirá la API de PayPal para el procesamiento de pagos

Módulo	Descripción
Pedidos	En este módulo los clientes podrán realizar pedidos del menú ya sea para pedir a domicilio o cuando estén dentro del establecimiento
Reservaciones	En este módulo los clientes podrán realizar su reservación donde deberán proporcionar los datos necesarios para la reservación
Inicio	En este módulo los clientes podrán visualizar información principal del establecimiento
Menú	En este módulo los clientes podrán visualizar todos los platillos que el administrador haya registrado para el menú

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

El presente trabajo se encuentra bajo la línea de investigación de Tecnologías de la Información y Comunicación (TIC's) en el programa de ingeniería de software en el ámbito de Análisis y Diseño de Software. En cuanto al Plan Nacional de Desarrollo corresponde al eje Social, basado en el objetivo 5: “Proteger a las familias, garantizar sus derechos y servicios, erradicar la pobreza y promover la inclusión social” según la POLÍTICA: 5.5 Mejorar la conectividad digital y el acceso a nuevas tecnologías de la población Pol 5.4. A4. Fortalecer la conectividad y el acceso a las TIC como una vía para mejorar el acceso a otros servicios.

1.5 Objetivos

1.5.1 Objetivo General

Desarrollar una aplicación web progresiva para la gestión de reserva y pedidos del restaurante “El fogón de COZ” implementando SSR.

1.5.2 Objetivos Específicos

- Analizar las características del desarrollo de una aplicación web progresiva con SSR.
- Determinar el proceso actual para la gestión de reservas y pedidos en el restaurante “El fogón de COZ.
- Desarrollar los módulos de pedidos, reservación, menú, autenticación, pagos e inicio que formarán parte de la aplicación web progresiva. mediante la implementación de *server side rendering*
- Evaluar la eficiencia y disponibilidad que tiene la gestión de reservas y pedidos de la

aplicación web progresiva a través del modelo ISO/IEC 25000.

CAPITULO II

2 FUNDAMENTOS TEÓRICOS

En el presente capítulo, se da a conocer los conceptos básicos y fundamentales del desarrollo de este proyecto de integración curricular.

Para comenzar se realizará una investigación bibliográfica acerca de los de sistemas de gestión de restaurantes con el objetivo de conocer los tipos de sistemas que existen y las características de estos, de igual forma se recolectará información acerca de los tipos de aplicaciones web enfatizando las aplicaciones web progresivas, analizando sus ventajas, características, composición, etc. Además, se visualizará los tipos de técnicas de renderizado que se manejan en la actualidad, donde principalmente se centrará la atención en la técnica *server side rendering* (SSR).

Se analizará la norma ISO/IEC 25000 que servirá de referencia para medir las métricas de calidad, y finalmente se integra una serie de trabajos relacionados con el tema planteado, lo que permitirá utilizarlos de guía durante el desarrollo de la investigación e incluso como precedente durante la fase de análisis de resultados.

2.1 Proceso de Sistema de Reservas

El proceso de reservas en un restaurante comúnmente comienza desde la solicitud del cliente, se evalúa la disponibilidad de espacio, se decide si se adjunta un pedido de comida o si la gestión del pedido se desea realizar luego, se confirma o se cancela la reservación y se finaliza la reserva cuando el tiempo de esta ha concluido.

2.2 Proceso de Sistema de Pedidos

El proceso de pedidos en un restaurante comúnmente comienza desde la exposición del menú al cliente, se espera a que este escoja los productos que desea consumir, se anota este pedido en algún lugar físico, se pasa esta información a cocina en orden en que los pedidos son receptados, se elabora los productos, se entrega el producto al cliente, el pago del consumo se puede pagar previo o posterior a la entrega del producto y finalmente se marca como completado al pedido.

2.3 Sistema Informático de Reservas

Un sistema de reservas informático es una plataforma que permite que los clientes puedan comunicarse de forma eficiente con el restaurante. El cliente puede acceder y consultar la disponibilidad del restaurante o establecimiento. Posteriormente, queda confirmada la fecha y

hora de su reserva (Bartalent Lab, 2021). También se lo conoce con las siglas CRS, por su acrónimo en inglés de *Central Reservation System* o *Computer Reservation System*. Originalmente los CRS fueron diseñados para realizar reservas de aerolíneas, es decir comprar tickets de avión, vuelos, transportes, entre otras cosas. Sin embargo, en los últimos años, prácticamente todo sitio comercial permite realizar reservas para su establecimiento (The Economist, 2012).

2.3.1 Características de un sistema de reservas informático

Las características de un CRS son:

- Deben estar accesibles para el cliente durante todo el día
- La disponibilidad del lugar debe mantenerse actualizada en tiempo real
- Ayudan a mejorar la organización del negocio

2.4 Sistema Informático de Pedidos

Los sistemas de pedidos informáticos permiten a los clientes realizar un pedido desde cualquier dispositivo inteligente, el cual alerta inmediatamente a la cocina cuando el pedido es solicitado. Los clientes tienen la posibilidad de elegir como desean gestionar su pedido ya sea que se les entregue en su domicilio, esté listo para recoger o para servirse en el establecimiento (Lavu, 2020).

La importancia de manejar sistemas de pedidos informáticos asegura la exactitud del pedido realizado ya que le permite al cliente verificar y darle seguimiento a lo que el solicitó en tiempo real, además, que ayuda agilizar el proceso de solicitud y pago por su pedido no solo para el cliente sino también para personal del restaurante que puede centrarse en desempeñar sus roles y ya no de revisar o contestar el teléfono para verificar si existen ordenes nuevas.

De igual forma en el artículo de Lavu (2020) clasifica a los sistemas de pedidos informáticos en dos tipos donde se define cada una de sus características y el funcionamiento que realiza cada uno de los modelos presentados.

2.4.1 Sistemas de pedidos internos

Los sistemas de pedidos internos por lo general cuentan con un *mPdV* que permite manejar los pedidos en la mesa, por lo que ayuda a que los servidores operen de manera eficiente, o permite que los clientes puedan realizar sus pedidos directamente a través de dispositivos proporcionados por el restaurante, sean una tableta que este en la mesa o un *smartphone* que tengan un software

preinstalado que le permita cumplir con este servicio, si bien uno de los principios de los sistemas de pedidos internos es reducir la interacción entre cliente y personal, donde de igual forma también ayuda a aumentar la satisfacción del cliente, dándole el control en todo momento a la hora de realizar su pedido. (Lavu, 2020).

2.4.2 *Sistemas de pedidos en línea*

Los sistemas de pedidos en línea o también denominados como externos son sistemas que le dan una mayor libertad a los clientes a la hora de realizar un pedido, ya que ellos pueden elegir el restaurante que desean y la forma en la que quieren que se despache su pedido ya sea para recoger, servirse o que se lo entreguen en su domicilio.

De los sistemas de pedidos en línea se puede decir que existen de dos tipos: los sistemas de pedidos internos propios y los que son de terceros tal como se lo puede observar en la Figura 1-2, donde los restaurantes deben regirse a las restricciones que les imponga este servicio de terceros, entre los más conocidos Uber Eats, Pedidos Ya, Rappi. Las restricciones más comunes que manejan estos servicios son los honorarios de comisión que varía el porcentaje dependiendo la empresa a la que se asocie, la falta de control ya que cuando usas un servicio de tercero significa que tu restaurante maneja tu nombre por separado y puede no representar lo que realmente es tu marca (Lavu, 2020).

2.5 *Aplicaciones Web*

Según (Luján-Mora, 2002), las aplicaciones Web son aquellas herramientas donde los usuarios pueden acceder a un servidor Web a través de la red mediante un navegador determinado. Por lo tanto, se define como una aplicación que se accede mediante la web por una red ya sea intranet o Internet. Por lo general se menciona como aplicación web a todo aquel programa informático que es ejecutado a través del navegador.

En la carrera de Software se puede definir a una aplicación web como aquella aplicación en la cual los usuarios pueden acceder al contenido de un servidor web a través de Internet utilizando el navegador de su preferencia, puede ser a través de un dispositivo móvil y/o un computador (Yumi Guamán, y Moina Zabala, 2016, p. 7).

2.5.1 *Características de las aplicaciones web*

A continuación, se puede visualizar las características de las aplicaciones web:

- Tienen un diseño único y son interactivas
- No requieren de instalación ni ocupan espacio de memoria
- Deben ser compatibles con cualquier navegador
- Se acceden a ellas a través de buscadores de internet y/o su URL

2.5.2 *Clasificación de las aplicaciones web*

2.5.2.1 *Aplicación web estática*

Las aplicaciones web estáticas son un tipo de páginas que están destinada a mostrar información de carácter informativo, además, de que también está pensada para no generar nuevo contenido es decir que su información no sea cambiada. En el caso de que se desee realizar nuevos cambios en la información de la aplicación web deben contactarse con el administrador de su sitio web o desarrollador que la realizo para que integre estos nuevos cambios. Estos cambios solo los podrá realizar personas con conocimiento básico de la estructura de una página web debido a que suelen estar desarrolladas en HTML, CSS y en algunos casos JavaScript, así mismo también se componen de otros elementos como imágenes, videos, GIFS, entre otros elementos (Jesuïtes Educació, 2017).

2.5.2.2 *Aplicación web dinámica.*

Estas aplicaciones a diferencia de las estáticas como lo mencionan su nombre son aplicaciones que su información es cargada de forma dinámica, usualmente la información de aplicación viene de una base de datos o fichero, así mismo su desarrollo son mucho más complejas de construir así mismo por otra parte facilita al cliente la modificación de la información de su aplicación web, esto usualmente se lo realiza a través de un panel de administración que suele contar con una interfaz fácil de usar para que el cliente pueda usarla, estos paneles de administración suelen denominarse CMS o también se conocen como Gestor de contenidos. Los gestores de contenidos más conocidos y de igual forma los más usados son WordPress, Joomla, Drupal, se dice que casi el 80% de todas las páginas que existen en la internet están desarrolladas con algunos de estos CMS.

Los lenguajes en los que se suelen desarrollar este tipo de aplicaciones son por ejemplo PHP, ASP, JAVA, NODEJS entre otros lenguajes, pero los mencionados son los más populares especial mente PHP ya que los CMS más usados están desarrollados en este lenguaje (Jesuïtes Educació, 2017).

2.6 Aplicaciones Web Progresivas

Una aplicación web progresiva es una aplicación web que utiliza las últimas tecnologías disponibles en los navegadores para ofrecer una experiencia lo más parecida posible a la de una aplicación nativa (Rodríguez, et al., 2019).

Con el fin de reducir la brecha que existe entre las aplicaciones nativas de las aplicaciones web surge el enfoque a lo que se conoce hoy en día como Aplicaciones Web Progresivas (PWA por sus siglas en inglés). Como es de imaginarse las PWA son una evolución de las aplicaciones web ordinarias que buscan reducir la barrera que existen con respecto a las aplicaciones nativas, debido que las PWA hacen uso de las tecnologías que ofrecen los navegadores con el fin de parecer cada vez más a una aplicación nativa. Al mencionar tecnologías del navegador, se hace referencia a las APIs que ofrecen los navegadores para acceder a recursos de hardware de un dispositivo móvil como por ejemplo la geolocalización, notificaciones, giroscopio, cámara, micrófono, vibrador entre otros recursos. Las PWA se desarrollaron con el fin de darle al usuario una experiencia muy similar de una aplicación nativa, pero en este caso para la web, buscando mejorar el rendimiento en los dispositivos incluso haciendo que la aplicación cargue de forma instantánea con una interfaz similar a una nativa donde los usuarios puedan recibir notificaciones y de igual forma darle la capacidad de por usar la aplicación sin conectividad a internet (Rodríguez, et al., 2019).

2.6.1 Características

Algunas características que una PWA puede ofrecer según Carneiro (2018) son:

- Accesibilidad a los usuarios, sin importar el navegador que usen
- Adaptabilidad a la pantalla del dispositivo para que tenga una vista *responsive*
- Independencia de Internet cuando se tenga conectividad limitada o nula
- Similitud a una aplicación nativa
- Actualizaciones de contenido a través de *service workers*
- Seguridad aplicando el protocolo HTTPS
- Permitir almacenar un acceso a la aplicación en algún dispositivo

2.6.2 Componentes

Para que una aplicación web pueda llamarse aplicación web progresiva o PWA y pueda instalarse en un teléfono móvil o escritorio, primero debe cumplir con los siguientes componentes o requisitos, ya que sin estos la PWA no podrá ejecutarse (Luna, 2021b).

2.6.3 *Web App Manifest*

El archivo manifiesto o como también se lo conoce como archivo Manifest. Este archivo sencillo cuya estructura tiene formato JSON es decir se maneja en clave – valor. Este archivo agrupa principalmente la descripción de la aplicación, nombre, tipo de licencia, iconos entre otras características, las cuales se mostrarán a continuación:

- *name*: nombre corto de la aplicación
- *description*: descripción general de lo que hace la aplicación
- *icons*: array de iconos con distintas resoluciones
- *start_url*: URL de inicio cuando se abra la aplicación
- *display*: Define el modo de visualización (*standalone*, *fullscreen*, *minimal-ui* o *browser*)
- *Lang*: Especifica el idioma principal de la aplicación
- *orientation*: Define la orientación por defecto (*landscape*, *natural*, *portrait*, etc)
- *theme_color*: Define el color por defecto para la aplicación
- *background_color*: Define el color de fondo que se mostrara cuando cargue la aplicación (Luna, 2021a)

2.6.4 *Service Worker*

El *service worker* actúa de una forma parecida un proxy debido a que está operando entre la aplicación web y el servidor al cual se le envía las peticiones. El *service worker* siempre está ejecutándose a en segundo plano, separado de una página web, esta es encargada de permitir la creación de experiencias *offline*, debido a que si la red es inestable esta las registra en el almacenamiento local y las envía cuando ya la conexión este bien. Además, en la actualidad, también se incorporan funciones como notificaciones *push* y sincronización en segundo plano (MDN Contributors, 2022).

2.6.4.1 *El ciclo de vida del Service Worker*

El ciclo de vida del *service worker* es totalmente independiente al de la aplicación web, puesto que al instalarse la PWA este proceso se lo realiza en un archivo JavaScript en segundo plano al momento de realizar la instalación. Durante la etapa de instalación se comenzará a almacenar en cache todos los elementos estáticos necesarios para el funcionamiento de la aplicación de manera *offline*, recursos comunes como imágenes, videos, estilos, páginas HTML entre otros elementos, este paso se cumple de manera exitosa el *service worker* se activará correctamente es decir primero se guarda en cache y luego se activa, en el caso de que haya un error en el almacenamiento de la cache lanza un error y no se activará el *service worker* y la instalación fallará y volverá a

intentar el proceso la siguiente vez como se puede visualizar en la **Figura 1-2**. Una vez terminada la instalación y la activación del *service worker*, este será el encargado de controlar las páginas a las que se tiene acceso las cuales se configuraron en el archivo de manifiesto (Gaunt, 2019).

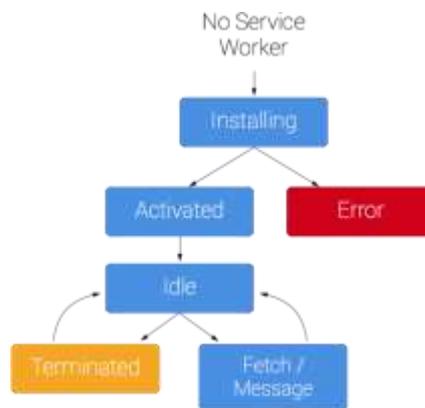


Figura 1-2: Ciclo de vida del *service worker* de una aplicación web progresiva

Fuente: (Gaunt, 2019)

2.6.4.2 Conexión HTTPS

Utilizar un certificado de *Secure Sockets Layer* (SSL) que le permita usar el protocolo HTTPS a una página no solo es una buena práctica en la actualidad, sino que también significa confiabilidad y seguridad para los usuarios que van a usar la aplicación porque representa que el manejo de sus datos e información se lo realizara a través de una conexión segura y que por ende estarán encriptadas (MDN Contributors, 2021). Pero porque es necesario HTTPS para la implementación de PWA, pues esto se debe a que la mayoría de las funciones como la geolocalización, giroscopio, incluso los servicios worker que son encargados de manejar, crean y filtran respuestas que se envíen, es por ello que esto solamente está disponible cuando la aplicación es cargada mediante el protocolo seguro HTTPS (Gaunt, 2019).

2.6.4.3 Ventajas

Las ventajas de desarrollar una aplicación web progresiva según Tandem (2022) son:

- Son multiplataforma y se puede acceder desde cualquier navegador y sistema operativo
- Adaptan su funcionalidad al navegador utilizado por el cliente
- Notifican a los usuarios a través de notificaciones *push*
- No se necesita descargarlos, por lo que no ocupan espacio en su teléfono
- Gracias a la tecnología *Service Worker*, se pueden utilizar sin conexión a Internet
- Su velocidad de carga es similar a la de las aplicaciones nativas y superior a la de las aplicaciones web

- Su navegación es sencilla, intuitiva y adictiva, como la *app* nativa
- Las actualizaciones se realizan de forma automática sin inconvenientes para el usuario
- Utilizan el protocolo *Transport Layer Security* (TLS)

2.7 Técnicas de Renderización en la Web

Internet ha cambiado de una manera drástica desde un conjunto de páginas únicamente estáticas a un contenido dinámico que permite una interacción del usuario enriquecida (Beke, 2018).

A partir de este cambio, las organizaciones luchan por decidir qué estrategia de renderizado deben elegir para sus aplicaciones. Renderizado es el proceso de convertir el código en una representación HTML para mostrar la interfaz de usuario. Esta representación puede tener lugar en el servidor o en el cliente. Puede ocurrir antes de tiempo en el momento de la compilación o en cada solicitud en el tiempo de ejecución (Next, 2022). Twitter, por ejemplo, originalmente usaba renderizado del lado del servidor (SSR), luego cambió al renderizado del lado del cliente (CSR) y finalmente decidió volver a SSR (Beke, 2018).

2.7.1 Client Side Rendering

A través de esta técnica el servidor deja de ser responsable de la lógica de la página y se posee aplicaciones que se procesan en el cliente por completo. El cliente recibe inicialmente una página HTML que contiene etiquetas de datos que tendrá que resolver solo, contrario a las aplicaciones web con SSR donde el cliente carga el HTML ya renderizado (Schneider, 2016, p. 21). El backend expone sus datos a través de una API. El cliente obtiene los datos y los representa como se visualiza en la **Figura 2-2**. Esto significa que el servidor envía el código de la aplicación web con elementos mínimos y código con toda la lógica al cliente para que este renderice la información, principalmente mediante el uso de frameworks modernos.

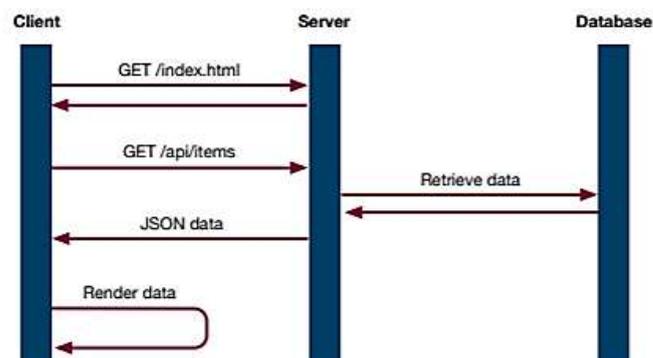


Figura 2-2: Secuencia o flujo del proceso de renderizado por lado del cliente

Fuente: (Beke, 2018)

2.7.2 Server Side Rendering

En las aplicaciones web, la renderización del lado del servidor (SSR) se basa en una técnica en la que existe un cliente/navegador, que hace peticiones; y un servidor, que se encarga de analizar y procesar estas solicitudes cuyo proceso se puede observar en la **Figura 3-2**. A partir de ahí, el servidor responde con un archivo HTML que será interpretado por el navegador y presentado al usuario final. Por lo tanto, en modelos con SSR, el cliente es simplemente el agente que realiza solicitudes y recibe respuestas (Schneider, 2016, p. 17).

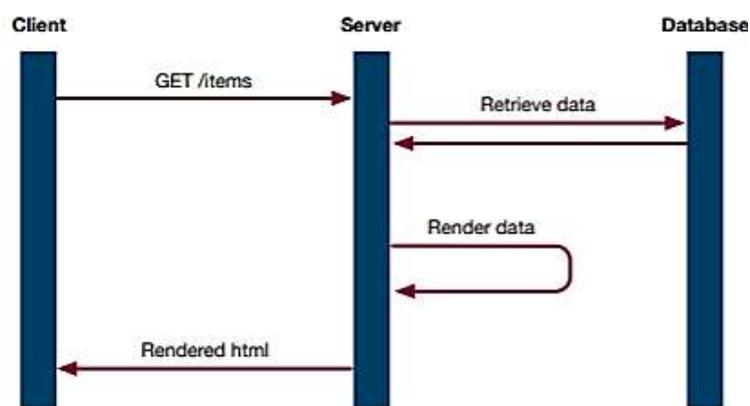


Figura 3-2: Secuencia o flujo del proceso de renderizado por lado del servidor

Fuente: (Beke, 2018)

El autor Bryant Alejandro (2020), define SSR como un conjunto de técnicas que son empleadas para poder realizar el renderizado de la información del lado del servidor para posteriormente entregarla al cliente y esta sea manejada como una página normal, pero con un potencial de rendimiento superior.

2.7.2.1 Características

Según Fadhilah Iskandar (2020) la técnica SSR brinda un mejor rendimiento de la aplicación con respecto a otras técnicas de renderizado. SSR viene con una serie de beneficios de rendimiento que mejoran la experiencia de usuario. Las páginas SSR tienen un tiempo de carga y una primera vista de contenido más rápidas, porque el contenido está disponible en el navegador antes. Lo que significa menos tiempo que su usuario tiene para mirar una pantalla de carga.

Google otorga una búsqueda preferencial a los sitios con la velocidad de carga de página más rápida. Los tiempos de carga más rápidos mejoran las métricas del usuario, como la duración de

la sesión y la tasa de rebote; SSR brinda las tres cosas que se desea de una aplicación web: rapidez, motores de búsqueda eficientes y usuarios felices (Prerender, 2021).

2.7.2.2 Ventajas

Las ventajas de utilizar la técnica de renderizado por parte del servidor son:

- Proporcionan una carga inicial rápida
- Mejora el desempeño referente a la optimización para motores de búsqueda (SEO)
- Funcionamiento correcto de los esquemas de metadatos (Bryant Alejandro, 2020).

¿Cómo implementarlo?

Los *frameworks* y librerías más comunes para el desarrollo web en la actualidad están entre React.js, Vue.js y Angular, para cada una de estas tecnologías existen otros *frameworks* que integran otras características que ayudan a la optimización de las tecnologías antes mencionadas, por ejemplo, para React.js existe Next.js y Gastby, para Vue.js se puede utilizar Nuxt para Angular existe Angular Universal, también hay otras formas de implementar *server side rendering* por mencionar a SvelteKit, Quasar Framework e incluso de forma nativa con ayuda de Express y Nodejs (Sakhi, 2021).

Para el desarrollo de este trabajo de integración curricular se ha optado por utilizar la tecnología de React.js debido a que es una tecnología desarrollada por Facebook, además, es la librería más popular entre los desarrolladores web para la creación de interfaces según lo dice el autor Patel (2021) en su artículo “*List of 10 Best Frontend Frameworks*” por lo tanto para la implementación de una PWA con SSR se optó por utilizar Next.js que es “El *framework* de React para producción”

Framework Next.js

Creado por Vercel, Next.js es un *framework* muy popular de código abierto que está basado en React.js y que se ejecuta sobre Node.js, con Next.js se puede realizar las mismas cosas que se realizan con React.js, la diferencia está en que en Next no necesita de una configuración inicial extensa, además de que cuenta con su propio sistema de administración de rutas, además, de que ofrece *Static Generation* y *Server side rendering* sin la necesidad de implementar una configuración compleja (Macoretta, 2021).

a) Características de Next.js

Next.js tiene todas las herramientas necesarias para un desarrollo web mucho más rápido y con

las mejores prácticas, algunas de sus principales características son:

- Configuración mínima
- Representación previa de páginas mediante SSG y SSR
- Optimización de imágenes
- Componentes dinámicos
- Compatibilidad con TypeScript
- Enrutamiento de sistemas de archivos
- Compatibilidad integrada con CSS

Todas las características fueron recuperadas de sitio oficial (Vercel, 2021)

Para la creación de una aplicación de Next.js basta con tener instalado el entorno Node.js y el gestor de paquetes NPM, dirigiéndonos a la consola o *Command Prompt* y escribir “*npx create-next-app [our-app-name]*” (Asperbrothers, 2021).

2.8 Calidad del Software

Según la Real Academia Española la calidad se define como una propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar el valor de un objeto. Sin embargo, determinar la calidad dependerá de los parámetros a evaluar, el objetivo, las personas que evalúen y los recursos que se tienen para determinar la calidad. Por lo tanto, se llega a concluir que el término calidad es relativo, no es subjetivo, ni objetivo y está sujeto a restricciones (Tello, 2016, p. 15). La calidad del software por otra parte se refiere a una concordancia entre requerimientos funcionales explícitamente establecidos y los estándares de desarrollo documentados con características implícitas que se esperan de todo software profesional, para lograr estas evaluaciones de la calidad, se debe utilizar medidas técnicas que evalúan la calidad con objetividad y no con subjetividad (DB Technologies, 2012).

2.8.1 Modelos de Calidad del Software

Para poder estandarizar las métricas de calidad del software se han logrado crear modelos y/o estándares que incluyan un conjunto de características, factores y criterios que guían a los ingenieros de software a seguir buenas prácticas de medición y lograr medir la calidad de sus productos. Existen modelos de calidad a nivel de proceso, producto o en uso. Para el presente trabajo de integración curricular se centrará en la calidad del producto de software. Los modelos de calidad de software a nivel de producto más populares según (Callejas-Cuervo, Alarcón-Aldana, y Álvarez-Carreño, 2017) son:

- Modelo Mc Call
- Modelo FURPS
- Modelo GQM o *Goal Question Metric*
- Modelo Boehm
- Modelo GILB
- ISO/IEC 9126
- Modelo SQAE o *Software Quality Assessment Exercise*
- Modelo WebQEM
- ISO/IEC 25000

Algunos de estos modelos de calidad han sido base para crear otros modelos más recientes. Este es el caso de la ISO/IEC 25000 que surge del modelo ISO/IEC 9126 pero agrega algunas características que han ayudado a determinar de mejor manera la calidad de un producto de software. Según el artículo de DB Technologies (2012), se llega a la conclusión de que la ISO/IEC 25000 o SQuaRE es el modelo que mejor se centra en el producto y es la más completa con respecto a otros modelos por ello se ha decidido aplicar este modelo al presente trabajo de integración curricular.

2.8.1.1 Norma ISO/IEC 25000

La ISO/IEC 25000 tiene por objetivo la creación de un marco de trabajo para evaluar la calidad de un producto de software, de tal forma que se establezca algunos criterios para especificar los requisitos de calidad, sus métricas y su evaluación (ISO 25000, 2022).

La norma ISO/IEC 25000 esta dividida en cinco partes según Balseca Chisaguano (2014):

- a) **2500n: Gestión de la Calidad:** La ISO 25000 es una guía conocida también por el nombre de sistemas y requisitos de evaluación y calidad del software (SQuaRE). Su propósito es proporcionar una visión general del contenido, modelos de referencia y ciertas definiciones generales y relaciones entre documentos (Crespo, 2018).
- b) **2501n: Modelo de Calidad:** Estos estándares internacionales incluyen modelos detallados del sistema informático y la calidad del software, el uso y la calidad de los datos. una guía práctica para usar modelos de calidad (Instituto Ecuatoriano de Normalización, 2011, p. 6).
- c) **2502n: Medición de la Calidad:** Los estándares internacionales incluyen un modelo de

referencia para medir la calidad del producto, una guía práctica de software, definiciones matemáticas de medición de calidad y sus aplicaciones. Proporciona medidas internas y externas de la calidad del software y ejemplos del uso de métricas de calidad (Instituto Ecuatoriano de Normalización, 2011, p. 6).

- d) **2503n: Requisitos de la Calidad:** Estos estándares internacionales ayudan a definir los requisitos de calidad basados en modelos de calidad y mediciones de calidad. Estos requisitos de calidad se pueden utilizar durante la recopilación de requisitos de calidad para software en desarrollo o como entrada para el proceso de evaluación (Instituto Ecuatoriano de Normalización, 2011, p. 7).
- e) **2504n: Evaluación de Calidad:** Estos estándares internacionales incluyen requisitos, recomendaciones y pautas para revisores, compradores o desarrolladores de software. También se ha introducido la compatibilidad con el registro de métricas como módulo de evaluación (Instituto Ecuatoriano de Normalización, 2011, p. 7).

El modelo de Calidad ISO/IEC 25010 define las características y subcaracterísticas que es la parte de la norma que servirá para determinar la calidad del producto de software. Mientras que la ponderación del modelo ISO 2504 servirá para estandarizar los puntajes de calidad obtenidos.

2.8.2 Características y Subcaracterísticas del modelo de calidad 25010

Existen 8 características y 30 subcaracterísticas definidas para este modelo de calidad. A continuación, se detallará a que se refiere cada una de ellas y cuales se evaluará en el presente proyecto:

Adecuación Funcional: Representa la capacidad del software para proporcionar una funcionalidad que satisfaga las necesidades explícitas e implícitas (Cahueñas, 2017). Sus subcaracterísticas son:

- Completitud Funcional
- Exactitud Funcional

Facilidad de Uso: Es la capacidad del producto de software de ofrecer a los usuarios la facilidad de ser entendido, aprendido, usado y agradable a la vista cuando sea usado por el usuario bajo ciertas condiciones determinadas (Cahueñas, 2017). Sus subcaracterísticas son:

- Capacidad para reconocer su adecuación
- Capacidad de ser entendido

- Operatividad
- Protección contra errores de usuario.
- Estética de la interfaz de usuario
- Accesibilidad técnica

Seguridad: La capacidad de proteger la información y los datos para que las personas o los sistemas no autorizados no puedan leerlos ni modificarlos (Cahueñas, 2017). Sus subcaracterísticas son:

- Confidencialidad
- Integridad
- No Repudio
- Responsabilidad
- Autenticidad

Compatibilidad: La capacidad de dos o más sistemas o componentes para intercambiar información y/o realizar funciones deseadas mientras comparten el mismo entorno de hardware o software (Cahueñas, 2017). Sus subcaracterísticas son:

- Coexistencia
- Interoperabilidad

Mantenibilidad: Esta propiedad representa la capacidad del software para modificarse de manera eficiente y efectiva debido a la necesidad de desarrollo, reparación o refinamiento (Cahueñas, 2017). Sus subcaracterísticas son:

- Modularidad
- Reusabilidad
- Capacidad de ser analizado
- Capacidad de modificación
- Capacidad de ser probado

Portabilidad: La capacidad de transferir de manera eficiente y eficaz un producto o componente de un hardware, software, entorno operativo o uso a otro hardware, software, entorno operativo o uso (Cahueñas, 2017). Sus subcaracterísticas son:

- Adaptabilidad
- Facilidad de instalación
- Capacidad de ser reemplazado

Eficiencia en el Desempeño: Esta función representa el rendimiento en relación con la cantidad de recursos utilizados en determinadas condiciones (Cahueñas, 2017). Sus subcaracterísticas son:

- Comportamiento Temporal
- Utilización de Recursos
- Capacidad

Fiabilidad: La capacidad de un sistema o componente para realizar una función específica cuando se usa en condiciones específicas y durante un período de tiempo específico (Cahueñas, 2017). Sus subcaracterísticas son:

- Madurez
- Disponibilidad
- Tolerancia a Fallos
- Recuperabilidad

2.8.3 Eficiencia

Eficiencia o Eficiencia en el desempeño como se conoce a través de la norma ISO/IEC 25000, se refiere a la capacidad que tiene el software para brindar un alto desempeño según los recursos utilizados y las condiciones establecidas en diferentes lapsos de tiempo.

2.8.4 Disponibilidad

La disponibilidad es la capacidad que tiene un servicio o un sistema de ser accesible y poder ser utilizado por los usuarios cuando ellos lo requieran. Hay que considerar que no será posible garantizar el sistema al cien por ciento por diversos factores, siempre va a depender del entorno donde estará el sistema, usualmente el sistema puede fallar por ataques cibernéticos y problemas físicos.

2.9 Trabajos Relacionados

A continuación, se va a analizar el trabajo que han desarrollado algunos autores con temas relacionados al tema planteado, e incluso con relación a las tecnologías de desarrollo que se tiene planeado emplear durante el desarrollo del proyecto, sin más que acotar, a continuación, se visualizará los análisis de cada uno de los trabajos que se ha recolectado.

En el primer artículo del cual se ha indagado información buscando relación con el presente trabajo de integración curricular se puede visualizar una comparativa en la que se establecen

algunas diferencias entre lo que es el lado del cliente con el lado del servidor, estableciendo algunos resultados como la forma en la que las capas interactúan entre sí partiendo desde capas superiores hasta llegar con los datos recolectados a capas inferiores, además que al hablar de rendimiento por el apartado del servidor existe un 100% de rendimiento mientras que el rendimiento por parte del cliente tiene solamente un 1% de rendimiento (Fadhilah Iskandar, et al., 2020). Para medir el rendimiento en el proyecto actual se utilizó métricas como TFB (*time to first byte*), tiempo de carga del documento, número de recursos, entre otros. El autor indica que el tiempo promedio en que la aplicación debería renderizarse por parte del servidor es de aproximadamente 2.1 segundos siendo este un valor significativo y referencial para obtener el tiempo de respuesta en el presente trabajo de integración curricular.

El autor Farid Said Tahirshah (2019) realizó una comparativa entre una página web tradicional es decir común contra una aplicación web progresiva, donde el autor realiza un análisis tomando en cuenta aspectos de: compatibilidad, prerequisites, rendimiento, tiempos de carga inicial, análisis del CPU, latencia, espacio, entre otros. Lo que se extrae de este trabajo es el resultado del análisis en base al rendimiento con el que se ha desenvuelto las aplicaciones web progresivas a la hora de realizar el test, para el autor el rendimiento se basa en el tiempo total en que se presenta la primera vista, para nosotros este valor hace referencia al tiempo de respuesta para los pedidos y/o reservas. El autor indica que este valor promedio es de 1.157 segundos para una aplicación web progresiva y 1.205 segundos para una aplicación web normal, dando como resultado que las aplicaciones web progresivas son 4.1% más eficientes que las aplicaciones web normales. Por este y otros motivos se concluye que se tomó la decisión correcta al desarrollar una aplicación web progresiva para solucionar la problemática de baja eficiencia.

En el artículo de programador clic (2022) se indica que las aplicaciones web progresivas con SSR pueden trabajar sin conexión a internet utilizando los *service workers* que almacenan ciertas páginas en caché, esto permite que la aplicación tenga alta disponibilidad, sin embargo, el autor no brinda un valor preciso debido a que su disponibilidad dependerá de las condiciones a las cuales será expuesta la PWA. Por lo tanto, se debe entender que la disponibilidad de la aplicación web progresiva será relativa a las condiciones en las que se encuentre.

Para guiar este trabajo de titulación en referencia a la medición de métricas de calidad según la ISO 25000 se revisó el trabajo de Titulación de (Balseca Chisaguano, 2014) “Evaluación de calidad de productos de software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000” donde se obtiene las métricas del comportamiento temporal y disponibilidad además de una guía de ponderación para obtener una calificación de calidad del producto a desarrollar.

Para concluir con este capítulo de investigación y búsqueda bibliográfica, se ha podido determinar que la implementación de la técnica de renderizado SSR puede aumentar el rendimiento de una aplicación web a la hora de dibujar información en el navegador, además de que al transformar una aplicación web regular en una progresiva se le da la capacidad de ser instalada en los dispositivos y por ende se puede hacer uso de la aplicación sin conectividad a internet, así mismo como ya cuenta con todos los recursos descargados y almacenados en memoria los tiempos de carga de la página se reducen considerablemente según lo observado en los trabajos relacionados.

CAPITULO III

3 MARCO METODOLÓGICO

En el presente capítulo se detalla los métodos, técnicas y otros pasos que se emplean para lograr la solución al problema planteado previamente y cumplir los cuatro objetivos específicos. Se analiza la estrategia teórica y metodológica de los diferentes procesos investigativos. El primer objetivo se cumple a través de una investigación bibliográfica que permite analizar si las tecnologías empleadas son las más adecuadas para la solución. Para lograr el segundo objetivo se realiza una investigación mixta en conjunto a una entrevista para identificar los procesos que se llevan a cabo dentro del restaurante “El fogón de Coz”, y así proceder con la automatización de los mismos. En el objetivo tres se detalla las herramientas utilizadas para construir el producto de software y la metodología de software empleada es Scrumban. Finalmente, para el objetivo 4 se aplica una investigación analítica con el uso del modelo ISO/IEC 25000 para determinar las métricas de evaluación del presente Trabajo de Integración Curricular.

3.1 Tipo de Estudio

Durante el desarrollo del presente proyecto, este se ha ido adaptando a los siguientes tipos de estudio: descriptivo, aplicativo y cuasi experimental, los cuales se han ido aplicando conforme se presentan las fases del proyecto.

Se realizó un estudio de tipo descriptivo, ya que se necesitaba partir a describir los tipos de aplicaciones web que existen en la actualidad, de igual forma los tipos de técnicas de renderizados que se pueden implementar dentro de las aplicaciones web, además, este tipo de estudio ayudo a describir y entender que son las aplicaciones web progresivas y porque aplicarlas la técnica de renderizado *server side rendering*. Así mismo este estudio ayudo a describir el funcionamiento actual del proceso de reservas y pedidos los cuales son los que automatizará con la implementación de este proyecto.

En cambio, durante el desarrollo de los módulos de la aplicación se aplicó el estudio aplicativo, debido a que para esta fase se puso en práctica todos los conocimientos adquiridos durante las fases de investigación anteriores con el fin de implementar la automatización de los procesos de reservas y pedidos con los módulos de menú, autenticación, pagos e inicio, además así mismo este estudio permite adquirir nuevos conocimientos lo que permite desarrollar solidificar los conocimientos ya obtenidos, plasmándolos en la aplicación que se construye.

De igual forma para lograr de evaluar la eficiencia y la disponibilidad en la gestión de reservas y

pedidos de la aplicación web progresiva aplicando el modelo ISO/IEC 25000 se utilizó el estudio cuasi experimental, ya que se va a replicar el proceso de reservas y pedidos a través de una simulación controlada con ayuda de los instrumentos que se describen más adelante.

3.2 Métodos, Técnicas y Fuentes del Trabajo de Integración Curricular

A continuación, en la **tabla 2-3** se va a describir brevemente los métodos, técnicas, fuentes e instrumentos que se han utilizado en el presente trabajo de integración curricular por cada uno de los objetivos planteados y posteriormente se detallará las características y el resultado de su aplicación:

Tabla 2-3: Tabla de descripción del marco metodológico del presente trabajo de integración curricular

Objetivo	Métodos	Técnicas	Fuentes	Instrumentos
Analizar las características del desarrollo de una aplicación web progresiva con SSR	Deductivo	Formular preguntas Formular palabras claves	BibDigital DSpace Repositorio UNEMI Books Google IOPScience Páginas oficiales y blogs de expertos	Navegador Google Chrome 102.0.5005.63. Buscador de Google
Determinar el proceso actual para la gestión de reservas y pedidos en el restaurante “El fogón de COZ”	Analítico	Entrevista	Chef Alejandro Cabrera	Zoom Microsoft Word
Desarrollar los módulos de pedidos, reservación, menú, autenticación, pagos e inicio que formarán	Scrum	Modelamiento de datos Diagramas UML Normalización de base de datos GitFlow Notación CamelCase	Next.js El framework de React para producción – Curso de Udemy Documentación Oficial de Express, Express validator, Next, Material UI, Sequelize,	Herramientas Servicios en la nube IDE Gestor de base de datos Entorno en tiempo de ejecución

parte de la aplicación web progresiva. mediante la implementación de <i>server side rendering</i>			Bootstrap	Frameworks Librerías Paquetes Computadoras portátiles Servidores Web
Evaluar la eficiencia y disponibilidad que tiene la gestión de reservas y pedidos de la aplicación web progresiva a través del modelo ISO/IEC 25000	Analítico	<i>Profiling speed site</i> Prueba ping Prueba Shapiro-Wilk Prueba Wincoxon	Norma ISO/IEC 25000	Chrome DevTools Application Insights Azure R

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.2.1 Objetivo 1: Analizar las características del desarrollo de una aplicación web progresiva con SSR

3.2.1.1 Métodos

El método aplicado para cumplir con el objetivo planteado es la investigación es el deductivo, debido a que este método permite partir desde ¿Qué son las aplicaciones web y su clasificación? Luego centrarse en las aplicaciones web progresivas y nuevamente partir desde los tipos de técnicas de renderizado en la web, analizando específicamente la técnica server side rendering y como realizarla en una aplicación web progresiva.

3.2.1.2 Técnicas

Las técnicas empleadas en la investigación bibliográficas son las siguientes:

1. Formulación de preguntas concretas en lenguaje natural que delimiten el objeto de estudio.
 - ¿Qué son las aplicaciones web?

- ¿Qué son las aplicaciones web progresivas?
 - ¿Cuáles son las ventajas y desventajas de una aplicación web progresiva?
 - ¿Cómo es el funcionamiento una aplicación web progresiva?
 - ¿Qué son las técnicas de renderizado?
 - ¿Qué es el *server side rendering*?
 - ¿Cuáles son las ventajas y desventajas de aplicar *server side rendering*?
2. Formulación de palabras claves que permitan realizar una búsqueda más concreta.
- Ingeniería de software
 - Aplicación web progresiva
 - Renderización del lado del servidor
 - Software de gestión de restaurantes
 - Modelo de calidad ISO/IEC 25000

3.2.1.3 Fuentes

Para el presente Trabajo de Integración Curricular se ha optado por utilizar bibliotecas y repositorios digitales con artículos científicos y tesis publicadas a través de Internet. Toda la información recopilada debe ser lo más actual posible, debido a que la tecnología siempre está en constante actualización. Entre las bibliotecas más destacadas están:

- BibDigital: El Repositorio Digital Institucional de la Escuela Politécnica Nacional
- DSpace: El Repositorio Digital Institucional de la Escuela Superior Politécnica de Chimborazo
- Repositorio UNEMI: El Repositorio Digital Institucional de la Universidad Estatal de Milagro
- Books Google: Plataforma digital diseñada para almacenar libros de manera digital
- IOPScience: Contenido de las revistas publicadas por IOP *Publishing*.

Para definiciones técnicas se tomó la información de las páginas oficiales y blogs de expertos para cada una de las tecnologías utilizadas.

3.2.1.4 Instrumentos

Para realizar la búsqueda bibliográfica se utilizó el navegador Google Chrome en la versión 102.0.5005.63. Además, se utilizó el buscador de Google entre las fechas: 07/03/2022 al 18/03/2022.

3.2.2 Objetivo 2: Determinar el proceso actual para la gestión de reservas y pedidos en el restaurante “El fogón de COZ”

3.2.2.1 Métodos

Se ha optado por utilizar el método analítico para determinar los procesos de reservas y pedidos debido a que se partirá del flujo organizacional del restaurante, se toma los procesos enfocados y se analiza cada una de sus características junto a los actores de estas.

3.2.2.2 Técnicas

El autor Díaz-Bravo (2013) define la entrevista como: “Una técnica de gran utilidad en la investigación cualitativa para recabar datos; se define como una conversación que se propone un fin determinado distinto al simple hecho de conversar”.

Para la elaboración de la entrevista, se ha elaborado una serie de 5 preguntas de tipo abiertas, debido al escaso conocimiento del entrevistador acerca de los procesos internos del restaurante. Para conocer más detalles de la entrevista realizada observar el Anexo A.

3.2.2.3 Fuentes

Para poder cumplir con el objetivo de determinar el proceso actual de la gestión de reservas y pedidos del restaurante “El fogón de Coz” se consideró como la fuente más confiable de información al Chef Alejandro Cabrera, propietario y gerente del establecimiento; esta persona está al tanto y conoce cada una de las actividades que se llevan a cabo en el restaurante y forma parte de los *skateholders* del proyecto.

3.2.2.4 Instrumentos

Los instrumentos para realizar la entrevista con el propietario del establecimiento fueron:

- **Zoom:** Es una herramienta colaborativa que facilita la comunicación sincrónica entre los investigadores y el entrevistado.
- **Microsoft Word:** Este instrumento permite procesar texto que ayude a registrar las preguntas y respuestas obtenidas en la entrevista ejecutada.

3.2.3 **Objetivo 3: Desarrollar los módulos de pedidos, reservación, menú, autenticación, pagos e inicio que formarán parte de la aplicación web progresiva. mediante la implementación de server side rendering**

3.2.3.1 *Métodos*

La metodología aplicada para realizar el producto de software es Scrumban. Esta permite realizar la gestión de proyectos combinando dos estrategias ágiles comunes: Scrum y Kanban. Se optó por utilizar Scrumban debido a que funciona mejor en equipos que ya conocen algunas de las facetas de Scrum o de Kanban. Además, es útil porque aporta controles graduales y como se trabaja en *sprints*, se puede supervisar si el trabajo sigue avanzando. También consta de más herramientas que otras metodologías por lo que se obtiene un estado más real del proyecto y una mejor productividad.

3.2.3.2 *Técnicas*

Con la finalidad de cumplir con el objetivo de construir una aplicación web progresiva con la técnica de renderizado *server side rendering* se hizo uso de algunas técnicas de desarrollo, destacando las más importante a continuación:

- **Normalización de base de datos:** Es un proceso que permite aplicar una serie de reglas a las relaciones obtenidas entre entidades de una base de datos. Existen cinco formas normales, pero debido al tamaño de la base de datos y la complejidad de sus relaciones para el presente proyecto se ha aplicado hasta la tercera forma normal.
- **GitFlow:** Es una técnica, estrategia y también flujo de trabajo muy utilizado por los equipos de desarrollo y programación. Se destaca por ayudar en la organización de la versión de un código. Para el presente proyecto se utilizó ramas principales: *main* y *develop*, y ramas de soporte: *feature*, *release* y *hotfix*.
- **Notación CamelCase:** Es una convención de nomenclatura en la que la primera letra de cada palabra en una palabra compuesta está en mayúscula, excepto la primera palabra. Esta convención de nomenclatura hace que los nombres compuestos sean más legibles.
- **JSON Web Token:** es un estándar dentro del RFC 7519 y define un mecanismo para poder propagar de forma segura, la identidad de un usuario con una serie de privilegios (López Magaña, 2020) . En el proyecto se ha utilizado para brindar seguridad entre la comunicación

de APIs con las vistas.

3.2.3.3 Fuentes

Durante el desarrollo del proyecto se ha empleado las documentaciones oficiales de cada tecnología respectiva que se está utilizando, además, también se complementó con videos del autor Fernando Herrera instructor de la plataforma de Udemy, a continuación, se lista las fuentes y sus respectivos enlaces a la documentación.

- **Express.js** <https://expressjs.com/en/4x/api.html>
- **Express validator** <https://express-validator.github.io/docs/>
- **Next.js** <https://nextjs.org/docs/getting-started>
- **Material UI** <https://mui.com/material-ui/getting-started/overview/>
- **Sequelize** <https://sequelize.org/docs/v6/>
- **Bootstrap** <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
- **Next.js El framework de React para producción** <https://www.udemy.com/course/nextjs-fh/>

3.2.3.4 Instrumentos

Los instrumentos utilizados para cumplir con el desarrollo de la aplicación web progresiva RespedApp con SSR están enlistados en la siguiente **Tabla 3-3**:

Tabla 3-3: Instrumentos que han sido utilizados en el desarrollo e implementación de la aplicación RespedApp

Herramientas	<ul style="list-style-type: none">• Azure DevOps• Git• SASS• Sequelize• Postman
Servicios en la nube	<ul style="list-style-type: none">• GitHub• Cloudinary
IDE	<ul style="list-style-type: none">• Visual Studio Code
Gestor de base de datos	<ul style="list-style-type: none">• MySQL + phpMyAdmin
Entorno en tiempo de ejecución	<ul style="list-style-type: none">• Node.js
Frameworks	<ul style="list-style-type: none">• Express.js, Next.js, Material UI,

	Bootstrap
Librerías	<ul style="list-style-type: none"> • React.js
Paquetes	<ul style="list-style-type: none"> • Express validator
Computadoras portátiles	<ul style="list-style-type: none"> • Computadora ASUS Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 32Gb RAM 1Tb HDD • Computadora Dell Inspiron 1493 Intel(R) Core(TM) i5 @ 1.70GHz 12Gb RAM 1Tb HDD + 512Gb
Servidores Web	<ul style="list-style-type: none"> • DigitalOcean

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.2.4 *Objetivo 4: Evaluar la eficiencia y disponibilidad que tiene la gestión de reservas y pedidos de la aplicación web progresiva a través del modelo ISO/IEC 25000*

Para cumplir con el presente objetivo se ha optado por plantear una hipótesis para determinar la calidad de la gestión de reservas y pedidos. Como se había mencionado previamente, la estructura del proyecto contiene un sistema de *backend* y otro de *frontend*. Las características de eficiencia y disponibilidad se evaluarán con respecto al sistema de *frontend* dado que es donde se implementa la aplicación web progresiva con SSR, y se descartará la medición de tiempos de *backend* debido a que es un análisis que no se encuentra dentro del alcance del presente trabajo de integración curricular.

3.2.4.1 *Definición de niveles de puntuación final para calidad.*

La escala de medición que se presenta en la **tabla 4-3** se utilizará para analizar el resultado final de características de calidad de la aplicación, valores que determinarán el nivel obtenido final y que se asignará al producto software después de su análisis.

Tabla 4-3: Niveles de puntuación final de calidad para la aplicación RespedApp.

Escala de medición	Nivel de puntuación	Grado de satisfacción
8.76 – 10.00	Cumple con requisitos	Muy satisfactorio
5.10 – 8.75	Aceptable	Satisfactorio
2.76 – 5.00	Mínimamente aceptable	
0.00 – 2.75	Inaceptable	Insatisfactorio

Fuente: (ISO/IEC 25040, 2011)

La calidad del producto de software RespedApp están determinadas por su eficiencia y

disponibilidad en los porcentajes que se visualizan en la **tabla 5-3**. Las subcaracterísticas que se ha optado para analizar en el trabajo de integración curricular son el comportamiento temporal debido a que la misma incluye parámetros de tiempo que son evaluables respecto al proyecto, además se analizará la subcaracterística de disponibilidad perteneciente a la característica de fiabilidad. El desarrollo de RespedApp aspira cumplir con estas características de calidad, mejorar el proceso actual de la gestión de pedidos y reservas del restaurante, y así cumplir con el objetivo principal de brindar una aplicación de calidad.

Tabla 5-3: Ponderación de las características de calidad de la aplicación RespedApp

Características	Subcaracterísticas	Métricas	Porcentaje máximo	Valor máximo ponderado
Eficiencia en el desempeño	Comportamiento en el tiempo	Tiempo de Respuesta	15%	1.5
		Rendimiento	15%	1.5
Fiabilidad	Disponibilidad	Tiempo de Servicio	70%	7
Total			100%	10

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

El motivo por el que se ha descartado las subcaracterísticas de utilización de recursos y capacidad es que se está evaluando un proceso realizado por personas contra un proceso realizado a través de una aplicación web, siendo el tiempo el único factor que determine si la eficiencia de los procesos ha mejorado.

3.2.4.2 *Métodos*

Como método para cumplir con el objetivo cuatro, se optó por utilizar método cuasi experimental debido que permite consiste en poner a prueba la validez de una hipótesis sometiéndola a experimentación. Algunas características que debe cumplir este método son: Ser de tipo cuantitativo, bajo situaciones controladas, se trabaja con una muestra, etc. En el proyecto sirve para obtener datos a través de una simulación en la aplicación RespedApp y determinar si se acepta la hipótesis nula planteada previamente o se rechaza y se acepta la hipótesis alternativa.

3.2.4.3 *Técnicas*

Para receptar los valores requeridos para el análisis de la eficiencia de calidad del software se realizó cuarenta tomas de pedidos locales y una toma por reserva. Se utilizará la prueba de Shapiro-Wilk para determinar la distribución de los datos, debido a que es una prueba potente utilizada en grupos de datos menores a 50. Una vez obtenido el valor p, se utilizará T Student o Winconson para comparar si existe una diferencia estadísticamente significativa entre los dos grupos de datos a evaluar, es decir los tiempos de respuesta sin la aplicación y con la aplicación RespedApp. Para medir la disponibilidad se ha realizado pruebas ping de disponibilidad con la aplicación de monitoreo de Azure: Application Insights.

- **Formulación de Hipótesis**

Una hipótesis es una afirmación que puede o no ser cierta. Sin embargo, se formula en base a un indicio o a una serie de hechos, a los cuales se puede añadir determinados supuestos (Westreicher, 2022).

La hipótesis según Westreicher (2022) debe cumplir con estas características:

- Las variables con las que se quiere probar una hipótesis deben de ser reales y tener relación con dicha hipótesis.
- Para que una hipótesis sea probada debe de estar fundamentada en circunstancias que puedan ser observadas.
- Las variables sobre las que se fundamenta una hipótesis deben de poder ser cuantificadas y estudiadas.
- Las hipótesis deben de poder ser tanto aprobadas como rechazadas, si sólo contemplan una de estas opciones, no puede ser consideradas hipótesis.

- **Hipótesis nula y alterna**

HN:

La aplicación web progresiva RespedApp tendrá un grado de calidad menor a 8.76.

HA:

La aplicación web progresiva RespedApp tendrá un grado de calidad mayor o igual a 8.76.

Para establecer el puntaje que se utilizará para verificar que la hipótesis establecida, sí se ha cumplido o no, se tomó en cuenta, que característica de calidad es más relevante para el propietario del restaurante el “El Fogón de COZ”, como se definió en la entrevista que se encuentra en el Anexo A.

Según el requerimiento del propietario, la disponibilidad tendrá una ponderación de 7 puntos, que equivale al 70% de calidad del sistema, mientras que la eficiencia se ponderará sobre 3 puntos,

que representa el 30% de la calidad del sistema.

3.2.4.4 *Medición de Eficiencia*

La eficiencia dentro del presente trabajo de integración curricular se presenta como el tiempo de respuesta y rendimiento que tiene la aplicación RespedApp bajo condiciones determinadas. Las métricas que define la ISO/IEC 25000 con relación al proyecto son las siguientes: El tiempo de respuesta del servicio (tiempo del proceso de registro de pedidos y reservas) y Rendimiento (número de registros de pedidos y reservas / tiempo total de evaluación).

- **Población y Muestra**

La información obtenida de la toma de datos del proceso de registro de pedidos y reservas se dio a través de la técnica de muestreo intencional. Esta técnica permitirá al investigador elegir el número de pedidos y reservas que se evaluarán. Se optó por analizar cuarenta pedidos y una reserva debido a que esta es la media que se suele tener en el restaurante diariamente. Los datos fueron extraídos el día 16 de Julio del 2022.

- ***Profiling Speed Site***

Para obtener los tiempos de respuesta se utilizó la técnica de *profiling*, que consiste en grabar un perfil de rendimiento que incluya los tiempos de carga, *scripting*, renderizado, dibujado, sistema, entre otros; y así determinar el tiempo de respuesta de los procesos a evaluar.

- **Distribución de datos**

Una vez obtenidos los datos, se procedió a realizar una prueba de normalidad. Se optó por utilizar el test de Shapiro-Wilk debido a que se recomienda esta prueba cuando se tiene menos de 50 elementos en la muestra, además que es uno de los más utilizados para realizar análisis estadísticos.

- **Comparación de muestras**

Para determinar si existe una diferencia significativa entre los tiempos de respuesta sin la aplicación RespedApp y con la misma se utilizó el test no paramétrico de Wilcoxon. Esta prueba se suele utilizar cuando los datos no siguen una distribución normal y para ello se compara sus medianas determinando su diferencia estadísticamente.

3.2.4.5 *Medición de Disponibilidad*

La disponibilidad dentro del presente trabajo de integración curricular se presenta como la capacidad que tiene RespedApp para estar accesible para el cliente, las métricas evaluadas son el tiempo de servicio (tiempo de actividad de la aplicación/ tiempo de actividad acordado con el cliente en el SLA (*Service Level Agreement*) y el tiempo medio de inactividad (Tiempo de baja de la aplicación/ número de fallos).

- **Prueba Ping**

Las pruebas ping realizan solicitudes HTTP para validar si un punto de conexión responde cada determinado tiempo. Se realizó alrededor de docemil pruebas ping con una frecuencia de 5 minutos durante una semana, es decir diez mil ochenta minutos desde el 05/07/2022 hasta el 12/07/2022.

3.2.4.6 *Fuentes*

Para determinar las métricas de las subcaracterísticas de la eficiencia y disponibilidad de un producto de software se tomó como referencia el trabajo de titulación de Evelyn Balseca denominado: “Evaluación de calidad de productos de software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000”.

3.2.4.7 *Instrumentos*

R: Es un entorno enfocado al análisis estadístico de datos.

Chrome DevTools: Una herramienta gratuita de Chrome es el perfil de rendimiento el cual brinda una amplia gama de información disponible que puede correlacionar diferentes tipos de actividad de la página e identificar los tiempos de rendimiento.

Application Insights Azure: Es una característica de Azure Monitor que proporciona la administración extensible del rendimiento y disponibilidad para aplicaciones web en vivo.

3.3 Aplicación de la metodología Scrumban en el desarrollo del software RESPEDAPP

Como se mencionó previamente el presente trabajo de integración curricular utiliza la metodología de desarrollo de software Scrumban. La estrategia Scrum ofrece describir las historias de usuario, valores, roles y pautas sobre cómo trabajar de manera efectiva con los sprints o iteraciones. Por otra parte, Kanban sirve como marco ágil para distribuir equitativamente el trabajo según la capacidad productiva disponible de cada uno de los integrantes.

A continuación, se describirá cada una de las fases aplicadas al proyecto, para conocer más acerca del proceso de cada una de las fases dirigirse al Anexo B que incluye un paso a paso de cada fase.

3.3.1 Fase 1: Análisis preliminar

El fogón de Coz es un restaurante muy popular que se encuentra al servicio de la ciudad de Baños de Agua Santa – Ecuador. Este restaurante tiene la necesidad de implementar un sistema de automatización de procesos, el cual surgió del crecimiento que ha tenido en el último año, viéndose en la necesidad de tener un sistema propio para llevar a cabo el control de reservas y pedidos que se realizan en este restaurante a lo largo de cada día. En la actualidad estos procesos se encuentran manejados rústicamente por el personal del lugar, lo cual genera demora y pérdida de tiempo al momento de brindar el servicio a sus clientes.

3.3.1.1 Personas y roles del sistema

Scrum se caracteriza por su excelente gestión en proyectos y construir software de calidad, lo cual se tomó en cuenta los roles en base a la metodología aplicada. Para el equipo de desarrollo se asignaron los siguientes roles **Tabla 6-3**:

Tabla 6-3: Roles de la Metodología Scrum para el proyecto RespedApp

Persona	Rol
Chef Alejandro Cabrera	Product Owner
Ing. Gloria Arcos	Scrum Master
Emily Chimbo	Developer
Angello Ordoñez	Developer

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

En la **Tabla 7-3** se describe los roles que tienen cada uno de los usuarios, los cuales fueron definidos por el *product owner*.

Tabla 7-3: Roles del aplicativo RespedApp

Usuario	Rol
Administrador	Visor reservas y pedidos Gestión de menú Gestión de productos Gestión de reservas

	Gestión de pedidos Realizar reservas Realizar pedidos Autenticarse
Mesero	Visor reservas y pedidos Gestión de reservas Gestión de pedidos Realizar reservas Realizar pedidos Autenticarse
Cliente	Realizar reservas Realizar pedidos Autenticarse

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.1.2 *Proceso actual del control de proyectos*

Con el objetivo de describir el proceso actual que realiza el restaurante “El fogón de Coz” para el control de reservas y pedidos se detalla cada una de las actividades que siguen los involucrados que intervienen para que se desarrolle cada uno de estos procesos.

Para el proceso de reservas existen dos actores: el cliente y administrador. El proceso inicia con la solicitud del cliente hacia el lugar, posterior a ello se determina si existe espacio o no, en el caso afirmativo se registra la reserva y se puede adjuntar un pedido, o en el caso que no exista lugar no se registra la reserva como se lo visualiza en la **Figura 4-3**.

Para el proceso de pedidos están involucrados cuatro actores: el cliente, el mesero, el administrador y cocina. Se inicia con la visualización del menú por parte del cliente, posterior a ello se escogen los alimentos, se determina el lugar de entrega, si es local el mesero estará encargado de informar a cocina y llevar los alimentos. En el caso de domicilio, se le informa a cocina y el administrador es el encargado de gestionar el envío al cliente. Finalmente se realiza el pago de los productos como se lo visualiza en la **Figura 5-3**.

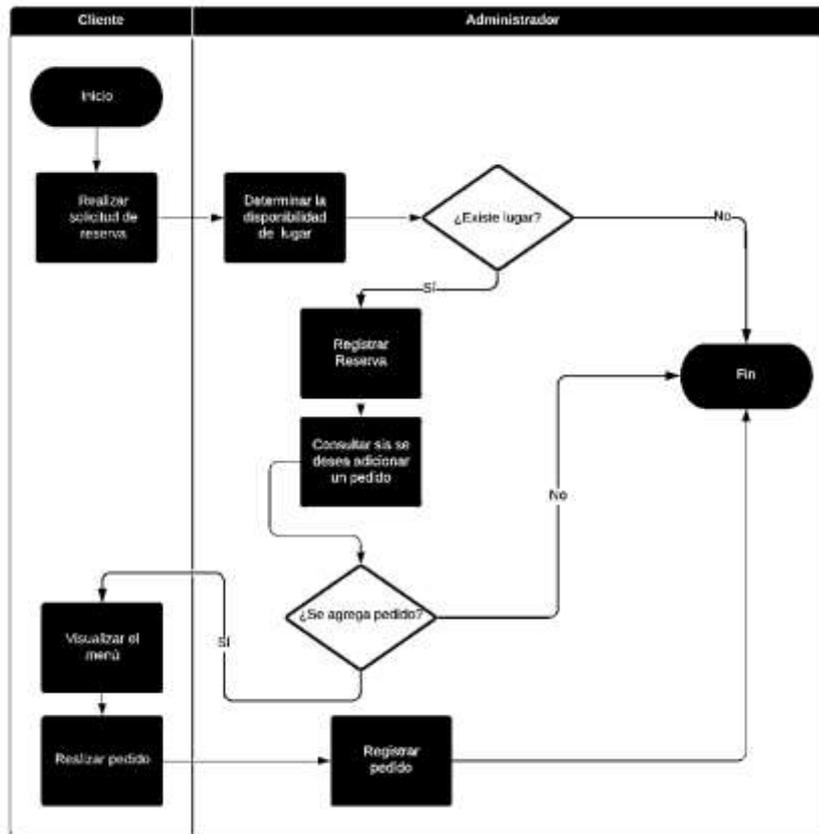


Figura 4-3: Proceso actual de reservas del restaurante “El fogón de Coz”

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

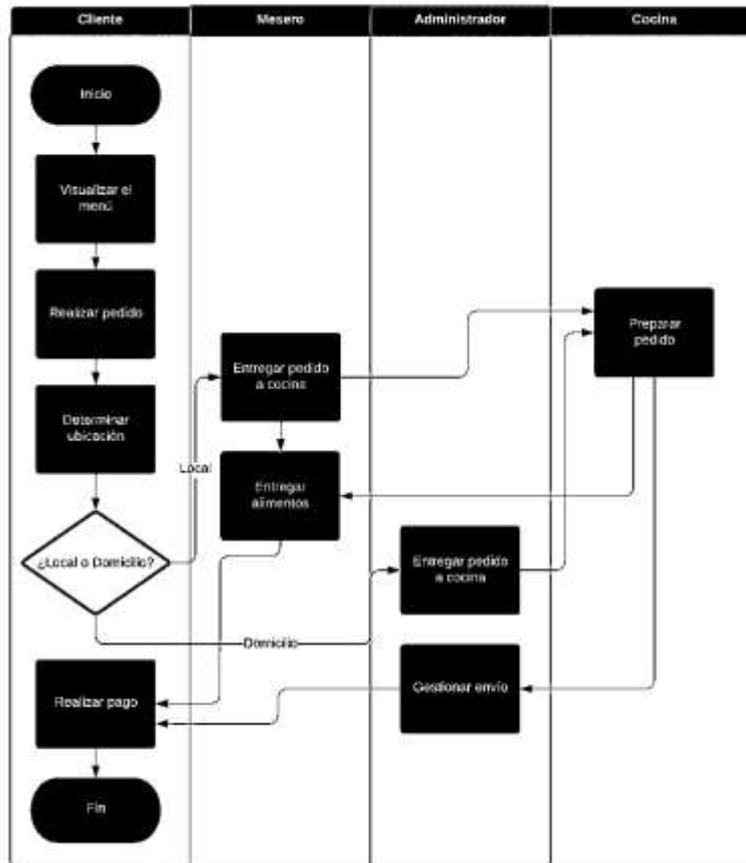


Figura 5-3: Proceso actual de pedidos del restaurante “El fogón de Coz”

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.1.3 Especificación de Requerimientos

En esta sección se detalla los requerimientos que debe cumplir la aplicación a manera de historias de usuario debido a la metodología empleada a través de la siguiente **tabla 8-3**:

Tabla 8-3: Historias de Usuario requeridas para la aplicación web progresiva RespedApp

ID	Title	Description
RQ01	Creación de Registro	Como cliente quiero una página que incluya un formulario con datos para registrarme en la aplicación
RQ02	Creación de Ingreso	Como cliente quiero una página que incluya un formulario con datos para ingresar en la aplicación
RQ03	Creación del Menú	Como usuario quiero visualizar el menú clasificado por categorías y productos para

ID	Title	Description
		posteriormente elegir los productos deseados y realizar un pedido en el restaurante
RQ04	Solicitud de Reserva	Como usuario quiero solicitar una reserva en el restaurante para asegurar un lugar y poder asistir sin necesidad de hacer filas
RQ05	Creación de Pedidos Locales	Como usuario quiero visualizar una página que incluya el pedido previo, registrar número de mesa y nota de pedido para ingresar un pedido local en la aplicación.
RQ06	Creación de Pedidos Domicilio	Como usuario quiero visualizar una página que incluya el pedido previo, registrar dirección y nota de pedido para ingresar un pedido domicilio en la aplicación.
RQ07	Creación de Pedidos Reserva	Como usuario quiero visualizar una página que incluya el pedido previo junto al número de reserva y nota de pedido para ingresar un pedido reserva en la aplicación.
RQ08	Creación de Pago	Como usuario quiero pagar a través de Paypal o tarjeta de crédito mi pedido para evitar pagar en efectivo
RQ09	Creación de Galería	Como usuario quiero visualizar en la página principal una galería de fotos para obtener una referencia de la comida del restaurante El fogón de Coz.
RQ10	Creación de Nosotros	Como usuario quiero visualizar una sección nosotros en la página principal para obtener información descriptiva e histórica del restaurante El fogón de Coz
RQ11	Creación de Contacto	Como usuario quiero visualizar una sección de contacto que incluya un formulario para lograr comunicación con el restaurante sin necesidad de registro.
RQ12	Creación de Carrito	Como usuario quiero una página que muestre los productos que escogí para visualizar un resumen del pedido

ID	Title	Description
RQ13	Creación de Sección Usuarios	Como administrador quiero una página que me permita insertar, editar y eliminar usuarios de la aplicación para gestionar los usuarios
RQ14	Creación de Sección Pedidos	Como administrador quiero una página que me permita ver y editar los pedidos de la aplicación para gestionar los pedidos
RQ15	Creación de Sección Reservas	Como administrador quiero una página que me permita ver y editar las reservas de la aplicación para gestionar las reservas
RQ16	Creación de Sección Productos	Como administrador quiero una página que me permita insertar, editar y eliminar las categorías de comida y productos de la aplicación para gestionar los productos.
RQ17	Creación de Sección Perfil	Como cliente quiero una página que me permita ver y editar mis datos para gestionar mi cuenta en la aplicación
RQ18	Creación de Sección Ordenes	Como cliente quiero una página que me permita ver los pedidos que he realizado en la aplicación para gestionar mis pedidos
RQ19	Creación de Dashboard Restaurante	Como administrador/mesero quiero visualizar un tablero de pedidos para cambiar el estado de los pedidos del restaurante Como administrador/mesero quiero visualizar una tabla de reservas del día para obtener la información de reservas diarias del restaurante
RQ20	Creación de Sección Direcciones	Como cliente quiero una página que me permita insertar editar y eliminar mis direcciones para gestionar mis próximos pedidos a domicilio.

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.1.4 Estudio de Factibilidad

El estudio de factibilidad técnico dio como resultado que se cuenta con el equipo necesario para el desarrollo de la aplicación web progresiva para la gestión de reservas y pedidos del restaurante

“El fogón de COZ” con implementación de *server side rendering* debido a que se posee tanto el hardware como el software, en conjunto al equipo de desarrollo que se encuentra capacitado con conocimientos y experiencia necesarios para el desarrollo de cada una de las etapas planificadas.

Se optó por realizar una estimación de costos de COCOMO debido a la baja complejidad en su implementación, toma en cuenta varios factores y sus resultados son fáciles de comprender. La estimación dio tres resultados que son el esfuerzo, el tiempo de duración del proyecto y el personal necesario para ejecutar el proyecto:

- Esfuerzo: 5.17 personas/mes
- Tiempo de duración: 4.66 meses
- Personal promedio: 1.11 = 2 personas aprox.

Operativamente el desarrollo del proyecto es factible, debido a que se posee la tecnología informática necesaria para la puesta en operación del sistema de información propuesto, además el costo de adquirir los servicios de un proveedor externo no sobrepasa el presupuesto establecido.

La estimación del costo del proyecto fue de \$6852 tomando en cuenta los recursos existentes y por adquirir, esto se puede visualizar en el Anexo B sección 3.2.

3.3.1.5 Análisis de Riesgo

El resultado de analizar los posibles riesgos que se puedan presentar en el proyecto RespedApp junto a su plan de mitigación se encuentra descrito en la **tabla 9-3** que se observa a continuación:

Tabla 9-3: Matriz de riesgos y proceso de mitigación para el proyecto RespedApp

Código del Riesgo	Descripción del riesgo	Fase	Nivel del Riesgo	Tipo de Respuesta	Responsable	Plan de mitigación
RI-01	Falta de compromiso por parte del cliente con el proyecto	Análisis / Panificación	Media	Corregir	Angello Ordoñez	<ul style="list-style-type: none"> • Establecer reuniones de emergencia • Evidenciar la asistencia de las reuniones • Enviar

						comunicados por correo electrónico, llamada, mensaje de texto
RI-02	Fallo en los equipos utilizados para el desarrollo	Implementación	Media	Mitigar	Angello Ordoñez / Emily Chimbo	<ul style="list-style-type: none"> • Respaldos diarios en la nube de la documentación • Uso de un sistema de control de versiones
RI-03	Interfaces graficas poco amigables que dificulten obtener los resultados en la fase de pruebas	Revisión y retrospectiva	Poco probable	Mitigar	Angello Ordoñez	<ul style="list-style-type: none"> • Asesoramiento con personal capacitado de UI/UX.
RI-04	Entidades definidas de forma incorrecta o no definidas	Análisis	Baja	Corregir	Emily Chimbo	<ul style="list-style-type: none"> • Reestructurar el diseño de la base de datos
RI-05	Reactivación de la pandemia COVID-19	Todas las fases	Media	Aceptar	Angello Ordoñez / Emily Chimbo	<ul style="list-style-type: none"> • Emplear sistemas de comunicación virtuales
RI-06	El personal del equipo	Todas las fases	Poco probable	Aceptar	Angello Ordoñez /	<ul style="list-style-type: none"> • Tratar de cumplir con

	de desarrollo tenga una calamidad				Emily Chimbo	las actividades del compañero
--	-----------------------------------------	--	--	--	-----------------	----------------------------------

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.2 Fase 2: Planificación

La fase de planificación es importante en el desarrollo del proyecto, debido a que se debe tener una

calendarización con cada tarea en el desarrollo del sistema.

Los requerimientos para el sistema fueron establecidos juntamente con el product owner y la prioridad se estableció en base a la técnica de estimación T-Shirt Size, el cual mide el tamaño de las historias de usuario en base a la experiencia del desarrollador, en referencia a las tallas de ropa.

Dichos detalles están descritos a continuación en la **Tabla 10-3**.

Tabla 10-3: Método T-Shirt para la aplicación web progresiva RespedApp

Talla	Puntos estimados	Tiempo de trabajo
S - Small	10	1 semana
M - Medium	20	2 semanas
L - Large	40	4 semanas
XL – Extra Large	80	8 semanas

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.2.1 Iteraciones

Las iteraciones o sprints son el corazón de la metodología scrum, de aquí se define las partes en las que se divide el proyecto, un sprint tiene una duración de mínimo 2 semanas y máximo de 4 semanas, dentro del sprint se define épicas, características, historias de usuario y tareas lo que se conoce como Product Backlog y la asignación de cada una de las tareas a un sprint se conoce como Sprint Backlog.

Para el desarrollo del proyecto se definió 4 sprints de 4 semanas cada uno. Una semana laboral consta de 6 días, 8 horas diarias.

3.3.2.2 Product Backlog

Consiste en una lista con todos los requerimientos iniciales del producto que se va a desarrollar. Se trata de una lista dinámica, que irá evolucionando a medida que lo hace el producto y el entorno del proyecto. Esta lista detallada se puede visualizar en el Anexo B sección 4.3, la cual incluye las siguientes actividades:

- **Épicas**

Es una historia de usuario que, por su gran tamaño, el equipo descompone en historias con un tamaño más adecuado para ser gestionada con los principios y técnicas ágiles. Se ha definido 4 épicas para este proyecto.

- **Características**

Representan las partes del producto que aportan un valor significativo a los usuarios. Las características suelen ser demasiado grandes para que se puedan trabajar directamente, por lo que se dividen en unidades de negocio más pequeñas: las historias. Se ha definido 10 características para este proyecto.

- **Historias de Usuario**

Aporta valor a los *skateholders*/usuarios, como una nueva función o una mejora en el producto. Se ha definido 20 historias de usuario para este proyecto.

- **Tareas**

Es una unidad de trabajo necesaria para terminar una historia. No es parte del resultado del proyecto, es más bien el medio para producir el resultado. Se ha definido 51 tareas para este proyecto.

- **Historias Técnicas**

Son aquellas tareas que son necesarias para el desarrollo del producto, pero no se ven reflejadas como un requerimiento por parte del cliente.

3.3.2.3 *Sprint Backlog*

Consiste en una lista de elementos seleccionados previamente del Product Backlog, para ser desarrollados en las diferentes iteraciones del proyecto. Tras crear la lista, el equipo del proyecto identifica las funcionalidades y prioriza las que se entregarán en cada Sprint. A continuación, se describe cada sprint:

- **Sprint 1 (04-04-2022 / 29-04-2022)**

En el primer sprint se definió 1 épica, 3 características, 4 historias de usuario y 13 tareas.

- **Sprint 2 (02-05-2022 / 27-05-2022)**

En el segundo sprint se definió 1 épica, 2 características, 8 historias de usuario y 7 tareas.

- **Sprint 3 (30-05-2022 / 24-06-2022)**

En el tercer sprint se definió 1 épica, 3 características, 6 historias de usuario y 8 tareas.

- **Sprint 4 (27-06-2022 / 22-07-2022)**

En el cuarto sprint se definió 1 épica, 2 características, 2 historias de usuario y 23 tareas.

Como resultado final se obtuvo el Product y Sprint Backlog, los cuales tienen de propósito realizar un proceso de desarrollo de software con orden, y obtener resultados satisfactorios. Tomando en cuenta que cada historia de usuario tiene sus puntos estimados asignados y su prioridad, la cual se toma en cuenta desde la más alta hasta la prioridad baja, como se visualiza en la **Tabla 11-3**.

Tabla 11-3: Estimación de historias de usuario para la aplicación web progresiva RespedApp.

Sprint	Historia de Usuario	Descripción	Prioridad	Puntos Estimados
1	HT01	Diseño e implementación de la base de datos	1	10
1	HT02	Instalación y configuración de servidores	1	10
1	HU01	Creación de Registro	1	10
1	HU02	Creación de Ingreso	1	10
1	HU03	Creación del Menú	1	20
1	HU04	Solicitud de Reserva	1	20
2	HU05	Creación de Pedidos Locales	1	10
2	HU06	Creación de Pedidos Domicilio	1	10
2	HU07	Creación de Pedidos Reserva	1	10
2	HU08	Creación de Pago	1	10
2	HU09	Creación de Galería	2	10
2	HU10	Creación de	2	10

		Nosotros		
2	HU11	Creación de Contacto	2	10
2	HU12	Creación de Carrito	2	10
3	HU13	Creación de Sección Usuarios	2	10
3	HU14	Creación de Sección Pedidos	2	10
3	HU15	Creación de Sección Reservas	2	20
3	HU16	Creación de Sección Productos	2	20
3	HU17	Creación de Sección Perfil	2	10
3	HU18	Creación de Sección Ordenes	2	10
4	HU19	Creación de Dashboard Restaurante	3	40
4	HU20	Creación de Sección Direcciones	3	40

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

Como resultado tenemos un total de 20 historias de usuario, 2 historias técnicas y un total de 320 puntos estimados. El desarrollo del proyecto tiene una duración de 16 semanas de trabajo y equivalen a un total de 4 Sprints, ya que cada sprint vale por cuatro semanas de trabajo y 80 puntos estimados por semana.

3.3.2.4 Estructura del tablero Kanban

El tablero Kanban es una de las características de la metodología seleccionada, en el tablero que se muestra en la **Figura 6-3** a través de la aplicación Azure DevOps es donde se manejó la gestión de las historias de usuario planteadas para el trabajo de titulación, el tablero se dividió en las siguientes partes:

- **New:** Aquí se listaron todas historias de usuario del proyecto y tareas internas de las mismas

conforme se iba avanzando los sprint.

- **Active:** En el estado activos se encontraban todas las historias de usuario y tareas que se comenzaron a realizar.
- **Resolved:** En este estado, las historias de usuario y tareas ya se encontraban resultas y listas para ser revisadas en cada final del sprint
- **Closed:** Por último, en esta etapa, se ponen las tareas que se encuentran ya finalizadas y revisadas por el equipo de desarrollo.

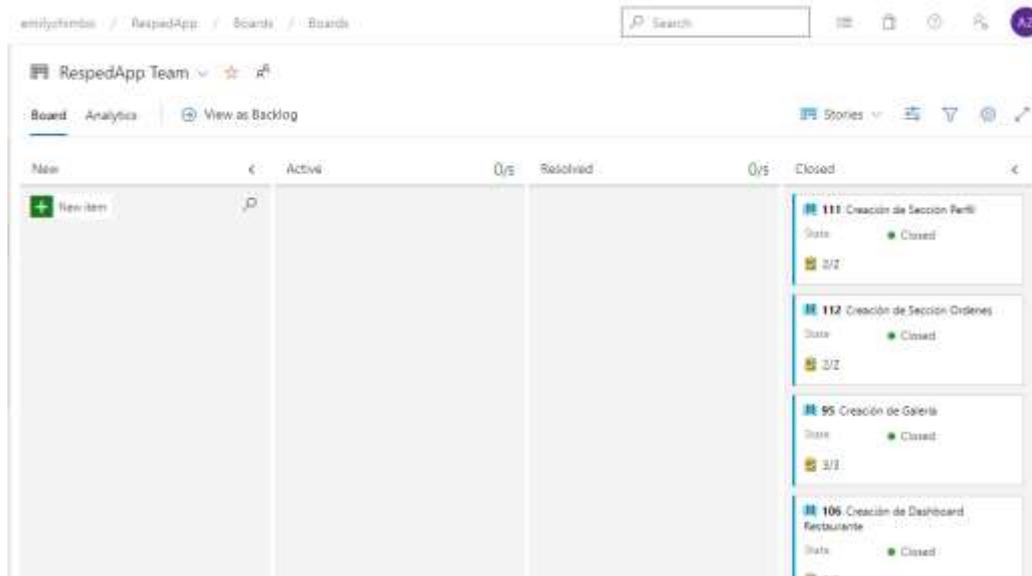


Figura 6-3: Estructura del tablero Scrumban utilizado para la gestión de historias de usuario

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.3 Fase 3: Implementación

3.3.3.1 Módulos

Un módulo de una aplicación es una unidad de software que podría trabajar de manera independiente a otras partes de la aplicación. A continuación, se describirá los módulos desarrollados a través de un diagrama jerárquico en la **Figura 7-3**.

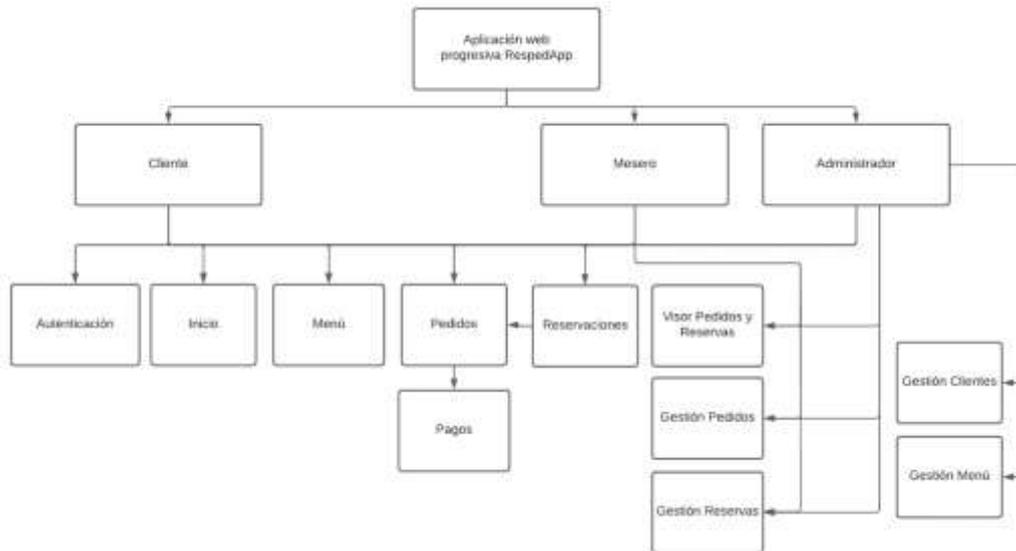


Figura 7-3: Diagrama jerárquico de módulos de la aplicación Respedapp

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.3.2 Arquitectura

La arquitectura REST ((Representational State Transfer) esta es una arquitectura no define con que tecnologías desarrollar, simplemente define como transferir los datos entre componentes, esta arquitectura se basa típicamente en cliente-servidor, usualmente para consumir los servicios REST desde el cliente se envían peticiones mediante AJAX al servidor, donde este se encarga de procesarlas, ya sea para leer, modificar, eliminar o agregar un recurso, para realizar estas acciones se debe tomar en cuenta que cada acción está definida por un verbo del protocolo HTTP de los cuales se emplearan los siguientes GET, POST, PUT, DELETE, cabe señalar que existen otros verbos, la arquitectura REST es transferencia de estado representacional que no guarda estados, sino que se limita a devolver lo que es solicitado por el cliente, además es ampliamente escalable ya que al ofrecer APIs están pueden ser consumidas desde cualquier plataforma ya que se regresan respuesta en formato de texto denominado JSO, para el desarrollo del aplicativo RespedApp se dividió en dos servidores como se lo puede visualizar en la Figura 8-3 (Dept. Ciencia de la computación, 2013).

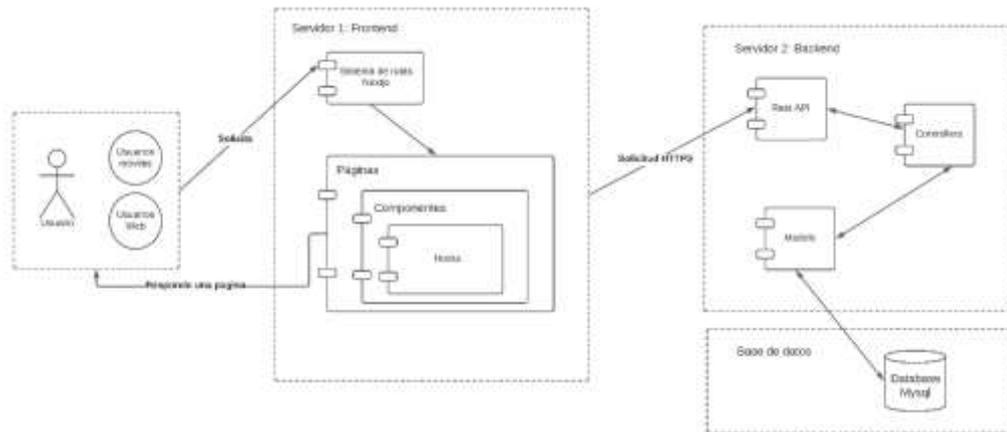


Figura 8-3: Diagrama de componentes del aplicativo RespedApp

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.3.3 *Diseño de base de datos*

Para el diseño de la base de datos para el aplicativo RespedApp se partió con la elaboración del diagrama de entidad relación simplificado, el cual puede ser encontrado en el Anexo B sección 4.2.1 a partir de este diagrama se encontraron seis entidades y seis relaciones, a la cual se le aplicó la técnica de normalización hasta la tercera forma normal.

Después de aplicar la normalización de tablas de la base de datos de RespedApp y la elaboración del diagrama lógico y físico se obtuvo 14 tablas en total como se lo puede observar en la **Figura 9-3**, se utilizó el gestor de base de datos de MySQL en su versión 8.0.29 para la creación y gestión de la base de datos. Además, también se realizó una estimación de la base de datos el cual se ha logrado definir un consumo de espacio en memoria de 90MB anual. Para conocer más acerca sobre los procesos realizados para el diseño de la base de datos, normalización, diccionario y estimación, se recomienda revisar el Anexo B sección 4.2.

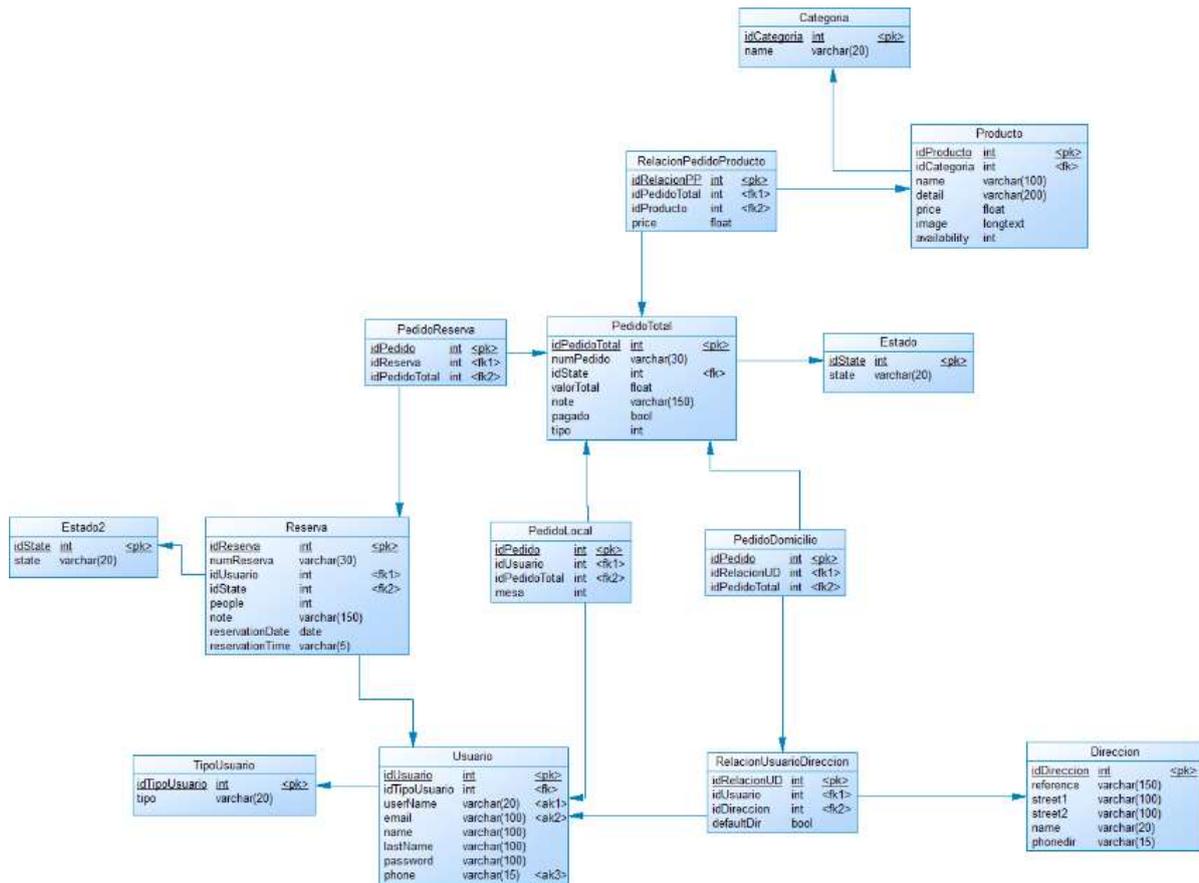


Figura 9-3: Diagrama físico de la base de datos del aplicativo RespedApp

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.3.4 Lenguaje de Programación

Para realizar el proyecto se ha decidido optar por el lenguaje de programación JavaScript debido a que es uno de los más populares para el desarrollo web, además, de su extensa documentación y librerías disponibles para un desarrollo eficiente y completo. Además, gracias al entorno de desarrollo de Node.js da la posibilidad de utilizar el mismo lenguaje de Javascript para el backend, de igual forma para el frontend se empleó React.js la cual es una librería de Javascript desarrollada por Facebook, basada en componentes, con el fin de darle más potencial a esta librería se utilizó el *framework* de Next.js. Este es un *framework* de React.js que potencializa todas las características de esta librería, además, también facilita el renderizado desde el servidor, debido a que ya tiene una preconfiguración establecida del servidor, por último, este *framework* tiene paquetes externos que agrega más características, como por ejemplo el paquete que ayuda a construir una PWA.

3.3.3.5 Diseño de Interfaces

Para la elaboración de las interfaces, se parte del diseño de los *wireframes*, el cual permitió tener

una idea inicial de como estará estructurada la información, luego se definió la paleta de colores, fuentes e iconos. También se utilizó figma para diseñar una interfaz final aplicando los colores y fuentes, en la **Figura 10-3** se puede observar la primera sección de la página principal del aplicativo RespedApp, en cambio en la **Figura 11-3** se presenta como se verá el menú del restaurante. En el Anexo B sección 3.5, se podrá encontrar todos los bosquejos del aplicativo.



Figura 10-3: Bosquejo final de la sección de la página principal de RespedApp

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.



Figura 11-3: Bosquejo final del menú de RespedApp

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

3.3.4 Fase 4: Revisión y Retrospectiva

Para cumplir con esta fase se optó por utilizar el artefacto de *Sprint Review/Retrospective* de la metodología Scrum. En total se realizó 4 reuniones de revisión las cuales se encuentran detalladas en el Anexo B sección 5.

3.3.5 Fase 5: Lanzamiento y Cierre

Al momento de poner en producción a la aplicación se optó por utilizar un *droplet* que permite hospedar aplicaciones web en el lenguaje de programación que se prefiera y sin necesidad de administrar la infraestructura. Este cambio de contratación de servicio en la nube implicó un sobrante de \$117.37. Para ver los detalles de los costos del servicio revisar el Anexo B sección 6. A continuación, se visualizará en la **figura 12-3** la estructura del despliegue de la aplicación web progresiva RespedApp:

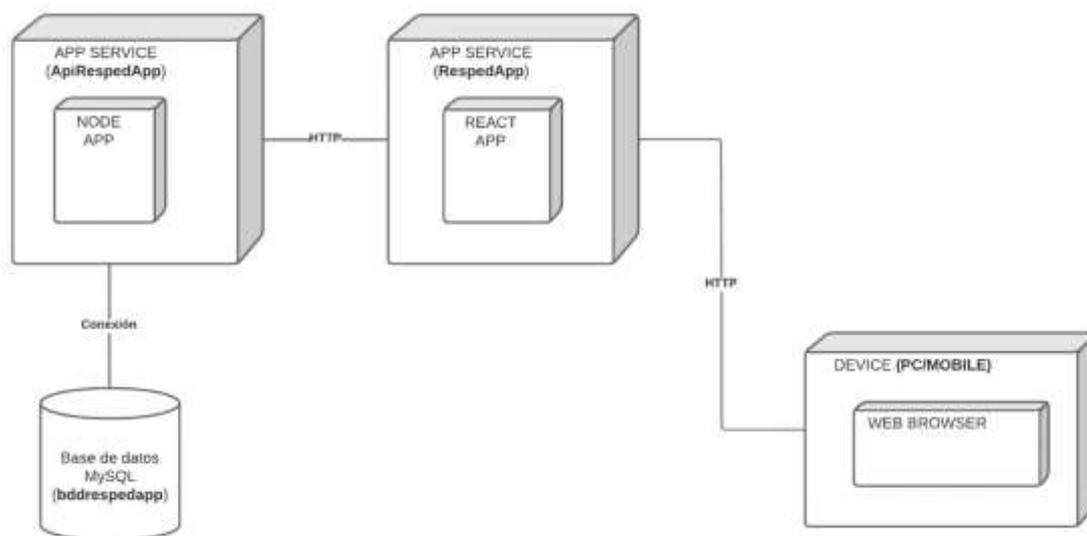
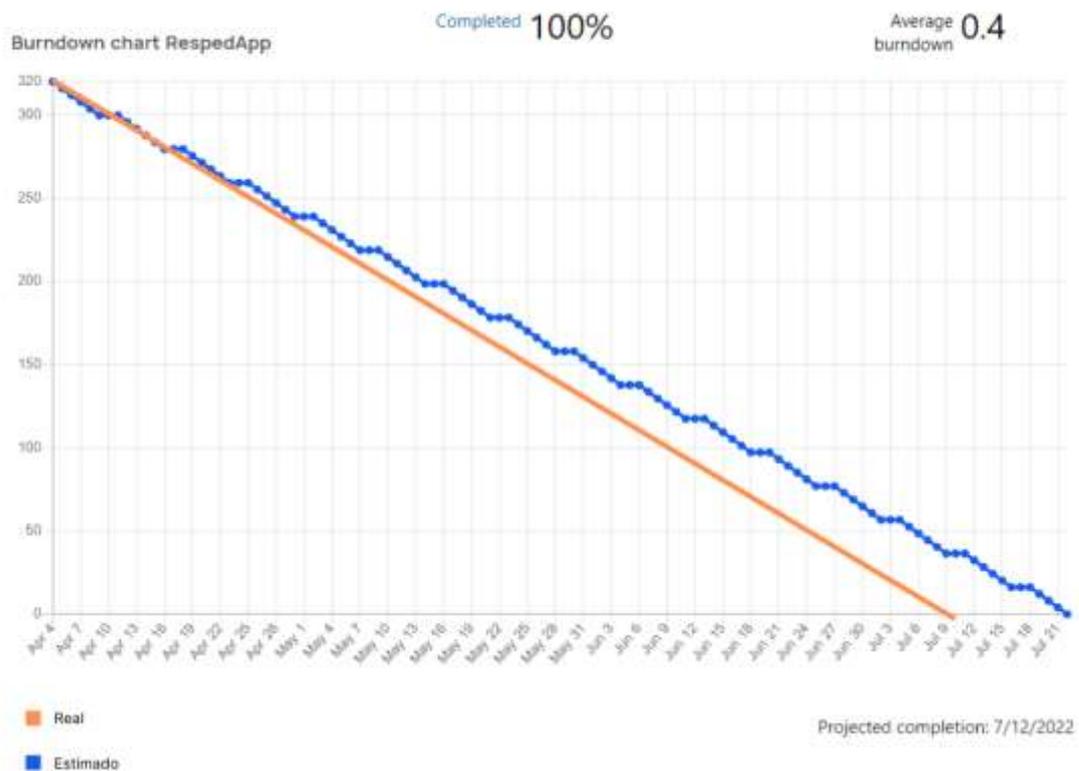


Figura 12-3: Diagrama de despliegue de la aplicación RespedApp

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

Se puede visualizar la aceptación de entregables en el documento de acta de cierre del proyecto en el Anexo D.

Con el objetivo de verificar si la estimación que se realizó al inicio del proyecto concuerda con los tiempos obtenidos al finalizarlo se realiza el BurnDown Chart, donde se refleja la comparación de los puntos reales (línea naranja) y los puntos estimados (línea azul), los cuales equivalen a las horas que se trabajó en el desarrollo del sistema. Esta comparación se detalla en la **figura 13-3**.



CAPITULO IV

4 MARCO DE RESULTADOS Y DISCUSIÓN

En el presente capítulo se presenta los resultados de la medición de calidad de la aplicación RespedApp con respecto a sus subcaracterísticas de eficiencia y disponibilidad para el proceso de registro de reservas y pedidos del restaurante “El fogón de Coz” basado en la norma ISO/IEC 25000.

4.1 Resultado de métricas de eficiencia

Para realizar la medición de la eficiencia se evaluó el proceso de registro de pedidos y reservas previo al uso de la aplicación y con el uso de esta. A partir de estos datos que se visualizan en la **Tabla 12-4** se realizará el análisis de las métricas de tiempo de respuesta y rendimiento.

Tabla 12-4: Resultados de toma de tiempos de respuesta con y sin la aplicación RespedApp.

Número de Cliente	Tiempo sin la aplicación	Tiempo con la aplicación
1	1.05	0.48
2	1.18	0.62
3	0.98	0.53
4	0.97	0.35
5	1.12	0.55
6	0.93	0.40
7	0.95	0.42
8	1.05	0.70
9	0.88	0.45
10	1.03	0.52
11	1.13	0.48
12	0.92	0.53
13	1.15	0.58
14	1.05	0.50
15	1.08	0.52
16	0.93	0.37
17	1.22	0.70
18	1.43	0.83
19	1.33	0.63

20	0.87	0.45
21	1.32	0.63
22	1.12	0.52
23	1.68	1.02
24	1.27	0.75
25	1.38	0.50
26	1.17	0.48
27	1.22	0.55
28	1.08	0.45
29	0.95	0.48
30	1.15	0.52
31	1.10	0.62
32	1.48	1.08
33	0.90	0.72
34	1.32	0.55
35	1.05	0.57
36	1.20	0.75
37	1.77	0.60
38	1.00	0.78
39	1.27	0.43
40	1.28	0.67
41	1.07	0.42
Suma	47.03	23.70
Promedio	1.15	0.58

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

- **Prueba de normalidad de los datos**

Para determinar si los datos obtenidos con la aplicación RespedApp siguen una distribución normal se aplicó la prueba de Shapiro-Wilk a través de RStudio. Dando como resultado que p-value es menor a 0.05 como se visualiza en la **figura 14-4** y en el **gráfico 1-4** que sirve para visualizar la distribución de datos en un intervalo de tiempo. Se rechaza la hipótesis nula y se acepta la hipótesis alterna, lo que determina que los tiempos obtenidos no tienen una distribución normal.

```

> Grupo.B = c(0.483333333333333,0.616666666666667,0.533333333333333,0.35,0.55,0.4,0.416666666666667,0.7,0.45,
0.516666666666667,0.483333333333333,0.533333333333333,0.583333333333333,0.5,0.516666666666667,0.366666666666666
7,0.7,0.833333333333333,0.633333333333333,0.45,0.633333333333333,0.516666666666667,1.016666666666667,0.75,0.5,
0.483333333333333,0.55,0.45,0.483333333333333,0.516666666666667,0.616666666666667,1.083333333333333,0.716666666
666667,0.55,0.566666666666667,0.75,0.6,0.783333333333333,0.433333333333333,0.666666666666667,0.416666666666666
7)
> shapiro.test(Grupo.B)

      shapiro-wilk normality test

data:  Grupo.B
w = 0.8968, p-value = 0.001336

```

Figura 14-4: Prueba Shapiro-Wilk en R de los datos de tiempo de respuesta con la aplicación RespedApp

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

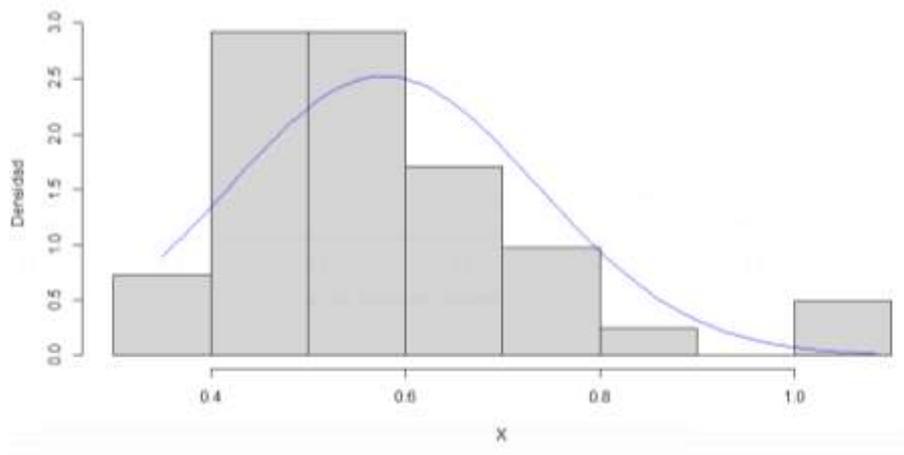


Gráfico 1-4: Distribución de tiempos de respuesta de la aplicación RespedApp

Realizado por: Autoría propia Ordoñez (2022), Chimbo (2022).

- **Resultado de la comparación de las muestras**

El valor p de la prueba de Wilcoxon es 0.00000000000005332, que es menor que el nivel de significación $\alpha = 0,05$. Podemos concluir que la mediana del tiempo con la aplicación es significativamente diferente de la mediana del tiempo sin la aplicación como se visualiza en la **figura 15-4** con un valor de $p = 5.332e-14$. Además, a través del diagrama de caja y bigotes en el **gráfico 2-4** podemos visualizar esta diferencia entre los dos conjuntos de datos.

```

> wilcox.test(Grupo.A, Grupo.B, conf.level = 0.95, exact = FALSE)

      wilcoxon rank sum test with continuity correction

data:  Grupo.A and Grupo.B
W = 1652, p-value = 5.332e-14
alternative hypothesis: true location shift is not equal to 0

```

Figura 15-4: Prueba Wilcoxon en R de los datos de tiempo de respuesta con y sin la aplicación RespedApp

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

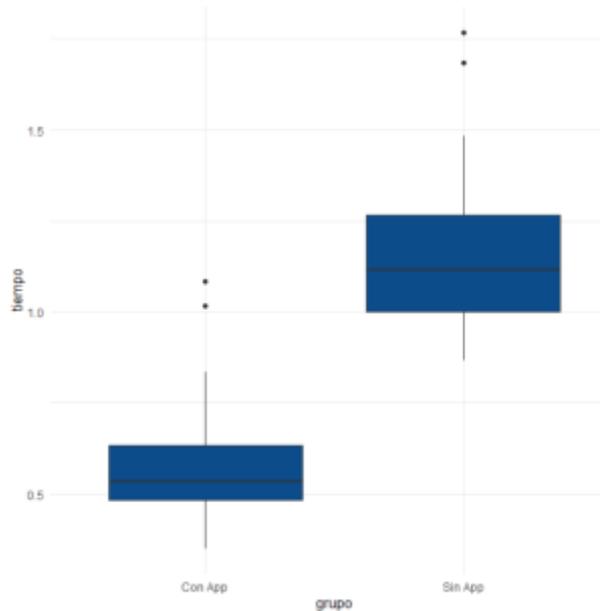


Gráfico 2-4: Diagrama de caja y bigotes con los datos de tiempo de respuesta con y sin la aplicación RespedApp

Realizado por: Autoría propia Ordoñez (2022), Chimbo (2022).

4.1.1 *Tiempo de respuesta*

Una vez analizados los datos y determinando que, si existirá una diferencia significativa entre los tiempos de respuesta con y sin la aplicación RespedApp, se procede a evaluar el tiempo promedio entre los dos grupos de datos. El tiempo promedio sin la aplicación es de 1.15 minutos, mientras que con la aplicación es de 0.58 minutos como se visualiza en el **gráfico 3-4**.

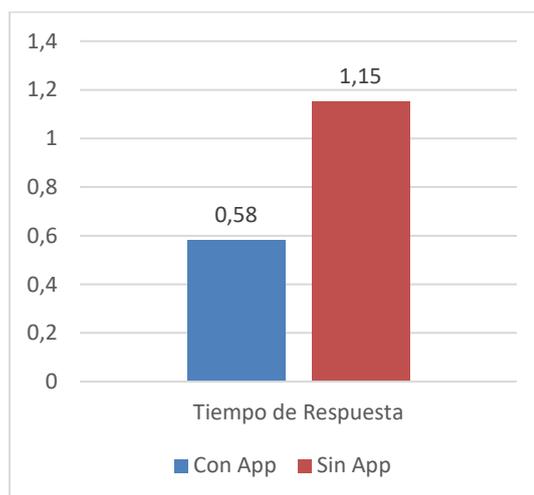


Gráfico 3-4: Comparación entre tiempos de respuesta con y sin la aplicación RespedApp

Realizado por: Autoría propia Ordoñez (2022), Chimbo (2022).

4.1.2 *Rendimiento*

Para calcular esta métrica se tomará en cuenta el número de pedidos y reservas registrados con

respecto a la suma de los tiempos de respuesta, es decir el tiempo total de registro de todas las reservas y pedidos de la muestra estudiada. El número de pedidos y reservas es 41, el tiempo total sin la aplicación es de 47.08 minutos y el tiempo total con la aplicación es de 23.70 minutos. El resultado de rendimiento sin la aplicación es de 0.87 y con la aplicación de 1.73 como se visualiza en el **gráfico 4-4**.

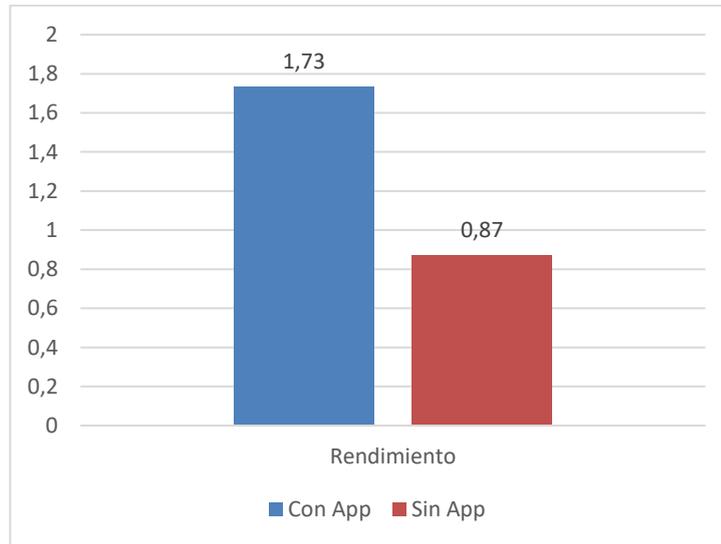


Gráfico 4-4: Comparación entre el rendimiento con y sin la aplicación RespedApp

Realizado por: Autoría propia Ordoñez (2022), Chimbo (2022).

4.1.3 Matriz de medición del comportamiento temporal

A continuación, en la **tabla 13-4** se describirá los resultados obtenidos al evaluar la subcaracterística de comportamiento temporal según la ISO/IEC 25000:

Tabla 13-4: Matriz de medición de calidad de la subcaracterística eficiencia según la ISO/IEC 25000 en la aplicación RespedApp

Subcaracterística	Métrica	Propósito de la Métrica	Método de aplicación	Fórmula	Valor Deseado	Valor Obtenido	Aplicación	Ponderación
Comportamiento en el tiempo	Tiempo de respuesta	¿Cuál es el tiempo estimado en realizar el registro de un pedido	Tomar tiempo desde que el cliente visualiza el menú hasta que el pedido es recibido en cocina.	$X = A - B$ A= Tiempo final del proceso registro B= Tiempo inicial	$X < 1.15$ minutos	X=0.58 minutos	Si	1.5

		y/o reserva?	Tomar tiempo desde que el cliente se contacta con el restaurante hasta que la reserva es recibida en administración.	del proceso registro B=0				
	Rendimiento	¿Cuántos registros de reservas y pedidos se pueden realizar en un periodo de tiempo?	Realizar un determinado número de pedidos y reservas y calcular el tiempo total a través de sus tiempos individuales	$X = A/T$ A = Número de pedidos y reservas registradas T = Tiempo total Dónde: A > 0	$X > 0.8$ 7	X=41/23. 70 X=1.73	Si	1.5

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

Como resultado se ha obtenido una calificación de 3 para la subcaracterística de comportamiento en el tiempo.

4.2 Resultado de métricas de disponibilidad

Para poder medir la disponibilidad de la aplicación RespedApp se consideró aplicar una prueba de ping en *Application Insights* de Azure con una frecuencia de cinco minutos durante siete días en la localidad donde se aloja la aplicación que es *Central US*. Cabe recalcar que el proveedor de servicio donde se aloja la aplicación (Digital Ocean) brinda un 99.99% de disponibilidad al año. Esto implica que por defecto la aplicación no estará disponible alrededor de 5256 minutos anualmente. En la semana en la que se hizo la toma de resultados este factor no influyó en la

disponibilidad de la aplicación por lo que será descartar. Un factor que si influyó en la disponibilidad de la aplicación RespedApp es la ejecución de despliegues, en función de lograr mejoras en la aplicación.

Fecha de Inicio: 05/07/2022 11:00 am

Fecha de Fin: 12/07/2022 11:00 am

4.2.1 Tiempo de Servicio

Para calcular la métrica de tiempo de servicio se debe considerar el tiempo de servicio de la aplicación sobre el tiempo regulado con el cliente en el SLA (Acuerdo de nivel de servicio). El tiempo de servicio obtenido a través de Azure Application Insights es del 99.95% de 10080 minutos que resulta en 10075 minutos como se lo observa en la **figura 16-4**. El tiempo regulado con el cliente es de 168 horas que equivale a 10080 minutos, este se lo puede observar en el Anexo C.

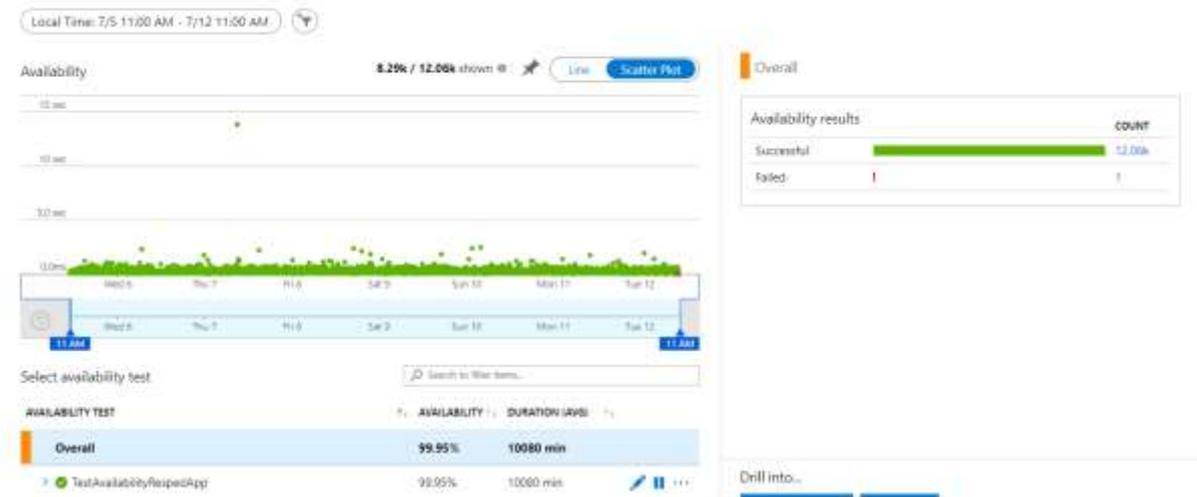


Figura 16-4: Resultado de tiempo de servicio de la aplicación RespedApp en Azure Application Insights

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

4.2.2 Tiempo medio de inactividad

El tiempo medio de inactividad se ha calculado a través del tiempo total de inactividad sobre el número de apagones ocurridos. En la **figura 17-4** se puede visualizar el tiempo total de inactividad (*downtime*) de 5 minutos y el número de fallos o apagones (*outages*) que resultó en 1. Este fallo se debió a la realización de un cambio en la aplicación en el tiempo de pruebas. El despliegue tuvo una duración de 5 minutos.



Figura 17-4: Resultado de tiempo medio de inactividad de la aplicación RespedApp en Azure Application Insights

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

4.2.3 Matriz de medición de la disponibilidad

A continuación, en la **tabla 14-4** se describirá los resultados obtenidos al evaluar la subcaracterística de disponibilidad según la ISO/IEC 25000:

Tabla 14-4: Matriz de medición de calidad de la subcaracterística disponibilidad según la ISO/IEC 25000 en la aplicación RespedApp

Subcaracterística	Métrica	Propósito de la Métrica	Método de aplicación	Fórmula	Valor Deseado	Valor Obtenido	Aplica	Ponderación
Disponibilidad	Tiempo de servicio	¿Cuál es el tiempo de servicio de la aplicación que proporciona realmente?	Realizar el acuerdo de nivel de servicio con el cliente. Tomar tiempo de servicio a través de Azure Application Insights.	$X = A/B$ $A =$ Tiempo de servicio del aplicativo que se proporciona actualmente $B =$ Tiempo de servicio del aplicativo regulado con el cliente Dónde: $B > 0$	$X >= 1$ Mientras más cercano a 1 mejor.	$X = 10075$ min/ 10080 min $X = 0.995$	Si	6.99
	Tiempo medio de inactividad	¿Cuál es el tiempo promedio que la aplicación	Tomar tiempo de inactividad a través de Azure	$X = T/A$ $A =$ Número de fallos	$X = 0$ Mientras más cercano a 0 sea	$X = 5$ min/ 1 $X = 5$ min	No	N/A

		está inactiva cuando ocurre un fallo?	Application Insights. Tomar número de fallos a través de Azure Application Insights	observados T = Tiempo total de inactividad Dónde: $A > 0$	mejor			
--	--	---------------------------------------	-------------------------------------------------------------------------------------	-----------------------------------------------------------	-------	--	--	--

Realizado por: Chimbo Encalada, Emily; Ordoñez Zapata, Angello, 2022.

Como resultado se ha obtenido una calificación de 6.99 para la subcaracterística de disponibilidad.

4.3 Resultados de la evaluación de la calidad de la aplicación RespedApp

Una vez realizada la evaluación de las métricas que determinan la calidad de la aplicación web progresiva “RespedApp” como se puede visualizar en la **tabla 15-4** con un puntaje de 9.99/10 se determina que la aplicación cumple con los requisitos solicitados, se rechaza la hipótesis nula y se acepta la hipótesis alterna que prueba que la aplicación web progresiva RespedApp tendrá un grado de calidad mayor o igual a 8.76.

Tabla 15-4: Matriz de resultados de calidad de la aplicación RespedApp

Característica	Métrica	Valor esperado	Valor obtenido	Porcentaje de Cumplimiento	Ponderación
Disponibilidad	Tiempo de servicio	1	0.99	99.95%	6.99
Eficiencia	Tiempo de respuesta	1.15	0.58	150.43%	1.5
	Rendimiento	0.87	1.63	187.35%	1.5
Total, puntaje de calidad					9.99

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

CONCLUSIONES

En el presente trabajo de titulación se ha desarrollado e implementado la aplicación web progresiva “RespedApp” la cual ha ayudado a la automatización y gestión del proceso de pedidos y reservas en el restaurante “El fogón de Coz”, utilizando los criterios de calidad de eficiencia y disponibilidad. A continuación, se describen una serie de conclusiones con respecto a los objetivos específicos definidos en el presente documento.

- Una aplicación web progresiva junto al uso de la técnica de renderizando por parte del servidor brinda características como lo son la alta disponibilidad, un mejor rendimiento, una carga inicial rápida y tiempos de carga reducidos.
- Mediante la interacción con el *product owner* se pudo definir el proceso de reservas y pedidos, para el primer proceso de registro de una reserva se incluye 2 actores con 3 actividades y para realizar el registro de un pedido están involucrados 4 actores con 8 actividades.
- Para el desarrollo de la aplicación RespedApp se aplicó la metodología ágil Scrumban, el desarrollo del proyecto tuvo una duración de 15 semanas de trabajo con 4 sprints de 4 semanas cada uno, se utilizó artefactos como el *product* y *sprint backlog*, junto a un tablero Kanban con 4 estados. Se trabajó 8 horas diarias para cumplir con la planificación, y de esta forma se permitió construir una aplicación que cumple satisfactoriamente con las necesidades del usuario.
- Para finalizar, mediante la evaluación estadística de la calidad del producto de software y el uso de la norma ISO/IEC 25010, se determinó que RespedApp obtuvo una puntuación de 3.00 de eficiencia y 6.99 de disponibilidad, dando un total de 9.99 de calidad en relación con las subcaracterísticas mencionadas, es decir que cumple con el grado de calidad mayor a 8.76 que era el esperado por parte del cliente.

RECOMENDACIONES

- Desplegar la aplicación web progresiva en la nube para lograr una mayor escalabilidad, flexibilidad, y obtener una gran cantidad de recursos a un costo bajo.
- Durante el proceso de investigación bibliográfica se descubrió otras técnicas de renderizado para la web, por lo que se recomienda investigar las técnicas de Static Side Generation (SSG) o Incremental Static Regeneration (ISR), las cuales pretenden mejorar los tiempos de renderizado de la web.
- Es recomendable utilizar el framework NextJs que tiene soporte para aplicaciones web progresivas con SSR. Además, también se debe utilizar buenas prácticas de programación para obtener un producto de software de calidad.
- Para obtener una mejor evaluación de la calidad de la aplicación web progresiva se recomienda evaluar cada una de las métricas de la norma ISO/IEC 25010.

•

BIBLIOGRAFÍA

ASPERBROTHERS, *Server-Side Rendering in React using Next.js – How it Works & Implementation Example* | ASPER BROTHERS. [en línea]. 2021. [Consulta: 27 abril 2022]. Disponible en: <https://asperbrothers.com/blog/server-side-rendering-in-react/#>.

BALSECA CHISAGUANO, E.A., Evaluación de calidad de productos de software en empresas de desarrollo de software aplicando la norma ISO/IEC 25000. [en línea], 2014. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/9113>.

BARALENT LAB, *Ventajas de tener un sistema de reservas online*. 2021. 2021. S.l.: s.n.

BEKE, M., *On the Comparison of Software Quality Attributes for Client-side and Server-side Rendering* [en línea]. 2018. S.l.: Universiteit Antwerpen. Disponible en: <https://denbeke.be/thesis/versions/mathias-beke-final.pdf>.

BRYANT ALEJANDRO, *Server Side Rendering y las PWAs PWA Experts I/O*. 2020. 2020. S.l.: s.n.

CAHUEÑAS, I.V., *Calidad del producto en el desarrollo de software en la Universidad Israel aplicando normas ISO*. 2017. S.l.: s.n.

CALLEJAS-CUERVO, M., ALARCÓN-ALDANA, A.C. y ÁLVAREZ-CARREÑO, A.M., Modelos de calidad del software, un estado del arte. *ENTRAMADO*, 2017. vol. 13, no. 1, pp. 236-250. ISSN 19003803. DOI 10.18041/ENTRAMADO.2017V13N1.25125.

CARNEIRO, J., *Progressive Web Apps: Concepts and Features*. , 2018. ISSN 2184-223X.

CRESPO, A., *ISO 25000: La calidad del producto software*. [en línea]. 2018. [Consulta: 25 abril 2022]. Disponible en: <https://www.excentia.es/iso-25000>.

DB TECHNOLOGIES, *ANALISIS COMPARATIVO DE MODELOS DE CALIDAD EN EL SOFTWARE*. [en línea]. 2012. [Consulta: 26 abril 2022]. Disponible en: <https://cristinadavila.wordpress.com/2012/05/13/analisis-comparativo-de-modelos-de-calidad-en-el-software/>.

DEPT. CIENCIA DE LA COMPUTACIÓN, *Introducción a los Servicios Web RESTful Índice*. , 2013.

FADHILAH ISKANDAR, T., LUBIS, M., FABRIANTI KUSUMASARI, T. y RIDHO LUBIS, A., Comparison between client-side and server-side rendering in the web development. *IOP Conference Series: Materials Science and Engineering*, 2020. vol. 801, no. 1. ISSN 1757899X. DOI 10.1088/1757-899X/801/1/012136.

FRÍAS, J., *Las aplicaciones móviles y el comportamiento del consumidor joven burgalés*. 2019. S.l.: Universidad de Burgos.

GAUNT, M., *Introducción a los service workers Web Fundamentals*. 2019. 2019. S.l.: s.n.

INSTITUTO ECUATORIANO DE NORMALIZACIÓN, NTE-INEN ISO/IEC:25010 -

Sistemas e Ingeniería de Software - Requerimientos y Evaluación de sistemas y calidad de software (SQUARE) - Modelos de calidad del sistema y software (ISO/IEC 25010:2011, IDT). , 2011.

ISO 25000, *NORMAS ISO 25000*. [en línea]. 2022. [Consulta: 25 abril 2022]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000>.

ISO/IEC 25040, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. 2011. 2011. S.l.: s.n.

JESUÏTES EDUCACIÓ, *¿Qué tipos de aplicaciones web existen?* 2017. abril 2017. S.l.: s.n.

LAURA DÍAZ, B., URI TORRUCO, G., MILDRED MARTÍNEZ, H. y RUIZ, M.V., La Entrevista, recurso flexible y dinámico. *Investigación en educación médica*, 2013. vol. 2, no. 7, pp. 162-167. ISSN 2007-5057. DOI 10.1016/s2007-5057(13)72706-6.

LAVU, *Sistema de pedidos en línea para restaurantes: La guía definitiva*. 2020. diciembre 2020. S.l.: s.n.

LÓPEZ MAGAÑA, L.M., *Qué es Json Web Token y cómo funciona*. [en línea]. 2020. [Consulta: 11 junio 2022]. Disponible en: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>.

LUJÁN-MORA, S., Programación de aplicaciones web: historia, principios básicos y clientes web. , 2002.

LUNA, F.O., *PWA - Desarrolla Aplicaciones Web Multidispositivos*. 2021a. s.n. S.l.: ISBN 978-987-8414-14-0.

LUNA, F.O., *PWA - Desarrolla Aplicaciones Web Multidispositivos: Potencia tus aplicaciones web progresivas*. 2021b. s.n. S.l.: ISBN 978-987-8414-28-7.

MACORETTA, L., *Por qué usar Next JS en 2021. Hola! Hoy les traigo un artículo sobre... | by Leandro Macoretta | Medium*. [en línea]. 2021. [Consulta: 27 abril 2022]. Disponible en: <https://leandrom-36593.medium.com/por-qué-usar-next-js-en-2021-d324278856fc>.

MDN CONTRIBUTORS, *HTML: Lenguaje de etiquetas de hipertexto MDN*. 2021. 2021. S.l.: s.n.

MDN CONTRIBUTORS, *Service Worker API - Referencia de la API Web*. 2022. 2022. S.l.: s.n.

MONTOYA, K.C. y SANCHEZ, J.S., *DESARROLLO DE SISTEMA WEB Y APLICACIÓN MÓVIL PARA LA GESTIÓN DE PEDIDOS DE COMIDA EN EL RESTAURANTE. “RÔTI GRILLÉ”*. 2020. Quito: ESCUELA POLITÉCNICA NACIONAL.

NEXT, *What is Rendering? - How Next.js Works | Learn Next.js*. [en línea]. 2022. [Consulta: 26 abril 2022]. Disponible en: <https://nextjs.org/learn/foundations/how-nextjs-works/rendering>.

NUXT, *Nuxt - Server Side Rendering*. 2022. 2022. S.l.: s.n.

PATEL, J., *10 Best Front end Frameworks for Web Development in 2022*. [en línea]. 2021. [Consulta: 27 abril 2022]. Disponible en: <https://www.monocubed.com/blog/best-front-end->

PRERENDER, *What Is Server Side Rendering | SSR Pros & Cons | Prerender. [en línea]*. 2021. [Consulta: 20 abril 2022]. Disponible en: <https://prerender.io/what-is-srr-and-why-do-you-need-to-know/>.

PROGRAMADOR CLIC, *¿Cómo realizar la disponibilidad sin conexión en la arquitectura SSR? (Uno) . [en línea]*. 2022. [Consulta: 27 abril 2022]. Disponible en: <https://programmerclick.com/article/99121113002/>.

RODRÍGUEZ, R.A., VERA, P.M., MARTÍNEZ, M.R., BELTRÁN, F.A.P., TRIGUEROS, A. y DOGLIOTTI, M.G., *Aplicaciones Web Progresivas Impulsadas por el Avance de los Estándares Web. , 2019. pp. 5.*

SAKHI, S., *SSR Frameworks: Next.js vs SvelteKit vs Nuxt vs Quasar vs Gatsby | Simply How. [en línea]*. 2021. [Consulta: 27 abril 2022]. Disponible en: <https://simply-how.com/server-side-rendering-web-frameworks>.

SANTILLÁN MARFETAN, A.G. y GUSÑAY CELA, J.S., *Desarrollo de aplicación web de gestión y venta para restaurantes. Repositorio de la Universidad Estatal de Milagro, 2020.*

SCHNEIDER, A.H., *Desenvolvimento web com Client Side Rendering: combinando Single Page Application e serviços de backend. , 2016.*

TAHIRSHAH, F.S., *Comparison between Progressive Web App and Regular Web App. , 2019.*

TELLO, P., *Evaluación de Calidad de un Producto de Software. Laboratorio Nacional de Calidad del Software de INTECO - Abril, 2016, 2016. vol. 1, pp. 132.*

THE ECONOMIST, *The ineluctable middlemen | . Business. 2012. 2012.*

TTANDEM, *¿Cuáles son las ventajas y desventajas de las PWA? . [en línea]*. 2022. [Consulta: 20 abril 2022]. Disponible en: <https://www.ttandem.com/blog/que-son-las-pwa-o-progressive-web-applications/ventajas-y-desventajas-de-las-pwa/>.

VERCEL, *Next.js by Vercel - The React Framework. 2021. 2021. S.l.: s.n.*

WESTREICHER, G., *Hipótesis - Qué es, definición y concepto | 2022 | Economipedia. [en línea]*. 2022. [Consulta: 5 junio 2022]. Disponible en: <https://economipedia.com/definiciones/hipotesis.html>.

YUMI GUAMÁN, G.Rocío. y MOINA ZABALA, T.C., *Desarrollo de una Aplicación Web para el Registro de la Distribución de la Jornada Laboral del Personal Académico de la Facultad de Informática y Electrónica. 2016. Riobamba: Escuela Superior Politécnica de Chimborazo.*



DBRAI

Ing. Cristian Castillo



ANEXOS

Los Anexos del presente Trabajo de Titulación se pueden visualizar en el siguiente link:

<https://drive.google.com/drive/u/0/folders/1p9hwdz9O7vSW0dwDx3YBZNbIbRZoYDwW>

ANEXO A: ENTREVISTA

ANEXO A. ENTREVISTA AL CLIENTE

Entrevista sobre los procesos de gestión de reservas y pedidos del restaurante "El fogón de Coz"

Nombre del entrevistado:	Chef Alejandro Cabrera
Nombre del entrevistador:	Emily Chimbo
Fecha:	21 de Marzo del 2022
Hora de inicio:	15:00
Hora de Finalización:	16:00

Preguntas

1. ¿Cómo se realizan actualmente los pedidos en el restaurante?

Respuesta:

Para realizar un pedido en el restaurante "El fogón de Coz" primero el cliente ve el menú ya sea en el local o a través de nuestras redes sociales, después realiza el pedido escribiéndonos o llamándonos o el mesero toma su pedido, después eso se le pasa a cocina. Si es en el local el mesero le pasa a cocina, sino el administrador que es el que gestiona los mensajes y llamadas, luego ya cocina prepara todo y le entrega al mesero para que le deje en la mesa o sino al administrador para que gestione el envío. Y bueno, respecto al pago se hace al último, el cliente puede pagar con tarjeta o efectivo. Y ya el envío es aparte, nosotros no cobramos de eso.

2. ¿Cómo se realizan actualmente las reservas en el restaurante?

Respuesta:

El cliente suele llamar o escribir si no hay nadie en el local, entonces el pregunta si hay disponibilidad de lugar en la fecha y hora que quiere y si es afirmativo, se le pregunta si desea incluir ya el pedido para prepararlo con anticipación o si no lo desea, solo se registra la reserva. En el caso de que no haya plaza disponible entonces se le informa al cliente.

3. ¿Cuál es el resultado que desea obtener de la PWA RespedApp?

Respuesta:

Se desea superar los niveles de calidad en eficiencia y disponibilidad. Es decir, tener una aplicación de calidad que sirva para registrar pedidos y reservas a través de la web. Considero la disponibilidad como una característica muy relevante debido a que me gustaría satisfacer a mis clientes en casi todo momento y actualmente no puedo permitirme abrir el local las 24 horas del día. Además, solo cuento con dos meseros y la cantidad de pedidos sobrepasa a la atención que les puedo brindar. Me gustaría receptor los pedidos de manera más rápida, así también receptor más pedidos en menos tiempo. Consideraré en un 70% del valor total de calidad a la disponibilidad y un 30% a la eficiencia, siendo la primera característica la más relevante para mi negocio.

4. ¿Qué módulos deben constar en la aplicación?

ANEXO A, ENTREVISTA AL CLIENTE

Respuesta:

Para conseguir la gestión de reservas y pedidos necesito que la aplicación tenga un módulo de autenticación para clientes, meseros y administradores, la gestión de Menú, la gestión de Usuarios, la gestión de Pedidos, un visor de pedidos, la gestión de Reservas, un módulo para pagos y los módulos ya de pedidos, reservas, el inicio y el menú.

5. ¿Qué otras características deberían incluirse en la aplicación?

Respuesta:

Me gustaría que utilizaran la paleta de colores que se usa en el restaurante, el logo del lugar, una tipografía no llamativa y que la aplicación se pueda utilizar en cualquier dispositivo.

Yo, Alejandro Cabrera certifico que la información obtenida de la presente entrevista es verdadera y permito su utilización para los fines pertinentes con respecto al desarrollo de la aplicación RespedApp.



Chef Alejandro Cabrera

Propietario del restaurante "El fogón de Cor"

ANEXO B: MANUAL TÉCNICO DE RESPEDAPP

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

MANUAL TÉCNICO

DESARROLLO DE UNA APLICACIÓN WEB PROGRESIVA PARA LA GESTIÓN DE RESERVA Y PEDIDOS DEL RESTAURANTE “EL FOGÓN DE COZ” IMPLEMENTANDO SSR

FECHA	VERSIÓN	DESCRIPCIÓN	AUTOR
30/04/2022	1.0	Primera versión manual técnico de software	Angello Rafael Ordoñez Zapata. Estudiante de Ingeniería de Software Emily Micaela Chimbo Encalada. Estudiante de Ingeniería de Software
07/06/2022	1.1	Integración de la base de datos	Angello Rafael Ordoñez Zapata.

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

FECHA	VERSIÓN	DESCRIPCIÓN	AUTOR
			Estudiante de Ingeniería de Software Emily Micaela Chimbo Encalada. Estudiante de Ingeniería de Software
07/06/2022	1.2	Agregar Product Backlog	Angello Rafael Ordoñez Zapata. Estudiante de Ingeniería de Software Emily Micaela Chimbo Encalada. Estudiante de Ingeniería de Software
18/07/2022	1.3	Inclusión de Retrospectiva	Angello Rafael Ordoñez Zapata. Estudiante de Ingeniería de Software Emily Micaela Chimbo Encalada. Estudiante de Ingeniería de Software

TABLA DE CONTENIDO

TABLA DE CONTENIDO	3
ÍNDICE DE TABLAS	6
ÍNDICE DE ILUSTRACIONES	9
INTRODUCCIÓN	12
OBJETIVOS	13
Objetivo general	13
Objetivos específicos	13
1 ESTUDIO DE FACTIBILIDAD	14
1.1 Alcance del estudio de factibilidad	14
1.2 Factibilidad Técnica	14
1.2.1 Proveedores de servicios de nube	14
1.2.2 Lenguaje de Desarrollo	15
1.2.3 Sistema Gestor de Base de Datos	16
1.2.4 Características del Hardware Disponible para el Desarrollo	16
1.2.5 Experiencia y Conocimiento del Equipo de Desarrollo	17
1.2.6 Conclusión de Factibilidad Técnica	18
1.3 Factibilidad Operativa	18
1.3.1 Usuarios	18
1.3.2 Sistema Gestor de Base de Datos	19
1.3.3 Aplicación Web Progresiva	19
1.3.4 Hardware	20
1.3.5 Conclusión de Factibilidad Operativa	20
1.4 Estimación del Proyecto	20
2 ANÁLISIS	25
2.1 Identificación de Riesgos	26
2.1.1 Estimación de Probabilidad	26

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

2.1.2	Probabilidad de ocurrencia del riesgo	26
2.1.3	Estimación de impacto y ocurrencia de los riesgos.....	27
2.1.4	Exposición al riesgo.....	28
2.1.5	Matriz de mitigación de Riesgos	28
3	PLANIFICACIÓN DEL PROYECTO	29
3.1	Ámbito del Software.....	29
3.2	Recursos	30
3.2.1	Recursos Humanos	30
3.2.2	Recursos de Software, Hardware y Otros.....	30
3.2.3	Conclusión	31
3.3	Cronograma de Actividades.....	32
3.4	Historias de Usuario.....	32
3.5	Fuentes y Colores.....	34
4	IMPLEMENTACIÓN.....	35
4.1	Bosquejos de Interfaces	35
4.2	Herramientas de Software.....	47
4.3	Base de datos.....	47
4.3.1	Diagrama entidad relación.....	47
4.3.2	Normalización de datos	48
4.3.3	Diagrama Lógico	61
4.3.4	Diagrama físico.....	62
4.3.5	Gestor de Base de Datos.....	63
4.3.6	Diccionario de datos.....	63
4.3.7	Estimación de Base de Datos.....	70
4.4	Product Backlog.....	76
4.5	Sprint Backlog	78
4.5.1	Sprint I	79

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

4.5.2	Sprint 2	79
4.5.3	Sprint 3	80
4.5.4	Sprint 4	80
5	REVISIÓN Y RETROSPECTIVA	81
5.1	Burndown Chart.....	83
6	LANZAMIENTO Y CIERRE.....	84
6.1	Entregables del Sprint 1	85
6.2	Entregables del Sprint 2	86
6.3	Entregables del Sprint 3	86
6.4	Entregables del Sprint 4	86

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Cronograma del proyecto en Project Libre.....	32
Ilustración 2: Tipografía requeridos en el proyecto RESPEDAPP.....	34
Ilustración 3: Colores HTML requeridos en el proyecto RESPEDAPP.....	35
Ilustración 4: Bosquejo de la sección navbar y header de la landing page del restaurante “El fogón de Coz”.....	36
Ilustración 5: Bosquejo de la sección reservaciones de la landing page del restaurante “El fogón de Coz”.....	36
Ilustración 6: Bosquejo de la sección galería de la landing page del restaurante “El fogón de Coz”.....	36
Ilustración 7: Bosquejo de la sección acerca de nosotros de la landing page del restaurante “El fogón de Coz”.....	37
Ilustración 8: Bosquejo de la sección contacto de la landing page del restaurante “El fogón de Coz”.....	37
Ilustración 9: Bosquejo de la sección footer de la landing page del restaurante “El fogón de Coz”.....	37
Ilustración 10: Bosquejo de la pantalla de Dashboard del restaurante “El fogón de Coz”.....	38
Ilustración 11: Bosquejo de la pantalla de Usuarios cerrada del restaurante “El fogón de Coz”.....	39
Ilustración 12: Bosquejo de la pantalla de Usuarios abierta del restaurante “El fogón de Coz”.....	39
Ilustración 13: Bosquejo de la pantalla de Pedidos cerrada del restaurante “El fogón de Coz”.....	40
Ilustración 14: Bosquejo de la pantalla de Pedidos abierta del restaurante “El fogón de Coz”.....	41
Ilustración 15: Bosquejo de la pantalla de Reservas cerrada del restaurante “El fogón de Coz”.....	42
Ilustración 16: Bosquejo de la pantalla de Reservas abierta del restaurante “El fogón de Coz”.....	43
Ilustración 17: Bosquejo de la pantalla de Productos cerrada del restaurante “El fogón de Coz”.....	44

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Ilustración 18: Bosquejo de la pantalla de Productos abierta del restaurante “El fogón de Coz”	44
Ilustración 19: Bosquejo de la pantalla de perfil cliente del restaurante “El fogón de Coz”	45
Ilustración 20: Bosquejo de la pantalla de cliente ordenes cerrada del restaurante “El fogón de Coz”	45
Ilustración 21: Bosquejo de la pantalla de cliente ordenes abierta del restaurante “El fogón de Coz”	46
Ilustración 22: Bosquejo de la pantalla de cliente direcciones del restaurante “El fogón de Coz”	46
Ilustración 23: Diagrama lógico de la base de datos de RESPEDAPP	62

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Índice de figuras

Figura 1: Diagrama entidad relación simplificado de la base de datos de RESPEDAPP	48
Figura 2: Entidad Categoría con tipos de datos.....	63
Figura 3: Relaciones de Entidad Categoría	63
Figura 4: Entidad Dirección con tipos de datos.....	64
Figura 5: Relaciones de Entidad Dirección.....	64
Figura 6: Entidad Estado de reservas con tipos de datos.....	64
Figura 7: Relaciones de Entidad Estado reservas.....	64
Figura 8: Entidad Estado con tipos de datos	65
Figura 9: Relaciones de Entidad Estado.....	65
Figura 10: Entidad Producto con tipos de datos.....	65
Figura 11: Relaciones de Entidad Producto	65
Figura 12: Entidad Reserva con tipos de datos	66
Figura 13: Relaciones de Entidad Reserva.....	66
Figura 14: Entidad Tipo Usuario con tipos de datos.....	66
Figura 15: Relaciones de Entidad Tipo Usuario	66
Figura 16: Entidad Usuario con tipos de datos	67
Figura 17: Relaciones de Entidad Usuario	67
Figura 18: Entidad Pedido Total con tipos de datos.....	67
Figura 19: Relaciones de Entidad Pedido Total.....	67
Figura 20: Entidad Pedido Domicilio con tipos de datos.....	68
Figura 21: Relaciones de Entidad Pedido Domicilio	68
Figura 22: Entidad Pedido Local con tipos de datos	68
Figura 23: Relaciones de Entidad Pedido Local	68
Figura 24: Entidad Reserva con tipos de datos	68
Figura 25: Relaciones de Entidad Reserva.....	69
Figura 26: Entidad Relación Pedido Producto con tipos de datos	69
Figura 27: Relaciones de Entidad Relación Pedido Producto.....	69
Figura 28: Entidad Relación usuario dirección con tipos de datos	69
Figura 29: Relaciones de Entidad Relación usuario dirección.....	70
Figura 30: Diagrama de quemado de la aplicación RespedApp	84
Figura 31: Diagrama de despliegue de la aplicación RespedApp.....	86

INTRODUCCIÓN

El presente documento servirá como una guía a cualquier persona que posea conocimientos relacionados con la carrera de ingeniería de software o afines pueda comprender a la brevedad el funcionamiento de la aplicación RESPEDAPP, la cual es una aplicación para la gestión de reservas y pedidos del restaurante “El Fogón de COZ” que será construida con el lenguaje de programación Javascript, para frontend se empleará el framework Next.js y para el backend se empleará el framework Express.js que se conectará a una base de datos MySQL con ayuda del ORM Sequelize, este manual técnico se dará a conocer los requerimientos mínimos necesarios de software y de hardware para el correcto funcionamiento del sistema dentro del servidor y del cliente, de igual forma la estructura con la que fue desarrollada el sistema y los procesos que se siguieron para la construcción.

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

OBJETIVOS

Objetivo general

Brindar la información necesaria para poder realizar el despliegue, configuración y mantenimiento del sistema RESPEDAPP en el servidor, así mismo la instalación del aplicativo en el cliente.

Objetivos específicos

- Elaborar el estudio de factibilidad, análisis de riesgos y planificación del aplicativo.
- Representar la funcionalidad técnica de la estructura, diseño y definición del aplicativo.
- Definir claramente el procedimiento de instalación del aplicativo.
- Detallar la especificación de los requerimientos de Hardware y Software necesarios para la instalación de la aplicación.
- Describir las herramientas utilizadas para el diseño y desarrollo del prototipo

1 ESTUDIO DE FACTIBILIDAD

Para que el proyecto tenga viabilidad es necesario determinar el estudio de factibilidad, el cual es uno de los análisis más importantes y esperados antes de comenzar un proyecto, ya que nos ayudará a determinar cómo lo dice su nombre la factibilidad del proyecto. Para ello, este estudio se centrará en dos fases: técnico y operativo.

1.1 Alcance del estudio de factibilidad

Con ayuda del estudio de factibilidad que se realizará, se busca determinar si el desarrollo de una aplicación web progresiva para la gestión de reservas y pedidos del restaurante “El Fogón de COZ” implementando SSR cumple con los requerimientos para iniciar el proceso de desarrollo, donde los puntos analizados serán el factor técnico y operativo, que es lo mínimo que debe cumplir para que el proyecto sea viable.

1.2 Factibilidad Técnica

Con ayuda de la factibilidad técnica podemos establecer si el proyecto a construir puede desarrollarse con los recursos técnicos con que se cuenta dentro del equipo de desarrollo; para esto se ha considerado los aspectos de hardware, software, factor humano dentro del cual se considera, la existencia de la tecnología y el conocimiento necesario para establecer la factibilidad técnica de la construcción del proyecto.

1.2.1 Proveedores de servicios de nube

Los proveedores de servicios de nube son empresas que poseen servidores generalmente por todo el mundo, con el fin de ofrecer los servicios de nubes públicas, privadas, híbridas, además, también ofrecen servicios de infraestructura tales como Infraestructura como servicio (IaaS), Plataforma como servicio (PaaS), Software como servicio (SaaS). Si bien hay muchas empresas que ofrecen estos servicios entre las más conocidas y populares son: Alibaba Cloud, Amazon Web Services (AWS), Google Cloud Platform (GCP), IBM Cloud, Oracle Cloud, Microsoft Azure, claro que no son los únicos, sino que también existen otros proveedores en todo el mundo como: DigitalOcean, Heroku, CloudFlare, entre otros.

Para este proyecto el cual debe ser alojado en un servicio de nube se ha considerado los proveedores: AWS, Azure, GCP y DigitalOcean para lo cual se ha hecho una comparativa

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

con relación a los precios y prestaciones ofrecidas las cuales podemos observar en la Tabla 1 a continuación.

Proveedor	CPU	Memoria	Almacenamiento	Ancho de banda	Precio
DigitalOcean	1	1 GB	25 GB	1 TB	\$ 5
	2	2 GB	60 GB	3 TB	\$ 15
	4	8 GB	160 GB	5 TB	\$ 40
	8	16 GB	320 GB	6 TB	\$ 80
Google Cloud Platform (GCP)	1	1 GB	25 GB	1 TB	\$ 6.11
	2	2 GB	60 GB	3 TB	\$ 12
	4	8 GB	160 GB	5 TB	\$ 111
	8	16 GB	320 GB	6 TB	\$ 221
Amazon Web Services (AWS)	1	1 GB	25 GB	1 TB	\$ 8.50
	2	2 GB	60 GB	3 TB	\$ 15.23
	4	8 GB	160 GB	5 TB	\$ 60.91
	8	16 GB	320 GB	6 TB	\$ 121.81
Microsoft Azure	1	1 GB	25 GB	1 TB	\$ 7.59
	2	2 GB	60 GB	3 TB	\$ 15.11
	4	8 GB	160 GB	5 TB	\$ 60.74
	8	16 GB	320 GB	6 TB	\$ 121.18

Tabla 1: Comparación de precios de los proveedores de nube seleccionados

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

1.2.2 Lenguaje de Desarrollo

El lenguaje de desarrollo del proyecto debe cumplir con las siguientes características mínimas:

- Amplia documentación
- Soporte a bases de datos relacionales
- Facilidad de desarrollo en distintas arquitecturas
- Estable y ampliamente usado en ambiente web
- Que se pueda realizar server side rendering

Una vez analizado las características que debe cumplir se ha encontrado que los siguientes lenguajes cumplen con los requerimientos:

- Java
- PHP
- ASP.NET
- JavaScript

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

1.2.3 Sistema Gestor de Base de Datos

Este es un punto muy importante para tomar en cuenta ya que se determinará la forma en que se almacenará la información, la velocidad de procesamiento, respaldo de los datos y la seguridad. El sistema gestor de base de datos debe cumplir con las siguientes características:

- Estable
- Seguro
- Escalable
- Soporte de grandes cantidades de información
- Conexión con diferentes lenguajes de programación vía ODBC

A continuación, se presentan bases de datos que cumplen con las características mencionadas en el punto anterior

- MySQL
- MariaDB
- SQLite

1.2.4 Características del Hardware Disponible para el Desarrollo

Las características de los equipos que se dispone actualmente para el desarrollo del sistema de gestión de reservas y pedidos son los mostrados a continuación:

Equipo	Marca	Componentes	Características
Laptop 1	Dell Inspiron 14 3493	Procesador	Intel(R) Core (TM) i5-1035G4 CPU @ 1.10GHz - 1.50 GHz
		Memoria RAM	12 GB
		Capacidad de disco	Acer SSD NVMe Express FA100 512GB / TOSHIBA MQ01ABD100V 1000 GB
		Pantalla	14.0 pulgadas
		Sistema Operativo	Windows 10 Home
		Tarjeta Gráfica	S/N
Laptop 2	ASUS TUF	Procesador	Intel(R) Core (TM) i7-9750H CPU @2.60GHz
		Memoria RAM	32 GB
		Capacidad de disco	1 TB SSD Boot
		Pantalla	15.6 pulgadas
		Sistema Operativo	Windows 10 pro

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Equipo	Marca	Componentes	Características
		Tarjeta Gráfica	NVIDIA GTX 1650

Tabla 2: Descripción detallada de las computadoras disponibles para el desarrollo

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

1.2.5 Experiencia y Conocimiento del Equipo de Desarrollo

El recurso humano es el factor que estará encargado de realizar el desarrollo y el seguimiento del proyecto, es por ello la experiencia y el conocimiento del equipo el cual desarrollara el proyecto, para este proyecto el personal involucrado es el siguiente:

Recurso Humano	Docente director de proyecto
	Docente Asesor de proyecto
	Docente de materia de sistemas de titulación
	Dos estudiantes de la carrera de Ing. Software
	Representante de la empresa
Experiencia	Administración de proyectos informáticos.
	Desarrollo de Sistemas.
	Aplicación de metodologías ágiles.
	Trabajo en equipo.
Conocimientos	Lenguaje de programación Javascript.
	Framework de desarrollo frontend Next.js.
	Framework de desarrollo backend Express.js.
	Manejador de Base de Datos SQL.
	Base de Datos Relacionales.
Unified Modeling Language UML.	

Tabla 3: Recurso humano disponible para el desarrollo del proyecto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

En la Tabla 3 observada se detalla el recurso humano del que se dispone para el desarrollo del proyecto, el Docente Director sirve como guía en cada una de las etapas en que está dividido el proyecto, el Docente Asesor brinda sugerencias y opiniones que faciliten el continuo avance en el proyecto, el Docente encargado de la materia ayuda con revisiones periódicas y de los avances del proyecto y los dos integrantes del trabajo de graduación se encargan del análisis, diseño y programación del Sistema.

Por lo tanto, con base a lo anterior podemos decir que se dispone de recurso humano calificado, capaz de llevar a cabo el desarrollo del proyecto, también poseen el

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

conocimiento y las capacidades necesarias para cumplir con los requisitos y concluir con éxito dicho proyecto.

1.2.6 Conclusión de Factibilidad Técnica

En conclusión, una vez analizado el factor técnico se puede determinar que se cuenta con el equipo necesario para el desarrollo de una Aplicación web progresiva para la gestión de reservas y pedidos con implementación de server side rendering ya que se posee tanto el hardware como el software, así mismo el equipo de desarrollo está capacitado ya que poseen los conocimientos y experiencia necesarios para que el desarrollo de cada una de las etapas planificadas que se realizaran en su debido tiempo, por lo cual es seguro decir que el proyecto es técnicamente factible.

1.3 Factibilidad Operativa

El dueño y administrador del restaurante “El Fogón de COZ” está consciente de que en la actualidad es necesario poder manejar el tema de reservas y pedidos de forma que el usuario pueda acceder cuando lo requiera, además, de que es necesario optimizar los procesos de reservas y pedidos, ya que cada vez más los clientes lo realizan directamente desde la facilidad de su teléfono.

Además, el personal encargado de servir y atender a los clientes cree que es más factible poder realizar los pedidos desde un smartphone y estos sean comunicados directamente en el área de preparación de alimentos, ya que esto disminuye el tiempo que le tomaría ir a notificar en cocina. Es por ello por lo que el sistema toma en cuenta los siguientes aspectos.

- Usuarios
- Gestor de Base de datos
- Aplicación web progresiva
- Hardware

1.3.1 Usuarios

Se ha agrupado en conjunto a toda persona que va a interactuar de forma directa con el sistema final ya sean administradores, cocineros, meseros y clientes, aunque no se ha determinado los clientes.

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Requerido	Disponible
1 cocinero	1 cocinero
1 administrador	1 administrador
2 meseros	4 meseros
Clientes	Clientes

Tabla 4: Usuarios destinados a operar el sistema propuesto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

1.3.2 Sistema Gestor de Base de Datos

Un sistema de gestor de base de datos (SGBD) o Database Management System (DBMS) es un programa que se encarga de administrar y gestionar los datos para convertir en información, en otras palabras, se utiliza para controlar operaciones DDL y DML lo que permite administrar una base de datos. Para el proyecto es necesario un SGBD que permita administrar base de datos relacionales y que posea licencia tipo GPL.

Requerido	Disponible
Sistema de base de datos MariaDB	Sistema de base de datos MariaDB

Tabla 5: Gestor de Base de Datos para la operación del sistema propuesto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

1.3.3 Aplicación Web Progresiva

La aplicación web progresiva le da al usuario la facilidad de instalar la aplicación web en una computadora o un teléfono, lo cual le brinda la disponibilidad, además, también facilita el trabajo de registrar, administrar, editar las reservas y pedidos, además hay la posibilidad de acceder a través de internet, otra ventaja es que no necesita actualizar el software en cada dispositivo en que se utiliza, ya sea para los clientes como para los administradores. Para ellos es necesario especificar la tecnología adecuada para el desarrollo de los cuales ya se definieron en la factibilidad técnica, por lo cual se determinó que se usara el lenguaje de Javascript para el cliente y para el servidor.

Requerido	Disponible
Node.js v10.13.0	Node.js v18.0.0
Express.js v3.0	Express.js v4.0
Javascript ES5 ECMAScript 5	Javascript ECMAScript 2018
React.js v15.6	React.js v18.0.0
Next.js v10.0.0	Next.js v12.1

Tabla 6: Tecnologías Web para la operación del sistema propuesto

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

1.3.4 Hardware

Para que el sistema de gestión de reservas y pedidos funcione de manera correcta y eficiente es necesario tener los siguientes elementos.

Requerido	Disponible
1 servidor (1 CPU, 1GB Memoria, 25GB de almacenamiento)	1 servidor (1 CPU, 1GB Memoria, 25GB de almacenamiento)
1 computadora personal	1 computadora personal
1 monitor	-
1 impresora	1 impresora
2 Smartphone	4 Smartphone

Tabla 7: Evaluación de hardware disponible

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

1.3.5 Conclusión de Factibilidad Operativa

Se determinó que operativamente el desarrollo del proyecto es factible, ya que el restaurante "El Fogón de COZ" cuenta con la tecnología informática necesaria para la puesta en operación del sistema de información propuesto, además se debe añadir que en el caso del servidor el restaurante tiene la capacidad de adquirir los servicios de un proveedor externo.

1.4 Estimación del Proyecto

El restaurante "El fogón de Coz" requiere de un software para la gestión de reservas y pedidos. El proyecto se desarrollará con Visual Studio Code, con el lenguaje de programación JavaScript y se requiere la estimación de esfuerzo en meses necesarios para completar el desarrollo del proyecto.

Tiempo requerido de entrega: 4 meses

Para estimar el tiempo y personal requerido se utilizará el modelo COCOMO en su modelo intermedio. Para ello se calculará los siguientes datos:

- Esfuerzo
- Tiempo de Duración del proyecto
- Personal Necesario

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

COCOMO basa su estimación en el cálculo de líneas de código de un programa fuente, estas son dadas en Kilo Líneas de Código (KLC).

En la siguiente tabla de reconciliación de métricas LDC/PF podremos tomar el valor estimado para JavaScript.

189.	<i>GW BASIC</i>	98	238.	<i>LAMBIT/L</i>	64
190.	<i>Haskell</i>	38	239.	<i>Lattice C</i>	128
191.	<i>High C</i>	128	240.	<i>Liana</i>	128
192.	<i>HLEVEL</i>	58	241.	<i>LILITH</i>	71
193.	<i>HP BASIC</i>	128	242.	<i>LINC II</i>	14
194.	<i>HTML</i>	42	243.		
195.	<i>HTML 2.0</i>	16	244.	<i>LISP</i>	64
196.	<i>HTML 3.0</i>	15	245.	<i>LOGLISP</i>	58
197.	<i>Huron</i>	16	246.	<i>LOOPS</i>	21
198.	<i>IBM ADF I</i>	20	247.	<i>LOTUS 123 DOS</i>	6
199.	<i>IBM ADF II</i>	18	248.	<i>LOTUS Macros</i>	107
200.	<i>IBM Advanced BASIC</i>	98	249.	<i>LUCID 3D</i>	6
201.	<i>IBM CICS/VS</i>	40	250.	<i>LYRIC</i>	53
202.	<i>IBM Compiled BASIC</i>	91	251.	<i>M</i>	16
203.	<i>IBM VS COBOL</i>	107	252.	<i>macFORTH</i>	64
204.	<i>IBM VS COBOL II</i>	91	253.	<i>MACHI</i>	40
205.	<i>ICES</i>	71	254.	<i>Machine language</i>	640
206.	<i>ICON</i>	80	255.	<i>Macro assembly</i>	213
207.	<i>IDMS</i>	40	256.	<i>MAESTRO</i>	16
208.	<i>IEF</i>	14	257.	<i>MAGEC</i>	16
209.	<i>IEW</i>	14	258.	<i>MAGIK</i>	21
210.	<i>IFPS/PLUS</i>	32	259.	<i>MAKE</i>	21
211.	<i>IMPRS</i>	40	260.	<i>MANTIS</i>	40
212.	<i>INFORMIX</i>	40	261.	<i>MAPPER</i>	53
213.	<i>INGRES</i>	40	262.	<i>MARK IV</i>	40
214.	<i>INQUIRE</i>	13	263.	<i>MARK V</i>	36
215.	<i>INSIGHT2</i>	49	264.	<i>MATHCAD</i>	5
216.	<i>INSTALL/I</i>	16	265.	<i>MDL</i>	36
217.	<i>INTELLECT</i>	53	266.	<i>MENTOR</i>	53
218.	<i>INTERLISP</i>	58	267.	<i>MESA</i>	107
219.	<i>Interpreted BASIC</i>	107	268.	<i>Microfocus COBOL</i>	80
220.	<i>Interpreted C</i>	128	269.	<i>microFORTH</i>	64
221.	<i>IQLISP</i>	58	270.	<i>Microsoft C</i>	128
222.	<i>IQRP</i>	13	271.	<i>MicroStep</i>	20
223.	<i>JANUS</i>	71	272.	<i>Miranda</i>	40
224.	<i>JAVA</i>	53	273.	<i>Model 204</i>	38
225.	JAVASCRIPT	55	274.	<i>J2EE (JAVA)</i>	50
226.	<i>JSP</i>	59	275.		
227.	<i>JCL</i>	221	276.	<i>MODULA 2</i>	80
228.	<i>JOSS</i>	107	277.	<i>MOSAIC</i>	6

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

229.	<i>JOVLAL</i>	107	278.	<i>MS C ++ V. 7</i>	53
230.	<i>KAPPA</i>	40	279.	<i>MS Compiled BASIC</i>	91
231.	<i>KBMS</i>	49	280.	<i>MSL</i>	64
232.	<i>KCL</i>	64	281.	<i>muLISP</i>	64
233.	<i>KEE</i>	49	282.	<i>MUMPS</i>	19
234.	<i>Keyplus</i>	40	283.	<i>NASTRAN</i>	71
235.	<i>KL</i>	64	284.	<i>NATURAL 1</i>	53
236.	<i>KLO</i>	64	285.	<i>NATURAL 2</i>	46
237.	<i>KNOWOL</i>	49	286.	<i>NATURAL Construct</i>	25
238.	<i>KRL</i>	58	287.	<i>Natural language</i>	320 0
239.	<i>KSH</i>	21	288.	<i>NETRON/CAP</i>	19
240.	<i>Ladder Logic</i>	36	289.	<i>NEXPERT</i>	49

Tabla 8: Tabla de reconciliación de métricas

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Como el proyecto no es muy complejo, se ha determinado que será de tipo Orgánico. La siguiente tabla nos servirá de referencia para determinar el coeficiente de cálculo para cada modelo.

Proyecto	a	b	c	d
Software				
Orgánico	3.2	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

Tabla 9: Coeficientes de cálculo para modelo Cocomo

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

1. Calcular los Puntos de Fusión

- Calcular los PF sin ajustar

Factor de Ponderación					
Parámetro de medición	Cuenta	Simple	Medio	Complejo	
Número de entradas del usuario	6	3	4	6	18
Número de salidas del usuario	8	4	5	7	32
Número de peticiones del usuario	8	3	4	6	24

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Número de archivos	1	7	10	15	7
Número de interfaces externas	1	5	7	10	5
Cuenta Total					86

Tabla 10: Puntos de fusión sin ajustar

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

- Calculamos los PF ajustados

Descripción del Proceso	Rango: 0-5
1. Sistema copias seguridad y recuperación fiables?	1
2. Requiere comunicación de datos?	3
3. ¿Existe funciones de procesamiento distribuido?	2
4. Es crítico el rendimiento?	4
5. Se ejecutará el sistema en un entorno operativo existente y frecuentemente utilizado?	2
6. Requiere el sistema entrada de datos interactivo?	3
7. La entrada de datos interactiva se llevan a cabo sobre múltiples o variadas operaciones?	2
8. Se actualizan los archivos en forma interactiva?	2
9. Son complejas las entradas, las salidas, los archivos o peticiones?	2
10. Es complejo el procesamiento interno?	1
11. Se ha diseñado el código para ser reutilizable?	3
12. Están incluidas en el diseño la conversión y la instalación	1
13. Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	1
14. Se ha diseñado la aplicación para facilitar cambios y ser fácilmente utilizada por el usuario?	3
Total	30

Tabla 11: Puntos de fusión ajustados

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

- Calculamos los puntos de fusión finales

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

$$FA = 0.65 + (0.01 * \text{Total PF ajustados})$$

$$FA = 0.65 + (0.01 * 30)$$

$$FA = 0.95$$

$$PF = CT * FA$$

$$PF = 86 * 0.95$$

$$PF = 81.7$$

2. Determinar Kilo Líneas de Código

$$KLDC = (PF * \text{Lineas de Codigo por cada PF}) / 1000$$

$$KLDC = 81.7 * 55 / 1000$$

$$KLDC = 4.4935$$

3. Calcular el coeficiente FAE (Factor de Ajuste de Esfuerzo)

Conductores de Coste	Valoración					
	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Fiabilidad requerida del software	0.75	0.88	1.00	1.15	1.40	-
Tamaño de la base de datos	-	0.94	1.00	1.08	1.16	-
Complejidad del producto	0.70	0.85	1.00	1.15	1.30	1.65
Restricciones del tiempo de ejecución	-	-	1.00	1.11	1.30	1.66
Restricciones del almacenamiento principal	-	-	1.00	1.06	1.21	1.56
Volatilidad de la página web	-	0.87	1.00	1.15	1.30	-
Tiempo de respuesta de la página web	-	0.87	1.00	1.07	1.15	-
Capacidad del analista	1.46	1.19	1.00	0.86	0.71	-
Experiencia en la aplicación	1.29	1.13	1.00	0.91	0.82	-
Capacidad de los programadores	1.42	1.17	1.00	0.86	0.70	-
Experiencia en SO utilizado	1.21	1.10	1.00	0.90	-	-
Experiencia en lenguaje de programación	1.14	1.07	1.00	0.95	-	-
Prácticas de programación modernas	1.24	1.10	1.00	0.91	0.82	-
Utilización de herramientas de software	1.24	1.10	1.00	0.91	0.83	-
Limitaciones de planificación del software	1.23	1.08	1.00	1.04	1.10	-

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Tabla 12: Valoración de costos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

El FAE se obtiene de los valores que se muestran sombreados, que nos da como resultado la siguiente operación:

$$\text{FAE} = 1.15 * 0.94 * 0.7 * 0.87 * 1.07 * 0.86 * 0.91 * 0.70 * 0.95 * 0.91 * 1 * 1$$

$$\text{FAE} = 0.333603$$

4. Realizamos la estimación de los factores de esfuerzo, tiempo, productividad y recursos necesarios.

- **Cálculo de Esfuerzo**

$$E = a * (\text{KLDC}^e * \text{FAE})$$

$$E = 3.2 * (4.4935^{1.05}) * 0.333603$$

$$E = 5.17 \text{ personas/mes}$$

- **Cálculo de Tiempo de Desarrollo**

$$T = c * E^d$$

$$T = 2.5 * (5.17)^{0.38}$$

$$T = 4.66 \text{ meses}$$

- **Cálculo de Productividad**

$$\text{PR} = \text{LDC}/E$$

$$\text{PR} = 4493.5/5.17$$

$$\text{PR} = 869.15$$

- **Cálculo de Personal promedio**

$$P = E/T$$

$$P = 5.17/4.66$$

$$P = 1.11 = 2 \text{ personas aprox.}$$

2 ANÁLISIS

El proceso para la identificación de riesgos es un tema muy importante para realizar debido a que nos permite a las personas a poder mitigar o responder contra un posible

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

riesgo que se da en el proyecto para ello es importante tomar en cuenta los siguientes pasos: identificación, Análisis, Planeación y Supervisión de riesgos.

Según con el Project Management Institute PMI (2004), riesgo es “un evento o condición incierta que, si ocurre, tiene un efecto positivo o negativo en al menos unos de los objetivos del proyecto, tales como tiempo, costo, alcance o calidad”.

2.1 Identificación de Riesgos

Una vez realizada el trabajo de ingeniería de software del proyecto se ha determinado los siguientes riesgos que se detallan en la Tabla 8.

Id riesgo	Descripción del riesgo
RI-01	Falta de compromiso por parte del cliente con el proyecto
RI-02	Fallo en los equipos utilizados para el desarrollo
RI-03	Interfaces graficas poco amigables que dificulten obtener los resultados en la fase de pruebas
RI-04	Entidades definidas de forma incorrecta o no definidas
RI-05	Reactivación de la pandemia COVID-19
RI-06	El personal del equipo de desarrollo tenga una calamidad
RI-07	Protestas y movilizaciones que restrinjan la atención en el restaurante

Tabla 13: Listado de los riesgos identificados dentro del proyecto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

2.1.1 Estimación de Probabilidad

Rango de probabilidad	Promedio para el cálculo	Expresión del lenguaje natural	Valor numérico
De 1% a 10%	5%	Baja	1
De 11% a 25%	18%	Poco probable	2
De 26% a 55%	40%	Media	3
De 56% a 80%	68%	Altamente probable	4
De 81% a 99%	90%	Casi seguro	5

Tabla 14: Cuantificación de incertidumbre

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

2.1.2 Probabilidad de ocurrencia del riesgo

ID	Riesgo	Expresión	Probabilidad
-----------	---------------	------------------	---------------------

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

RI-01	Falta de compromiso por parte del cliente con el proyecto	Media	40%
RI-02	Fallo en los equipos utilizados para el desarrollo	Media	30%
RI-03	Interfaces graficas poco amigables que dificulten obtener los resultados en la fase de pruebas	Poco probable	20%
RI-04	Entidades definidas de forma incorrecta o no definidas	Baja	10%
RI-05	Reactivación de la pandemia COVID-19	Media	50%
RI-06	El personal del equipo de desarrollo tenga una calamidad	Poco probable	13%
RI-07	Protestas y movilizaciones que restrinjan la atención en el restaurante	Altamente probable	70%

Tabla 15: Probabilidad de ocurrencia del riesgo

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

2.1.3 Estimación de impacto y ocurrencia de los riesgos

Criterio	Retraso en la planificación	Valor numérico
Insignificante	3 días	1
Marginal	1 semana	2
Medio crítico	2 semanas	3
Catastrófico	1 mes o más	4

Tabla 16: Estimación de impacto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Riesgo	Impacto
RI-01	Insignificante
RI-02	Insignificante
RI-03	Marginal
RI-04	Insignificante
RI-05	Marginal
RI-06	Medio crítico
RI-07	Medio crítico

Tabla 17: Ocurrencia de los riesgos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

2.1.4 Exposición al riesgo

ID	Riesgo	Probabilidad	Impacto	Exposición
RI-01	Falta de compromiso por parte del cliente con el proyecto	40%	1	0.4
RI-02	Fallo en los equipos utilizados para el desarrollo	30%	1	0.3
RI-03	Interfaces graficas poco amigables que dificulten obtener los resultados en la fase de pruebas	20%	2	0.1
RI-04	Entidades definidas de forma incorrecta o no definidas	10%	1	0.1
RI-05	Reactivación de la pandemia COVID-19	50%	2	1
RI-06	El personal del equipo de desarrollo tenga una calamidad	13%	3	0.39
RI-07	Protestas y movilizaciones que restrinjan la atención en el restaurante	70%	3	2.1

Tabla 18: Exposición del riesgo planteado

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

2.1.5 Matriz de mitigación de Riesgos

Código del Riesgo	Descripción del riesgo	Fase	Nivel del Riesgo	Tipo de Respuesta	Responsable	Plan de mitigación
RI-01	Falta de compromiso por parte del cliente con el proyecto	Análisis / Planificación	Media	Corregir	Angello Ordoñez	<ul style="list-style-type: none"> • Establecer reuniones de emergencia • Evidenciar la asistencia de las reuniones • Enviar comunicados por correo electrónico, llamada, mensaje de texto

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

RI-02	Fallo en los equipos utilizados para el desarrollo	Implementación	Media	Mitigar	Angello Ordoñez / Emily Chimbo	<ul style="list-style-type: none"> • Respaldos diarios en la nube de la documentación • Uso de un sistema de control de versiones
RI-03	Interfaces graficas poco amigables que dificulten obtener los resultados en la fase de pruebas	Revisión y retrospectiva	Poco probable	Mitigar	Angello Ordoñez	<ul style="list-style-type: none"> • Asesoramiento con personal capacitado de UI/UX.
RI-04	Entidades definidas de forma incorrecta o no definidas	Análisis	Baja	Corregir	Emily Chimbo	<ul style="list-style-type: none"> • Reestructurar el diseño de la base de datos
RI-05	Reactivación de la pandemia COVID-19	Todas las fases	Media	Aceptar	Angello Ordoñez / Emily Chimbo	<ul style="list-style-type: none"> • Emplear sistemas de comunicación virtuales
RI-06	El personal del equipo de desarrollo tenga una calamidad	Todas las fases	Poco probable	Aceptar	Angello Ordoñez / Emily Chimbo	<ul style="list-style-type: none"> • Tratar de cumplir con las actividades del compañero
RI-07	Protestas y movilizaciones que restrinjan la atención en el restaurante	Resultados	Altamente probable	Aceptar	Angello Ordoñez / Emily Chimbo	<ul style="list-style-type: none"> • Tratar de cumplir con el desarrollo de procesos que no necesiten presencialidad

Tabla 19: Matriz de proceso de mitigación de riesgos

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

3 PLANIFICACIÓN DEL PROYECTO

3.1 Ámbito del Software

El producto de software a construir tiene, como objetivo principal, apoyar la gestión de un restaurante. Dicho lugar está ubicado en la ciudad de Baños de Agua Santa, Ecuador. Se desea automatizar, fundamentalmente, la gestión de reservas y pedidos. En cuanto a los pedidos, debe facilitarse su gestión integral, desde el momento en que un cliente desea ver el menú pasando por la selección de alimentos, la elección de forma de pago y la posterior confirmación del pedido. Se contempla la posibilidad de utilizar correos electrónicos para agilizar la comunicación con los clientes.

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

También se desea implantar un módulo en la aplicación web en el que se pueda realizar reservas, de forma que se asegure disponibilidad de espacio para los clientes del restaurante. Los datos hechos públicos a través de la web deberán estar actualizados frecuentemente, siendo deseable que se sincronicen al instante con las bases de datos a utilizar. Nótese que el sistema debe ser desarrollado exclusivamente para la gestión de pedidos y reservas, y no otros procesos que se llevan en el restaurante.

3.2 Recursos

3.2.1 Recursos Humanos

Cantidad	Nombre del Recurso	Tiempo requerido	Costo Unitario	Costo Total
2	Desarrolladores de Software	4 meses	\$281.25	\$2550

Tabla 20: Recursos Humanos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

3.2.2 Recursos de Software, Hardware y Otros

Costos de Desarrollo y Testeo		
Componentes	Costo	Detalles
Consumo Eléctrico	\$240	\$120 por persona / 6 meses
Consumo de Internet	\$320.94	Plan Alfabet 40 Mbps x mes= \$29.99 Plan CNT 40 Mbps x mes \$23.50 179.94
Computadora ASUS Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 32Gb RAM 1Tb HDD	\$1,199.99	
Computadora Dell Inspiron 1493 Intel(R) Core(TM) i5 @ 1.70GHz 12Gb RAM 1Tb HDD + 512Gb	\$680.00	
Monitor LG	\$100.00	
Mouse Senda inalámbrico	\$20.00	
Teclado Senda inalámbrico	\$20.00	
Mouse Microsoft	\$20.00	Inalámbrico

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Audífonos Microsoft	\$54.99	Teams Integration
iPhone 7	\$170.00	
iPhone XR	\$400.00	
ISO/IEC 25000	\$ 93.00	
ISO/IEC 25040	\$ 162.38	
Azure DevOps	\$0.00	Plan Básico (Gratis 5 usuarios)
Visual Studio Code	\$0.00	
Git+GitHub	\$0.00	
MySQL	\$0.00	
Xampp	\$0.00	
Librerías y Frameworks	\$0.00	
Transporte	\$400.00	Santo Domingo – Riobamba - Baños
Windows 10	\$125.00	
Windows 11	\$125.00	
Microsoft Office 365	\$0.00	Plan Estudiantil
StarUML	\$0.00	
Curso React: De cero a experto (Hooks y MERN)	\$20.99	Plataforma: Udemy Instructor: Fernando Herrera Modalidad: Online
Curso PWA - Aplicaciones Web Progresivas: De cero a experto	\$20.99	Plataforma: Udemy Instructor: Fernando Herrera Modalidad: Online
Costos de Producción		
Componentes	Costo	Detalles
Servicios en la Nube en Azure	\$128.47	2 App Service + MySQL database (1 mes)
DNS Azure	\$0.90	Costo por mes
Costo Total		
Costos de Desarrollo y Testeo		\$4,173
Costos de Producción		\$129.37
Total		\$4,302

Tabla 21: Recursos de Software, Hardware y otros.

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

3.2.3 Conclusión

Según las cifras se requiere de un grupo de 2 personas trabajando alrededor de 4 meses para lograr el objetivo, con un costo estimado de \$6852 USD para ponerlo en producción durante un mes de servicio.

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

3.3 Cronograma de Actividades

Como el proyecto debe realizarse en un periodo de 4 meses aproximadamente se ha considerado las siguientes fechas:

Fecha de Inicio: 04/04/2022

Fecha de Fin: 22/07/2022

Además, para agilizar los procesos se ha decidido utilizar la metodología ágil Scrumban que nos brindará un flujo mejor en cada una de las actividades por realizar. El proyecto consta de 4 iteraciones con 5 fases cada una. A continuación, se puede visualizar el cronograma del proyecto.

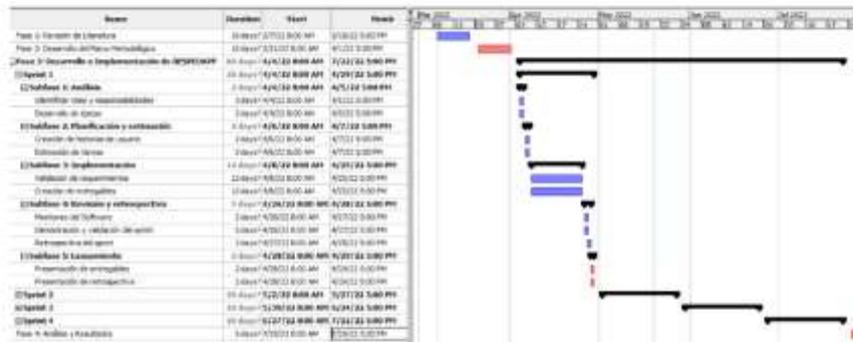


Ilustración 1: Cronograma del proyecto en Project Libre

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

3.4 Historias de Usuario

Para una mayor comprensión de cada una de las historias de usuario de la aplicación RESPEDAPP se describirán a continuación en la tabla 76 cada una de ellas:

ID	Title	Description
94	Creación de navbar	Como usuario quiero visualizar en la parte superior de la página principal una barra de navegación con enlaces a la sección de reservaciones, galería, nosotros, contacto, menú e inicio de sesión para poder acceder a las secciones mencionadas.
95	Creación de Galería	Como usuario quiero visualizar en la página principal una galería de fotos para obtener una referencia de la comida del restaurante El fogón de Coz.

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

96	Creación de Nosotros	Como usuario quiero visualizar una sección nosotros en la página principal para obtener información descriptiva e histórica del restaurante El fogón de Coz
97	Creación de Contacto	Como usuario quiero visualizar una sección de contacto que incluya un formulario para lograr comunicación con el restaurante sin necesidad de registro.
98	Creación de Registro	Como cliente quiero una página que incluya un formulario con datos para registrarme en la aplicación
99	Creación de Ingreso	Como cliente quiero una página que incluya un formulario con datos para ingresar en la aplicación
100	Creación de Sección Reservas	Como usuario quiero una sección de reservas en la página principal que incluya un formulario para poder registrar una reserva en el restaurante
101	Creación del Menú	Como usuario quiero visualizar el menú clasificado por categorías y productos para posteriormente elegir los productos deseados y realizar un pedido en el restaurante
102	Creación de Carrito	Como usuario quiero una página que muestre los productos que escogí para visualizar un resumen del pedido
103	Creación de Pedidos Locales	Como usuario quiero visualizar una página que incluya el pedido previo, registrar número de mesa y nota de pedido para ingresar un pedido local en la aplicación.
104	Creación de Pedidos Domicilio	Como usuario quiero visualizar una página que incluya el pedido previo, registrar dirección y nota de pedido para ingresar un pedido domicilio en la aplicación.
105	Creación de Pedidos Reserva	Como usuario quiero visualizar una página que incluya el pedido previo junto al número de reserva y nota de pedido para ingresar un pedido reserva en la aplicación.
106	Creación de Dashboard Restaurante	Como administrador/mesero quiero visualizar un tablero de pedidos para cambiar el estado de los pedidos del restaurante Como administrador/mesero quiero visualizar una tabla de reservas del día para obtener la información de reservas diarias del restaurante
107	Creación de Sección Usuarios	Como administrador quiero una página que me permita insertar, editar y eliminar usuarios de la aplicación para gestionar los usuarios
108	Creación de Sección Pedidos	Como administrador quiero una página que me permita ver y editar los pedidos de la aplicación para gestionar los pedidos
109	Creación de Sección Reservas	Como administrador quiero una página que me permita ver y editar las reservas de la aplicación para gestionar las reservas
110	Creación de Sección Productos	Como administrador quiero una página que me permita insertar, editar y eliminar las categorías de comida y productos de la aplicación para gestionar los productos.
111	Creación de Sección Perfil	Como cliente quiero una página que me permita ver y editar mis datos para gestionar mi cuenta en la aplicación

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

112	Creación de Sección Ordenes	Como cliente quiero una página que me permita ver los pedidos que he realizado en la aplicación para gestionar mis pedidos
113	Creación de Sección Direcciones	Como cliente quiero una página que me permita insertar editar y eliminar mis direcciones para gestionar mis próximos pedidos a domicilio.

Tabla 22: Sprint Backlog 4 de la aplicación RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

3.5 Fuentes y Colores

Para el proyecto se ha optado por utilizar algunos de los colores que forman parte de la marca del restaurante “El fogón de Coz” además de otros que ayudan a simbolizar acciones, encabezados, botones, entre otros. La fuentes y tipografía son de libre uso y han sido abstraídas de librerías de Google y Material UI. A continuación, se describirá en la ilustración 2 y 3 de mejor manera estos recursos:



Ilustración 2: Tipografía requeridos en el proyecto RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP



Ilustración 3: Colores HTML requeridos en el proyecto RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chámbo (2022)

4 IMPLEMENTACIÓN

4.1 Bosquejos de Interfaces

Página principal del restaurante “El Fogón de COZ”

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP



Ilustración 4: Bosquejo de la sección navbar y header de la landing page del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

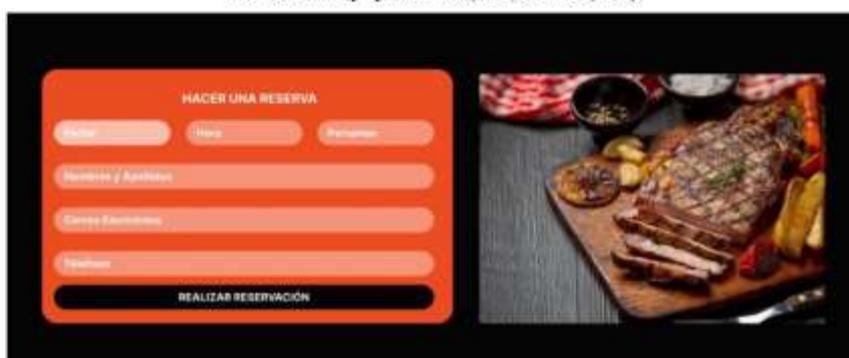


Ilustración 5: Bosquejo de la sección reservaciones de la landing page del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).



Ilustración 6: Bosquejo de la sección galería de la landing page del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP



Ilustración 7: Bosquejo de la sección acerca de nosotros de la landing page del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).



Ilustración 8: Bosquejo de la sección contacto de la landing page del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).



Ilustración 9: Bosquejo de la sección footer de la landing page del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Dashboard

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

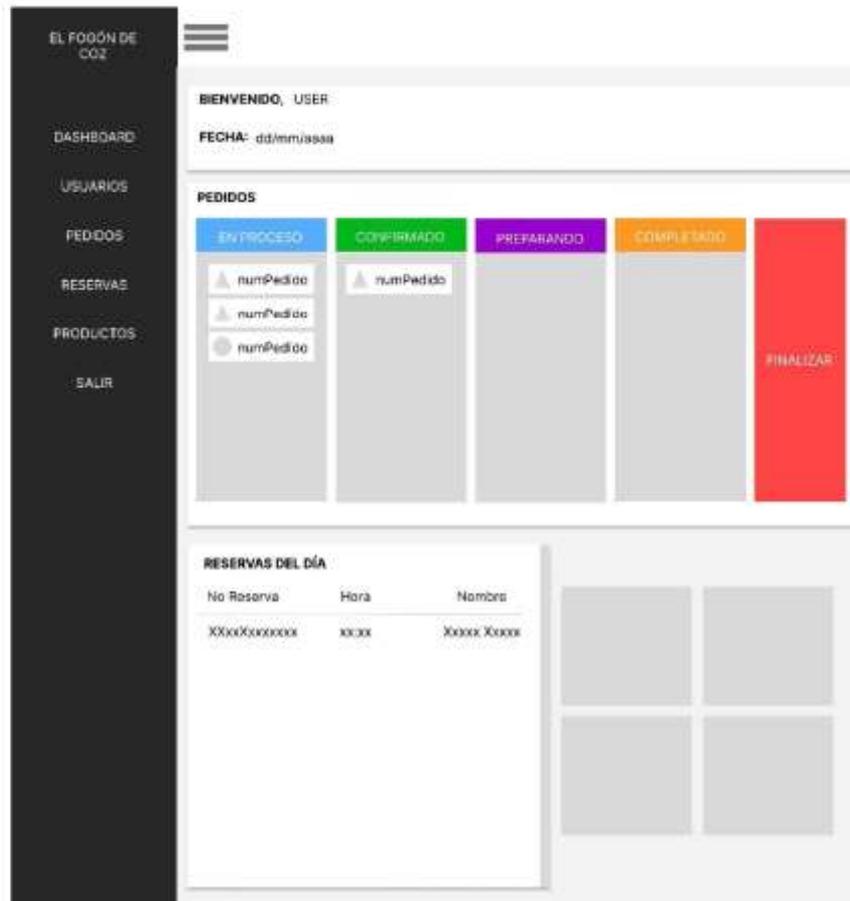


Ilustración 10: Bosquejo de la pantalla de Dashboard del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Usuarios

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP



Ilustración 11: Bosquejo de la pantalla de Usuarios cerrada del restaurante “El fogón de Coz”

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

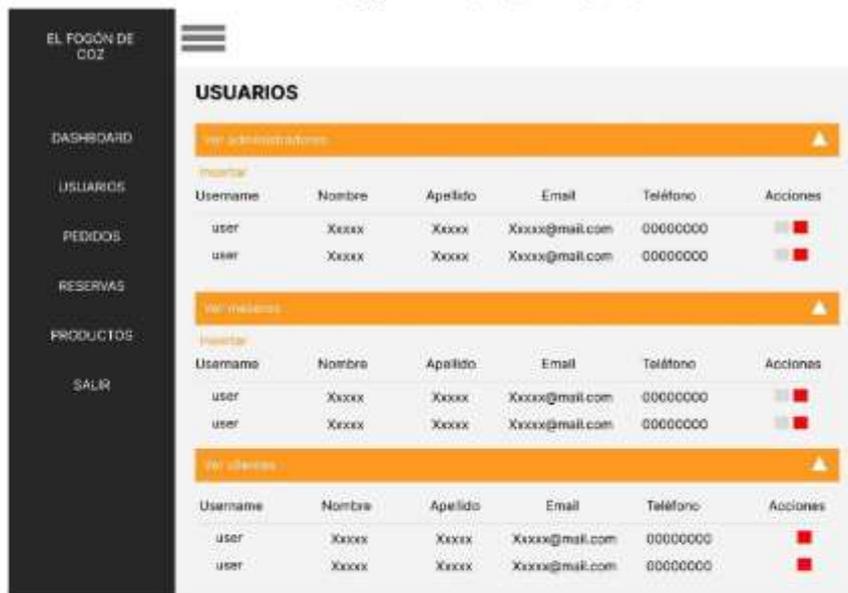


Ilustración 12: Bosquejo de la pantalla de Usuarios abierta del restaurante “El fogón de Coz”

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Pedidos



Ilustración 13: Bosquejo de la pantalla de Pedidos cerrada del restaurante “El fogón de Coz”

Fuente: Autoría propia Ordoñez (2022), Chámbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

EL FOGÓN DE COZ

PEDIDOS

Ver Pedidos Locales

No Pedido	Productos	Valor Total	Nota	Mesa	Estado	Acciones
numXXXXX	Xxxxx Xxxxx	Xxxxx	Lorem Ipsum xXXXX	X	Pendiente	

Ver Pedidos Disponibles

No Pedido	Productos	Valor Total	Nota	Dirección	Estado	Acciones
numXXXXX	Xxxxx Xxxxx	Xxxxx	Lorem Ipsum xXXXX	XXXX XXXXX	Completado	

Ver Pedidos Reserva

No Pedido	Productos	Valor Total	Nota	No Reservas	Estado	Acciones
numXXXXX	Xxxxx Xxxxx	Xxxxx	Lorem Ipsum xXXXX	numXXXXX	Pendiente	

Ver Pedidos Finalizados

No Pedido	Valor Total	Nota	Estado
numXXXXX	Xxxxx	Lorem Ipsum xXXXX	Finalizado

Ilustración 14: Bosquejo de la pantalla de Pedidos abierta del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Reservas

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP



Ilustración 15: Bosquejo de la pantalla de Reservas cerrada del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

No Reserva	Nombre	Personas	Fecha	Hora	Estado	Acciones
numXXXXX	Xxxxx Xxxxx	XX	dd/mm/aaaa	xx:xx	Pendientes	

No Reserva	Nombre	Personas	Fecha	Hora	Estado	Acciones
numXXXXX	Xxxxx Xxxxx	XX	dd/mm/aaaa	xx:xx	Confirmadas	

No Reserva	Nombre	Personas	Fecha	Hora	Estado
numXXXXX	Xxxxx Xxxxx	XX	dd/mm/aaaa	xx:xx	Finalizadas

No Reserva	Nombre	Personas	Fecha	Hora	Estado
numXXXXX	Xxxxx Xxxxx	XX	dd/mm/aaaa	xx:xx	Rechazadas

Ilustración 16: Bosquejo de la pantalla de Reservas abierta del restaurante “El fogón de Coz”

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Productos

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

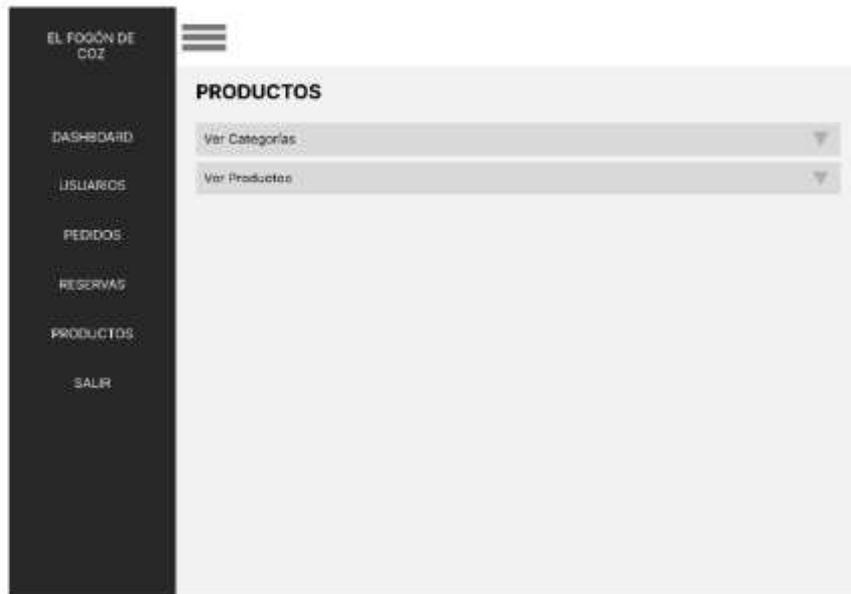


Ilustración 17: Bosquejo de la pantalla de Productos cerrada del restaurante “El fogón de Coz”

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

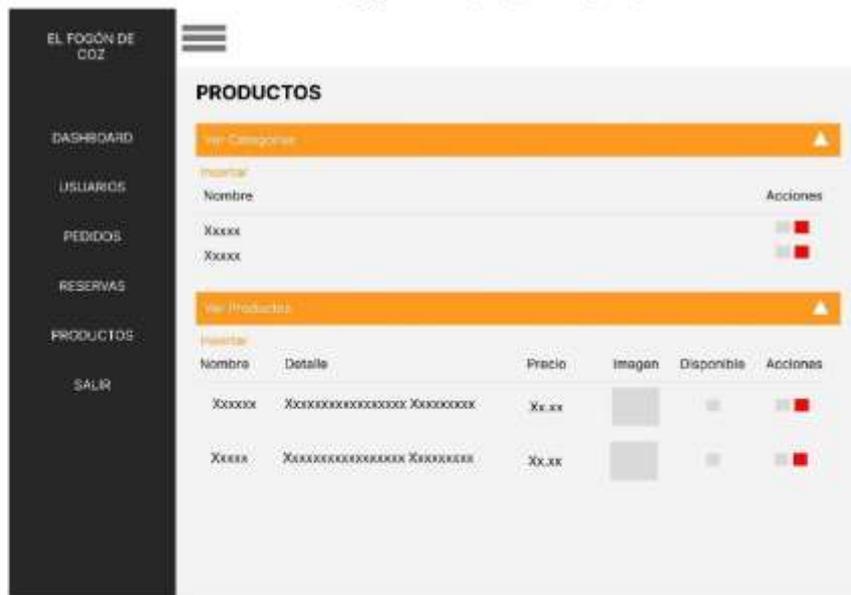


Ilustración 18: Bosquejo de la pantalla de Productos abierta del restaurante “El fogón de Coz”

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Ciente Perfil

The screenshot shows the 'Perfil' screen of the 'El Fogón de Coz' app. On the left is a dark sidebar with the restaurant name 'EL FOGÓN DE COZ' and menu items: Perfil, Ordenes, Direcciones, and Salir. The main content area is titled 'Usuario 1' and contains a form with the following fields: 'Nombre', 'Apellido', 'Número de celular', 'Teléfono', 'Correo electrónico', and 'Contraseña'. Each field has a light blue border. At the top right of the form area is a 'Editar' button. At the bottom center is an orange 'Guardar cambios' button.

Ilustración 19: Bosquejo de la pantalla de perfil cliente del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

Ciente Ordenes

The screenshot shows the 'Mis Ordenes' screen of the 'El Fogón de Coz' app. On the left is the same dark sidebar as in the previous screenshot. The main content area is titled 'Mis Ordenes' and contains a list of three order types, each with a dropdown arrow on the right: 'Pedidos Locales', 'Pedidos Domicilio', and 'Pedidos Reserva'.

Ilustración 20: Bosquejo de la pantalla de cliente ordenes cerrada del restaurante "El fogón de Coz"

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

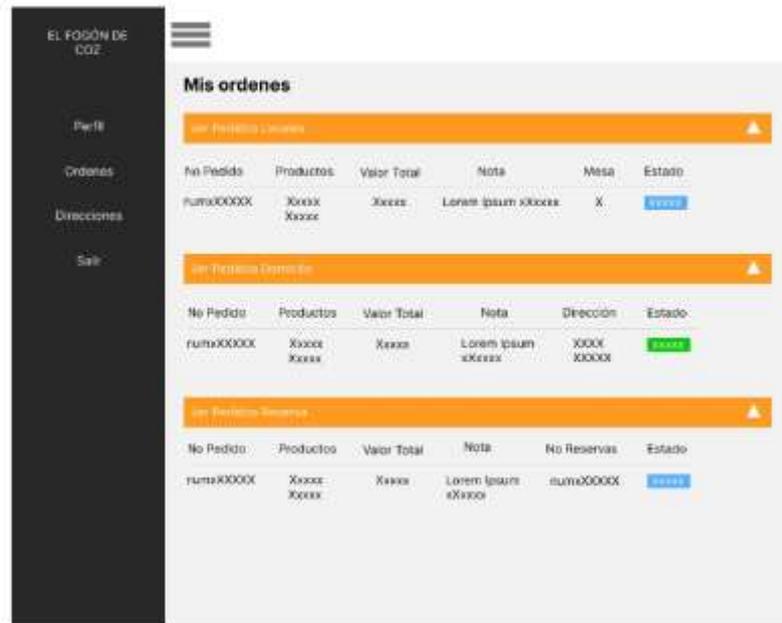


Ilustración 21: Bosquejo de la pantalla de cliente ordenes abierta del restaurante “El fogón de Coz”

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Cliente direcciones



Ilustración 22: Bosquejo de la pantalla de cliente direcciones del restaurante “El fogón de Coz”

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

4.2 Herramientas de Software

Para realizar el proyecto se ha decidido optar por el lenguaje de programación JavaScript debido a que es uno de los más populares para el desarrollo web además de su extensa documentación y librerías disponibles para un desarrollo más eficiente y completo. Además, vale recalcar que entre los recursos más importantes se ha utilizado ES6 como estándar de codificación, Node como entorno de ejecución, React como librería para el desarrollo de frontend y Next como framework para obtener una aplicación web progresiva con SSR.

4.3 Base de datos

La documentación de la base de datos es un punto fundamental a la hora de realizar un proyecto, ya que facilita el mantenimiento futuro del aplicativo por lo cual se decidió desarrollar los siguientes puntos.

4.3.1 Diagrama entidad relación

Los diagramas ER se componen de entidades, relaciones y atributos. También representan la cardinalidad, que define las relaciones en términos de números. En este caso se mostrará en la Figura 1 el diagrama ER de forma simplificada, es decir solo se definirá las entidades, relación y cardinalidad.

Una vez que se analizó la lógica del negocio provista por el propietario del restaurante se obtuvieron las siguientes entidades:

- Productos
- Categorías
- Pedidos
- Reservas
- Usuario
- Dirección

Las relaciones definidas según las entidades son las siguientes:

- Un **producto** pertenece a una sola **categoría**
- Una **categoría** puede tener cero o varios **productos**
- Un **producto** puede o no ser contenido por un **pedido**

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

- Un **pedido** contiene uno o varios **productos**
- Un **usuario** puede tener una o varias **direcciones**
- Una **dirección** puede ser tenida por un solo **usuario**
- Un **usuario** puede realizar cero o varios **pedidos**
- Un **pedido** puede ser realizado por un solo **usuario**
- Un **usuario** puede realizar cero o varias **reservas**
- Una **reserva** puede ser realizada por un solo **usuario**
- Una **reserva** puede tener cero o un solo **pedido**
- Un **pedido** puede ser tenido por una sola **reserva**

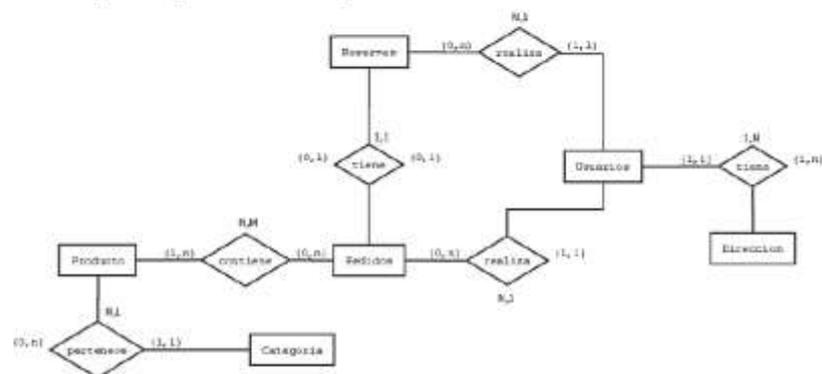


Figura 1: Diagrama entidad relación simplificado de la base de datos de RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

4.3.2. Normalización de datos

La normalización de bases de datos relacionales toma un esquema relacional y le aplica un conjunto de técnicas para producir un nuevo esquema que representa la misma información, pero contiene menos redundancias y evita posibles anomalías en las inserciones, actualizaciones y borrados. A continuación, se realizará el proceso de normalización para cada una de las entidades mencionadas previamente.

Usuario

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

id Usuario	userName	email	name	lastName	password	tipoUsuario	numUsuario	phone
1	echimbo	email@mail.com	Emily	Chimbo	12Abc	cliente	1	123456789
2	aordonez	email1@mail.com	Angello	Ordoñez	34Def	administrador	2	987654321
3	pperez	email2@mail.com	Pepe	Pérez	45Ghi	administrador	2	135792468

Tabla 23: Tabla 1 sin normalizar de entidad Usuario

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

La tabla cumple con la primera forma normal debido a que cada instancia de la entidad contiene únicamente un valor, pero varios atributos repetitivos. Cada atributo es independiente y exclusivo en su significado y en su nombre.

El resultado de realizar la segunda forma normal (2FN) es el siguiente:

idUsuario	userName	email	name	lastName	password	numUsuario	phone
1	echimbo	email@mail.com	Emily	Chimbo	12Abc	1	123456789
2	aordonez	email1@mail.com	Angello	Ordoñez	34Def	2	987654321
3	pperez	email2@mail.com	Pepe	Pérez	45Ghi	2	135792468

Tabla 24: Tabla 2 sin normalizar de entidad Usuario

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

numUsuario	tipoUsuario
1	cliente
2	administrador

Tabla 25: Tabla entidad TipoUsuario

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

El resultado de realizar la tercera forma normal (3FN) es el siguiente:

idUsuario	userName	email	name	lastName	password	phone	idTipoUsuario
1	echimbo	email@mail.com	Emily	Chimbo	12Abc	123456789	1
2	aordonez	email1@mail.com	Angello	Ordoñez	34Def	987654321	1
3	pperez	email2@mail.com	Pepe	Pérez	45Ghi	135792468	2

Tabla 26: Tabla normalizada de entidad Usuario

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

Finalmente, después de realizar el proceso de normalización se ha obtenido 2 tablas: Usuario, TipoUsuario.

Dirección

idDireccion	name	street1	street2	reference	idUsuario	phoneDir
1	casa	Diego de Ibarra	León Febres Cordero	Barrio Didonato	1	0987654332
2	Trabajo1	Av. Express	Estados Unidos	Pirámide de Medio Mundo	1	0987654332
3	Trabajo2	Diego de Ibarra	Argentinos	Loma de Quito	2	0987654332

Tabla 27: Tabla 1 sin normalizar de entidad Direccion

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

La tabla cumple con la primera forma normal debido a que cada instancia de la entidad contiene únicamente un valor, pero varios atributos repetitivos. Cada atributo es independiente y exclusivo en su significado y en su nombre.

El resultado de realizar la segunda forma normal (2FN) es el siguiente:

idDireccion	namec	idStreet1	idStreet2	reference	idUsuario	phoneDir
1	casa	1	1	Barrio Didonato	1	0987654332
2	Trabajo1	2	2	Pirámide de Medio Mundo	1	0987654332
3	Trabajo2	1	3	Loma de Quito	2	0987654332

Tabla 28: Tabla 2 sin normalizar de entidad Direccion
Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idStreet1	streetName
1	Diego de Ibarra
2	Av.Express

Tabla 29: Tabla entidad Street1
Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idStreet2	streetName
1	León Febres Cordero
2	Estados Unidos
3	Argentinos

Tabla 30: Tabla entidad Street2
Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

El resultado de realizar la tercera forma normal (3FN) es el siguiente:

idDireccion	name	idStreet1	idStreet2	reference	phoneDir
1	casa	1	1	Barrio Didonato	0987654332
2	Trabajo1	2	2	Pirámide de Medio Mundo	0987654332
3	Trabajo2	1	3	Loma de Quito	0987654332

Tabla 31: Tabla normalizada de entidad Direccion

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idUsuario	idDireccion
1	1
1	2
2	3

Tabla 32: Tabla entidad Relación Usuario Direccion

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Finalmente, después de realizar el proceso de normalización se ha obtenido 4 tablas: Direccion, Street1, Street2 y RelacionDireccionUsuario.

Categoría

idCategoría	name
1	Carnes
2	Postres
3	Bebidas

Tabla 33: Tabla normalizada Categoría

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

La tabla cumple con la primera forma normal debido a que cada instancia de la entidad contiene únicamente un valor, pero varios atributos repetitivos. Cada atributo es independiente y exclusivo en su significado y en su nombre. Cumple con la segunda norma puesto que cada atributo que no está en la clave primaria proporciona un hecho que depende de la clave completa. Y finalmente, cumple la tercera forma normal debido a que cada atributo de clave no primaria proporciona un hecho independiente de otros atributos no de clave y que depende únicamente de la clave. Por lo tanto, esta entidad no requiere normalización.

Producto

idProducto	name	detail	price	availability	image	idCategoria
1	Hamburguesa	Hamburguesa con 200gr de carne	2.5	si	null	1
2	Hamburguesa Simple	Hamburguesa con 300gr de carne	2.6	si	Imagenbase64	1
3	Coca Cola	Bebida gaseosa de 200ml	0.6	no	Imagenbase64_2	3

Tabla 34: Tabla 1 sin normalizar de entidad Producto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

La tabla cumple con la primera forma normal debido a que cada instancia de la entidad contiene únicamente un valor, pero varios atributos repetitivos. Cada atributo es independiente y exclusivo en su significado y en su nombre.

El resultado de realizar la segunda forma normal (2FN) es el siguiente:

idProducto	name	detail	price	idAvailability	image	idCategoria
------------	------	--------	-------	----------------	-------	-------------

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

1	Hamburguesa	Hamburguesa con 200gr de carne	2.5	1	null	1
2	Hamburguesa Simple	Hamburguesa con 300gr de carne	2.6	1	Imagenbase64	1
3	Coca Cola	Bebida gaseosa de 200ml	0.6	2	Imagenbase64_2	3

Tabla 35: Tabla 2 sin normalizar de entidad Producto

Fuente: Autoría propia Ordoñez (2022), Chámbo (2022).

idAvailability	availability
1	si
2	no

Tabla 36: Tabla entidad Availability

Fuente: Autoría propia Ordoñez (2022), Chámbo (2022).

El resultado de realizar la tercera forma normal (3FN) es el siguiente:

idProducto	name	detail	price	image
1	Hamburguesa	Hamburguesa con 200gr de carne	2.5	null
2	Hamburguesa Simple	Hamburguesa con 300gr de carne	2.6	Imagenbase64
3	Coca Cola	Bebida gaseosa de 200ml	0.6	Imagenbase64_2

Tabla 37: Tabla normalizada de entidad Producto

Fuente: Autoría propia Ordoñez (2022), Chámbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

idProducto	idCategoria
1	1
2	1
3	3

Tabla 38: Tabla entidad RelacionProductoCategoria

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idProducto	idAvailability
1	1
2	1
3	2

Tabla 39: Tabla entidad RelacionProductoAvailability

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Finalmente, después de realizar el proceso de normalización se ha obtenido 4 tablas: Producto, Availability, RelacionProductoAvailability y RelacionProductoCategoria.

Pedido

idPedido	idProducto	totalPrice	tipoPedido	extra	note	state	idUsuario	idDireccion
1	1	5.00	local	0.00	null	Recibido	1	null
1	2	5.00	local	0.00	null	Recibido	1	null
2	1	2.00	domicilio	2.00	Sin sal	En proceso	2	3
3	2	3.00	local	0.00	Término medio	Finalizado	1	null

Tabla 40: Tabla 1 sin normalizar de entidad Pedido

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

El resultado de realizar la primera forma normal (1FN) es el siguiente:

Tabla 19. Entidad Pedido sin normalizar

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

idPedido	idProducto	totalPrice	tipoPedido	note	state	idUsuario	idDireccion
1	1	5.00	local	null	Recibido	1	null
1	2	5.00	local	null	Recibido	1	null
2	1	2.00	domicilio	Sin sal	En proceso	2	3
3	2	3.00	local	Término medio	Finalizado	1	null

Tabla 41: Tabla 2 sin normalizar de entidad Pedido

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Tabla 20. Entidad tipoPedido

tipoPedido	extra
local	0.00
domicilio	2.00

Tabla 42: Tabla entidad TipoPedido

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

El resultado de realizar la segunda forma normal (2FN) es el siguiente:

idPedido	idProducto	totalPrice	idTipoPedido	note	idState	idUsuario	idDireccion	Pagado
1	1	5.00	1	null	1	1	null	true
1	2	5.00	1	null	1	1	null	false
2	1	2.00	2	Sin sal	2	2	3	false
3	2	3.00	1	Término medio	3	1	null	false

Tabla 43: Tabla 3 sin normalizar de entidad Pedido

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idTipoPedido	tipo
--------------	------

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

1	local
2	domicilio
3	reserva

Tabla 44: Tabla entidad TipoPedido

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Tabla 23. Entidad State

idState	state
1	Recibido
2	En Proceso
3	Finalizado

Tabla 45: Tabla entidad Estado

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

El resultado de realizar la tercera forma normal (3FN) es el siguiente:

Tabla 24. Entidad Pedido

idPedido	totalPrice	idTipoPedido	note	idState	Pagado
1	5.00	1	null	1	true
2	2.00	2	Sin sal	2	false
3	3.00	1	Término medio	3	false

Tabla 46: Tabla normalizada entidad Pedido

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Tabla 25. Entidad RelacionPedidoProducto

idPedido	idProducto
1	1

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

1	2
2	1
3	2

Tabla 47: Tabla entidad RelacionPedidoProducto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Tabla 26. Entidad RelacionPedidoUsuario

idPedido	idUsuario
1	1
2	2
3	1

Tabla 48: Tabla entidad RelacionPedidoUsuario

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idPedido	idDireccion
1	null
2	3
3	null

Tabla 49: Tabla entidad RelacionPedidoDireccion

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Tabla 27. Entidad RelacionPedidoTipoPedido

idPedido	idTipoPedido
1	1
2	2
3	1

Tabla 50: Tabla entidad RelacionPedidoTipoPedido

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

idPedido	idState
1	1
2	2
3	3

Tabla 51: Tabla entidad RelacionPedidoState

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Finalmente, después de realizar el proceso de normalización se ha obtenido 8 tablas: Pedido, TipoPedido, State, RelacionPedidoProducto, RelacionPedidoUsuario, RelacionPedidoTipoPedido, RelacionPedidoState y RelacionPedidoDireccion

Reserva

idReserva	reservationDate	reservationTime	people	note	idUsuario	idPedido
1	25/05/2022	10:00	2	null	1	null
2	26/05/2022	15:00	2	null	1	null
3	25/05/2022	10:00	4	Cerca de la puerta	2	2
4	01/06/2022	17:00	6	Cumpleaños	1	3

Tabla 52: Tabla 1 sin normalizar de entidad Reserva

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

La tabla cumple con la primera forma normal debido a que cada instancia de la entidad contiene únicamente un valor, pero varios atributos repetitivos. Cada atributo es independiente y exclusivo en su significado y en su nombre.

El resultado de realizar la segunda forma normal (2FN) es el siguiente:

idReserva	idTiempoReserva	people	note	idUsuario	idPedido
-----------	-----------------	--------	------	-----------	----------

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

1	1	2	null	1	null
2	2	2	null	1	null
3	1	4	Cerca de la puerta	2	2
4	3	6	Cumpleaños	1	3

Tabla 53: Tabla 2 sin normalizar de entidad Reserva

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idTiempoReserva	reservationDate	reservationTime
1	25/05/2022	10:00
2	26/05/2022	15:00
3	01/06/2022	17:00

Tabla 54: Tabla entidad TiempoReserva

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

El resultado de realizar la tercera forma normal (3FN) es el siguiente:

idReserva	people	note
1	2	null
2	2	null
3	4	Cerca de la puerta
4	6	Cumpleaños

Tabla 55: Tabla normalizada entidad Reserva

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idReserva	idUsuario
1	1

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

2	1
3	2
4	1

Tabla 56: Tabla entidad RelacionReservaUsuario

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idReserva	idPedido
1	null
2	null
3	2
4	3

Tabla 57: Tabla entidad RelacionReservaPedido

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

idReserva	idTiempoReserva
1	1
2	2
3	1
4	3

Tabla 58: Tabla entidad RelacionReservaTiempoReserva

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Finalmente, después de realizar el proceso de normalización se ha obtenido 5 tablas: Reserva, TiempoReserva, RelacionReservaUsuario, RelacionReservaPedido y RelacionReservaTiempoReserva.

4.3.3 Diagrama Lógico

Luego de realizar la normalización de base de datos y eliminar tablas redundantes y/o innecesarias se ha concluido en utilizar 13 tablas en total. A continuación, se puede visualizar el diagrama lógico de la aplicación RespedApp:

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Una vez definidas las entidades con sus relaciones y después de realizar la normalización de datos hasta su tercera forma normal, a continuación, se describirá la implementación de la base de datos para ser utilizada dentro del proyecto.

4.3.5 Gestor de Base de Datos

El gestor de base de datos que se va a utilizar en el proyecto es MySQL, este brinda un sistema de gestión de base de datos relacional. Es de código abierto y tiene alta popularidad entre desarrolladores debido a su fácil uso. Para la administración de la base de datos se usa la herramienta MySQL Workbench.

A continuación, se puede observar la consulta SQL para poder generar la base de datos en el siguiente link:

https://liveespochedu-my.sharepoint.com/:w/g/personal/emily_chimbo_esPOCH_edu_ec/ESPBW7AndBVOji61tW3I9LABbx-WWaVQxql8Zu0OHQQcFw?e=P6GTsh

4.3.6 Diccionario de datos

A continuación, se adjunta el diccionario de datos de la aplicación Respedapp, este diccionario permitirá que en un futuro mantenimiento se tenga una precisión semántica sobre los datos que se manejan en la base de datos. Para la generación de este diccionario de datos se utilizó phpMyAdmin, este gestor tiene un apartado que permite generar el diccionario en base a la estructura de tablas.

Categoría

Column	Type	Null	Default	Links to	Comments	Media type
IDCATEGORIA (<i>Primary</i>)	int(11)	No				
NAME	varchar(20)	No				

Figura 2: Entidad Categoría con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDCATEGORIA	5	A	No	

Figura 3: Relaciones de Entidad Categoría

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Dirección

Column	Type	Null	Default	Links to	Comments	Media type
IDDIRECCION (<i>Primary</i>)	int(11)	No				
REFERENCE	varchar(150)	Yes	NULL			
STREET1	varchar(100)	No				
STREET2	varchar(100)	No				
NAME	varchar(20)	No				
PHONEDIR	varchar(15)	No				

Figura 4: Entidad Dirección con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDDIRECCION	7	A	No	

Figura 5: Relaciones de Entidad Dirección

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Estado

Column	Type	Null	Default	Links to	Comments	Media type
IDSTATE (<i>Primary</i>)	int(11)	No				
STATE	varchar(20)	No				

Figura 6: Entidad Estado de reservas con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDSTATE	1	A	No	

Figura 7: Relaciones de Entidad Estado reservas

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Estado 2

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Column	Type	Null	Default	Links to	Comments	Media type
IDSTATE (<i>Primary</i>)	int(11)	No				
STATE	varchar(20)	No				

Figura 8: Entidad Estado con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDSTATE	1	A	No	

Figura 9: Relaciones de Entidad Estado

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Producto

Column	Type	Null	Default	Links to	Comments	Media type
IDPRODUCTO (<i>Primary</i>)	int(11)	No				
IDCATEGORIA	int(11)	No		CATEGORIA -> IDCATEGORIA		
NAME	varchar(100)	No				
DETAIL	varchar(200)	Yes	NULL			
PRICE	float	No				
IMAGE	longtext	Yes	NULL			
AVAILABILITY	int(11)	No				

Figura 10: Entidad Producto con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDPRODUCTO	2	A	No	
FK_RELATIONSHIP_17	BTREE	No	No	IDCATEGORIA	2	A	No	

Figura 11: Relaciones de Entidad Producto

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Reserva

Column	Type	Null	Default	Links to	Comments	Media type
IDRESERVA (<i>Primary</i>)	int(11)	No				
NUMRESERVA	varchar(30)	No				
IDUSUARIO	int(11)	No		USUARIO -> IDUSUARIO		
IDSTATE	int(11)	Yes	NULL	ESTADO2 -> IDSTATE		
PEOPLE	int(11)	No				
NOTE	varchar(150)	Yes	NULL			
RESERVATIONDATE	date	No				
RESERVATIONTIME	varchar(5)	No				

Figura 12: Entidad Reserva con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDRESERVA	1	A	No	
FK_RELATIONSHIP_11	BTREE	No	No	IDUSUARIO	1	A	No	

Figura 13: Relaciones de Entidad Reserva

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Tipo Usuario

Column	Type	Null	Default	Links to	Comments	Media type
IDTIPOUSUARIO (<i>Primary</i>)	int(11)	No				
TIPO	varchar(20)	No				

Figura 14: Entidad Tipo Usuario con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDTIPOUSUARIO	1	A	No	

Figura 15: Relaciones de Entidad Tipo Usuario

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Usuario

Column	Type	Null	Default	Links to	Comments	Media type
IDUSUARIO (<i>Primary</i>)	int(11)	No				
IDTIPOUSUARIO	int(11)	No		TIPOUSUARIO -> IDTIPOUSUARIO		
USERNAME	varchar(20)	No				
EMAIL	varchar(100)	No				
NAME	varchar(100)	Yes	NULL			
LASTNAME	varchar(100)	Yes	NULL			
PASSWORD	varchar(12)	No				
PHONE	varchar(15)	No				

Figura 16: Entidad Usuario con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDUSUARIO	2	A	No	
AK_IDENTIFIER_2	BTREE	No	No	USERNAME	2	A	No	
FK_RELATIONSHIP_16	BTREE	No	No	IDTIPOUSUARIO	2	A	No	

Figura 17: Relaciones de Entidad Usuario

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Pedido Total

Column	Type	Null	Default	Links to	Comments	Media type
IDPEDIDOTOTAL (<i>Primary</i>)	int(11)	No				
NUMPEDIDO	varchar(30)	No				
IDSTATE	int(11)	No		ESTADO -> IDSTATE		
VALORTOTAL	float	No				
NOTE	varchar(150)	Yes	NULL			
PAGADO	tinyint(1)	No				
TIPO	int(11)	Yes	NULL			

Figura 18: Entidad Pedido Total con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDPEDIDOTOTAL	1	A	No	
FK_RELATIONSHIP_18	BTREE	No	No	IDSTATE	1	A	No	

Figura 19: Relaciones de Entidad Pedido Total

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Pedido Domicilio

Column	Type	Null	Default	Links to	Comments	Media type
IDPEDIDO (<i>Primary</i>)	int(11)	No				
IDRELACIONUD	int(11)	No		RELACIONUSUARIODIRECCION -> IDRELACIONUD		
IDPEDIDOTOTAL	int(11)	No		PEDIDOTOTAL -> IDPEDIDOTOTAL		
ENVIO	float	No				

Figura 20: Entidad Pedido Domicilio con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDPEDIDO	0	A	No	
FK_RELATIONSHIP_10	BTREE	No	No	IDRELACIONUD	0	A	No	
FK_RELATIONSHIP_15	BTREE	No	No	IDPEDIDOTOTAL	0	A	No	

Figura 21: Relaciones de Entidad Pedido Domicilio

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Pedido Local

Column	Type	Null	Default	Links to	Comments	Media type
IDPEDIDO (<i>Primary</i>)	int(11)	No				
IDUSUARIO	int(11)	No		USUARIO -> IDUSUARIO		
IDPEDIDOTOTAL	int(11)	No		PEDIDOTOTAL -> IDPEDIDOTOTAL		
MESA	int(11)	No				

Figura 22: Entidad Pedido Local con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDPEDIDO	1	A	No	
FK_RELATIONSHIP_13	BTREE	No	No	IDPEDIDOTOTAL	1	A	No	
FK_RELATIONSHIP_9	BTREE	No	No	IDUSUARIO	1	A	No	

Figura 23: Relaciones de Entidad Pedido Local

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

Pedido Reserva

Column	Type	Null	Default	Links to	Comments	Media type
IDPEDIDO (<i>Primary</i>)	int(11)	No				
IDRESERVA	int(11)	No		RESERVA -> IDRESERVA		
IDPEDIDOTOTAL	int(11)	No		PEDIDOTOTAL -> IDPEDIDOTOTAL		

Figura 24: Entidad Reserva con tipos de datos

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDPEDIDO	0	A	No	
FK_RELATIONSHIP_12	BTREE	No	No	IDRESERVA	0	A	No	
FK_RELATIONSHIP_14	BTREE	No	No	IDPEDIDOTOTAL	0	A	No	

Figura 25: Relaciones de Entidad Reserva

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Relación Pedido Producto

Column	Type	Null	Default	Links to	Comments	Media type
IDRELACIONPP (<i>Primary</i>)	int(11)	No				
IDPEDIDOTOTAL	int(11)	No		PEDIDOTOTAL -> IDPEDIDOTOTAL		
IDPRODUCTO	int(11)	No		PRODUCTO -> IDPRODUCTO		
PRICE	float	No				

Figura 26: Entidad Relación Pedido Producto con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDRELACIONPP	0	A	No	
FK_REFERENCE_18	BTREE	No	No	IDPEDIDOTOTAL	0	A	No	
FK_REFERENCE_19	BTREE	No	No	IDPRODUCTO	0	A	No	

Figura 27: Relaciones de Entidad Relación Pedido Producto

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

Relación usuario dirección

Column	Type	Null	Default	Links to	Comments	Media type
IDRELACIONUD (<i>Primary</i>)	int(11)	No				
IDUSUARIO	int(11)	No		USUARIO -> IDUSUARIO		
IDDIRECCION	int(11)	No		DIRECCION -> IDDIRECCION		
DEFAULTDIR	tinyint(1)	No				

Figura 28: Entidad Relación usuario dirección con tipos de datos

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022)

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	IDRELACIONUD	2	A	No	
FK_RELATIONSHIP_19	BTREE	No	No	IDUSUARIO	2	A	No	
FK_RELATIONSHIP_20	BTREE	No	No	IDDIRECCION	2	A	No	

Figura 29: Relaciones de Entidad Relación usuario dirección

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

4.3.7 Estimación de Base de Datos

4.3.7.1 Tabla Categoría

Column	Type	Tamaño en Bytes
IDCATEGORIA (Primary)	int(11)	11
NAME	varchar(20)	20

Tabla 59: Tabla de tamaño de Bytes de entidad Categoría

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 6

Bytes por tabla: $6 \times 31 = 186$ Bytes

4.3.7.2 Tabla Dirección

Column	Type	Tamaño en Bytes
IDDIRECCION (Primary)	int(11)	11
REFERENCE	varchar(150)	150
STREET1	varchar(100)	100
STREET2	varchar(100)	100
NAME	varchar(20)	20
PHONEDIR	varchar(15)	15

Tabla 60: Tabla de tamaño de Bytes de entidad Dirección

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 6000

Bytes por tabla: $6000 * 396 = 2367000$ Bytes

4.3.7.3 Tabla Estado

Column	Type	Tamaño en Bytes
IDSTATE(<i>Primary</i>)	int(11)	11
NAME	varchar(20)	20

Tabla 61: Tabla de tamaño de Bytes de entidad Estado

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 6

Bytes por tabla: $6 * 31 = 186$ Bytes

4.3.7.4 Tabla Estado2

Column	Type	Tamaño en Bytes
IDSTATE (<i>Primary</i>)	int(11)	11
NAME	varchar(20)	20

Tabla 62: Tabla de tamaño de Bytes de entidad Estado2

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 3

Bytes por tabla: $3 * 31 = 124$ Bytes

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

4.3.7.5 Tabla Pedido Domicilio

Column	Type	Tamaño en Bytes
IDPEDIDO (Primary)	int(11)	11
IDRELACIONUD	int(11)	11
IDPEDIDOTOTAL	int(11)	11

Tabla 63: Tabla de tamaño de Bytes de entidad Pedido Domicilio

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 3360

Bytes por tabla: $3360 * 33 = 110880$ Bytes

4.3.7.6 Tabla Pedido Local

Column	Type	Tamaño en Bytes
IDPEDIDO (Primary)	int(11)	11
IDUSUARIO	int(11)	11
IDPEDIDOTOTAL	int(11)	11
MESA	int(11)	11

Tabla 64: Tabla de tamaño de Bytes de entidad Pedido Local

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 13440

Bytes por tabla: $13340 * 44 = 586960$ Bytes

4.3.7.7 Tabla Pedido Reserva

Column	Type	Tamaño en Bytes
IDPEDIDO (Primary)	int(11)	11
IDRESERVA	int(11)	11
IDPEDIDOTOTAL	int(11)	11

Tabla 65: Tabla de tamaño de Bytes de entidad Pedido Reserva

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Número de Filas estimadas: 200

Bytes por tabla: $200 * 33 = 6600$ Bytes

4.3.7.8 Tabla Pedido Total

Column	Type	Tamaño en Bytes
IDPEDIDOTOTAL (Primary)	int(11)	11
NUMPEDIDO	varchar(30)	30
IDSTATE	int(11)	11
VALORTOTAL	float	30
NOTE	varchar(150)	150
PAGADO	boolean	1
TIPO	int(11)	11

Tabla 66: Tabla de tamaño de Bytes de entidad Pedido Total

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 16800

Bytes por tabla: $16800 * 244 = 4099200$ Bytes

4.3.7.9 Tabla Producto

Column	Type	Tamaño en Bytes
IDPRODUCTO (Primary)	int(11)	11
IDCATEGORIA	int(11)	11
NAME	varchar(100)	100
DETAIL	varchar(200)	200
PRICE	float	30
IMAGE	longtext	200
AVAILABILITY	int(11)	11

Tabla 67: Tabla de tamaño de Bytes de entidad Producto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Número de Filas estimadas: 40

Bytes por tabla: $40 * 563 = 28150$ Bytes

4.3.7.10 Tabla Relacion pedido producto

Column	Type	Tamaño en Bytes
IDRELACIONPP (Primary)	int(11)	11
IDPEDIDOTAL	int(11)	11
IDPRODUCTO	int(11)	11
PRICE	float	30

Tabla 68: Tabla de tamaño de Bytes de entidad Relacion pedido producto

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 84000

Bytes por tabla: $84000 * 63 = 5292000$ Bytes

4.3.7.11 Tabla Relacion usuario direccion

Column	Type	Tamaño en Bytes
IDRELACIONUD (Primary)	int(11)	11
IDUSUARIO	int(11)	11
IDDIRECCION	int(11)	11
DEFAULTDIR	Boolean	1

Tabla 69: Tabla de tamaño de Bytes de entidad Relacion usuario direccion

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 6000

Bytes por tabla: $6000 * 34 = 204000$ Bytes

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

4.3.7.12 Tabla Reserva

Column	Type	Tamaño en Bytes
IDRESERVA (Primary)	int(11)	11
NUMRESERVA	varchar(30)	30
IDUSUARIO	int(11)	11
PEOPLE	int(11)	11
NOTE	varchar(150)	150
RESERVATIONDATE	date	50
RESERVATIONTIME	varchar(5)	5
IDSTATE	int(11)	11

Tabla 70: Tabla de tamaño de Bytes de entidad Reserva

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 400

Bytes por tabla: $400 * 279 = 111600$ Bytes

4.3.7.13 Tabla TipoUsuario

Column	Type	Tamaño en Bytes
IDTIPOUSUARIO (Primary)	int(11)	11
TIPO	varchar(20)	20

Tabla 71: Tabla de tamaño de Bytes de entidad TipoUsuario

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 3

Bytes por tabla: $3 * 31 = 93$ Bytes

4.3.7.14 Tabla Usuario

Column	Type	Tamaño en Bytes
--------	------	-----------------

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

IDUSUARIO (Primary)	int(11)	11
IDTIPOUSUARIO	int(11)	11
USERNAME	varchar(20)	20
EMAIL	varchar(100)	100
NAME	varchar(100)	100
LASTNAME	varchar(100)	100
PASSWORD	varchar(12)	12
PHONE	varchar(15)	15

Tabla 72: Tabla de tamaño de Bytes de entidad Usuario

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022).

Número de Filas estimadas: 3000

Bytes por tabla: $3000 * 369 = 1107000$ Bytes

4.3.7.15 *Total de Tamaño de Base de Datos*

13913979 Bytes = 13.2694 MB

4.4 Product Backlog

A continuación, se visualiza el listado ordenado y priorizado de los requisitos necesarios para la implementación del proyecto. Se ha dividido por épicas, características, historias de usuario y tareas como se puede visualizar en la tabla 71 que está a continuación:

ID	Work Item Type	Title	Tags
5	Epic	Registro Principal	
6	Epic	Pedidos	
7	Epic	Reservas-Menú	
8	Epic	Dashboards	
9	Feature	Base de Datos	
15	Feature	Landing Page	
19	Feature	Logín	
34	Feature	Menú	
87	Feature	Reservas	
88	Feature	Pedidos Locales	

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

89	Feature	Dashboard Cliente	
90	Feature	Pedidos Domicilio	
91	Feature	Pedidos Reserva	
92	Feature	Dashboard Restaurante	
94	User Story	Creación de navbar	
95	User Story	Creación de Galería	
96	User Story	Creación de Nosotros	
97	User Story	Creación de Contacto	
98	User Story	Creación de Registro	
99	User Story	Creación de Ingreso	
100	User Story	Creación de Sección Reservas	
101	User Story	Creación del Menú	
102	User Story	Creación de Carrito	
103	User Story	Creación de Pedidos Locales	
104	User Story	Creación de Pedidos Domicilio	
105	User Story	Creación de Pedidos Reserva	
106	User Story	Creación de Dashboard Restaurante	
107	User Story	Creación de Sección Usuarios	
108	User Story	Creación de Sección Pedidos	
109	User Story	Creación de Sección Reservas	
110	User Story	Creación de Sección Productos	
111	User Story	Creación de Sección Perfil	
112	User Story	Creación de Sección Ordenes	
113	User Story	Creación de Sección Direcciones	
114	Task	Crear Base de Datos en MySQL	BE
115	Task	Crear modelos con Sequelize	BE
116	Task	Crear enlaces entre secciones en Landing Page	FE
117	Task	Subir fotos de galería a Cloudinary	FE
118	Task	Crear galería interactiva en landing page	FE
119	Task	Colocar imagen e historia en el landing page	FE
120	Task	Generar Api de Contacto	BE
121	Task	Crear formulario de contacto en landing page	FE
122	Task	Crear Api de registro de usuario	BE
123	Task	Crear Api de ingreso de usuario	BE
124	Task	Crear formulario de registro	FE
125	Task	Crear formulario de ingreso	FE
126	Task	Crear Api de registro de reservas	BE
127	Task	Crear formulario de reservas en landing page	FE
128	Task	Crear Api de obtener categorías	BE
129	Task	Crear Api de obtener productos	BE
130	Task	Crear Api de registrar pedidoTotal	BE
131	Task	Crear grid de productos y categorías	FE
132	Task	Crear Resumen del pedido	FE
133	Task	Crear Api de registrar pedidoLocal	BE
134	Task	Crear formulario de PedidoLocal	FE
135	Task	Crear Api de registro de PedidoDomicilio	BE

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

136	Task	Crear Api de obtenerDireccionesUsuario	BE
137	Task	Crear grid de direcciones de usuario	FE
138	Task	Crear formulario de pedidoDireccion	FE
139	Task	Crear Api de registrar PedidoReserva	BE
140	Task	Crear formulario de PedidoReserva	FE
141	Task	Crear Apis de obtenerPedidos	BE
142	Task	Crear Drag&Drop de pedidos	FE
143	Task	Crear Api de Reservas del día	BE
144	Task	Crear tabla de reservas por día	FE
145	Task	Crear Apis de gestión usuarios	BE
146	Task	Crear secciones por tipo de usuario	FE
147	Task	Crear modales de insertar, editar y eliminar usuario	FE
148	Task	Crear Apis de gestión de pedidos	BE
149	Task	Crear secciones por tipo de pedido	FE
150	Task	Crear modal de editar pedido	FE
151	Task	Crear Apis de gestión de reservas	BE
152	Task	Crear secciones por tipo de reserva	FE
153	Task	Crear modal de editar reserva	FE
154	Task	Crear Apis de gestionar categorías	BE
155	Task	Crear Apis de gestionar productos	BE
156	Task	Crear secciones categorías y productos	FE
157	Task	Crear modales de insertar, editar y eliminar categorías y productos	FE
158	Task	Crear Api de editar usuario	BE
159	Task	Crear formulario de editar usuario	FE
160	Task	Crear Apis de obtenerPedidosUsuario por tipos	BE
161	Task	Crear secciones de pedidos por tipo	FE
162	Task	Crear Apis de gestionar direcciones	BE
163	Task	Crear grid de direcciones	FE
164	Task	Crear modales de insertar, editar y eliminar direccion	FE

Tabla 73: Product Backlog de la aplicación RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

4.5 Sprint Backlog

Como se ha mencionado previamente, para el presente proyecto se ha definido cuatro iteraciones con las siguientes fechas:

Sprint 1: 04/04/2022 - 29/04/2022

Sprint 2: 02/05/2022 - 27/05/2022

Sprint 3: 30/05/2022 - 24/06/2022

Sprint 4: 27/06/2022 - 22/07/2022

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

A continuación, se describirá en la tabla 72,73,74 y 75 cada uno de los ítems que se realizaron por iteración:

4.5.1 Sprint 1

ID	Work Item Type	Title 1	Title 2
97	User Story	Creación de Contacto	
120	Task		Generar Api de Contacto
121	Task		Crear formulario de contacto en landing page
94	User Story	Creación de navbar	
116	Task		Crear enlaces entre secciones en Landing Page
95	User Story	Creación de Galería	
117	Task		Subir fotos de galería a Cloudinary
118	Task		Crear galería interactiva en landing page
96	User Story	Creación de Nosotros	
119	Task		Colocar imagen e historia en el landing page
98	User Story	Creación de Registro	
122	Task		Crear Api de registro de usuario
124	Task		Crear formulario de registro
99	User Story	Creación de Ingreso	
123	Task		Crear Api de ingreso de usuario
125	Task		Crear formulario de ingreso

Tabla 74: Sprint Backlog 1 de la aplicación RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chimbo (2022).

4.5.2 Sprint 2

ID	Work Item Type	Title 1	Title 2
100	User Story	Creación de Sección Reservas	
126	Task		Crear Api de registro de reservas
127	Task		Crear formulario de reservas en landing page
101	User Story	Creación del Menú	
128	Task		Crear Api de obtener categorías
129	Task		Crear Api de obtener productos
131	Task		Crear grid de productos y categorías
102	User Story	Creación de Carrito	
130	Task		Crear Api de registrar pedidoTotal
132	Task		Crear Resumen del pedido

Tabla 75: Sprint Backlog 2 de la aplicación RESPEDAPP

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chámbo (2022).

4.5.3 Sprint 3

ID	Work Item Type	Title 1	Title 2
103	User Story	Creación de Pedidos Locales	
133	Task		Crear Api de registrar pedidoLocal
134	Task		Crear formulario de PedidoLocal
104	User Story	Creación de Pedidos Domicilio	
135	Task		Crear Api de registro de PedidoDomicilio
136	Task		Crear Api de obtenerDireccionesUsuario
137	Task		Crear grid de direcciones de usuario
138	Task		Crear formulario de pedidoDireccion
105	User Story	Creación de Pedidos Reserva	
139	Task		Crear Api de registrar PedidoReserva
140	Task		Crear formulario de PedidoReserva

Tabla 76: Sprint Backlog 3 de la aplicación RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chámbo (2022).

4.5.4 Sprint 4

ID	Work Item Type	Title 1	Title 2
113	User Story	Creación de Sección Direcciones	
162	Task		Crear Apis de gestionar direcciones
163	Task		Crear grid de direcciones
164	Task		Crear modales de insertar, editar y eliminar direccion
112	User Story	Creación de Sección Ordenes	
160	Task		Crear Apis de obtenerPedidosUsuario por tipos
161	Task		Crear secciones de pedidos por tipo

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

111	User Story	Creación de Sección Perfil
158	Task	Crear Api de editar usuario
159	Task	Crear formulario de editar usuario
106	User Story	Creación de Dashboard Restaurante
141	Task	Crear Apis de obtenerPedidos
142	Task	Crear Drag&Drop de pedidos
143	Task	Crear Api de Reservas del día
144	Task	Crear tabla de reservas por día
107	User Story	Creación de Sección Usuarios
145	Task	Crear Apis de gestión usuarios
146	Task	Crear secciones por tipo de usuario
147	Task	Crear modales de insertar, editar y eliminar usuario
108	User Story	Creación de Sección Pedidos
148	Task	Crear Apis de gestión de pedidos
149	Task	Crear secciones por tipo de pedido
150	Task	Crear modal de editar pedido
109	User Story	Creación de Sección Reservas
151	Task	Crear Apis de gestión de reservas
152	Task	Crear secciones por tipo de reserva
153	Task	Crear modal de editar reserva
110	User Story	Creación de Sección Productos
154	Task	Crear Apis de gestionar categorías
155	Task	Crear Apis de gestionar productos
156	Task	Crear secciones categorías y productos
157	Task	Crear modales de insertar, editar y eliminar categorías y productos

Tabla 77: Sprint Backlog 4 de la aplicación RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chámbo (2022).

5 REVISIÓN Y RETROSPECTIVA

En la tabla 77 se ha realizado el informe de cada una de las reuniones de revisión y retrospectiva con el cliente.

Número de Revisión	Fecha de reunión	Conclusiones del Sprint	Observaciones
1	29/04/2022	En el primer sprint se logró cumplir	<ul style="list-style-type: none"> • Agregar en el footer la otra

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

		con las historias de usuario planteadas en la planificación en el plazo acordado.	sucursal del restaurante. <ul style="list-style-type: none"> Colocar imágenes reales de la galería
2	27/05/2022	En el segundo sprint se logró cumplir con las historias de usuario planteadas en la planificación en el plazo acordado. Y además se completó las tareas pendientes del sprint anterior.	<ul style="list-style-type: none"> Agregar placeholders en el menú Implementar una animación al botón de agregar producto Generar automáticamente el número de pedido
3	24/06/2022	En el segundo sprint no se logró cumplir con las historias de usuario planteadas en la planificación en el plazo acordado debido a las tareas pendientes del sprint anterior.	<ul style="list-style-type: none"> No se presentó los pedidos de reserva
4	22/07/2022	En el primer sprint se logró cumplir con las historias de usuario planteadas en la planificación en el plazo	Ninguna

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

		acordado. Además, se quitó el desfase de las tareas pendientes relacionadas a pedidos de reserva.	
--	--	---------------------------------------------------------------------------------------------------	--

Tabla 78: Sprint Backlog 4 de la aplicación RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chumbo (2022).

5.1 Burndown Chart

A continuación, se presenta el gráfico de quemado en la figura 30 que permite visualizar y analizar el avance del equipo y conocer si se completó el trabajo comprometido a tiempo. Se utilizó la herramienta Azure DevOps para controlar los avances y como resultado tenemos que se tiene un promedio mayor a 0 lo que significa que se logró completar las tareas más rápido de lo que se esperaba inicialmente. Aunque en el sprint 3 hubo tareas que tuvieron que ser desplazadas al siguiente sprint, en el último sprint se realizó las tareas pendientes y las planificadas de mejor manera cambiando el cierre del proyecto al 11 de Julio (se estimaba concluir el 22 de Julio).

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

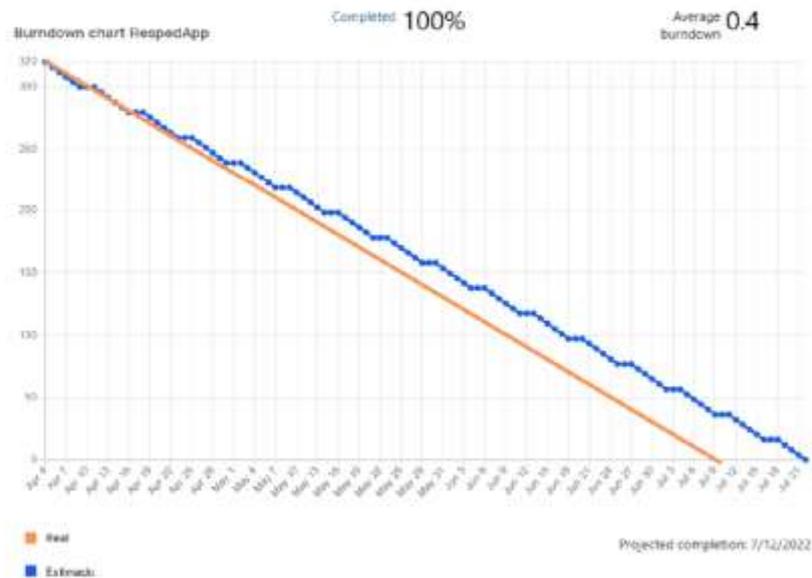


Figura 30: Diagrama de quemado de la aplicación RespedApp

6 LANZAMIENTO Y CIERRE

Para realizar el despliegue de las aplicaciones de backend y frontend de RespedApp se optó por utilizar los siguientes recursos:

Lugar de Publicación: DigitalOcean

Plan de Servicio: Droplets: 1 GB Memory / 25 GB Disk / NYC1 - Ubuntu 20.04 (LTS) x64

- ApiRespedApp (App Service backend)
- Bddrespedapp (MySQL database)

Lugar de Publicación: DigitalOcean

Plan de Servicio: B1 – NYC: \$12.00/mo – Pro

- RespedApp (App Service frontend)

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

Este cambio de lugar de despliegue implicó un gasto menor en producción por lo que se tuvo un sobrante de \$117.37. Dando como coste final del proyecto un total de \$6734.68.

Los riesgos que estuvieron presentes dentro del desarrollo e implementación de la aplicación fueron los siguientes:

Id del Riesgo	Descripción	Acciones para contrarrestar los efectos del riesgo
RI-01	Falta de compromiso por parte del cliente con el proyecto	<ul style="list-style-type: none">• Establecer reuniones de emergencia• Evidenciar la asistencia de las reuniones• Enviar comunicados por correo electrónico, llamada, mensaje de texto
RI-05	Reactivación de la pandemia COVID-19	<ul style="list-style-type: none">• Emplear sistemas de comunicación virtuales
RI-07	Se restrinja la atención del restaurante en fase de pruebas	<ul style="list-style-type: none">• Tratar de cumplir con el desarrollo de procesos que no necesiten presencialidad

Tabla 79: Matriz de gestión de riesgos de la aplicación RESPEDAPP

Fuente: Autoría propia Ordoñez (2022), Chumbo (2022)

A continuación, se describe cada uno de los entregables que se obtuvo por sprint y en la figura 31 se puede visualizar el diagrama de despliegue de la aplicación:

6.1 Entregables del Sprint 1

- Landing Page
- Login
- Registro

ANEXO B. MANUAL TÉCNICO DE RESPEDAPP

6.2 Entregables del Sprint 2

- Sección Reservas
- Menú
- Carrito

6.3 Entregables del Sprint 3

- Sección Pedidos Locales
- Sección Pedidos Domicilio

6.4 Entregables del Sprint 4

- Sección Pedidos Reserva
- Dashboard Restaurante
- Dashboard Cliente

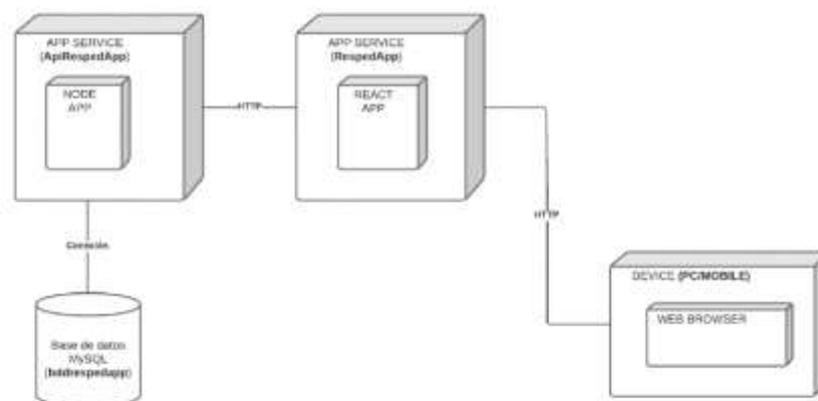


Figura 31: Diagrama de despliegue de la aplicación RespedApp

Fuente: Autoría propia Ordoñez (2022), Chímbo (2022)

ANEXO C: ACUERDO DE SERVICIO DE RESPEDAPP

ANEXO C: ACUERDO DE NIVEL DE SERVICIO

Acuerdo de nivel de servicio (SLA) para *El fogón de Coz*

Fecha: 02-07-2022

Propietario del documento:	Emily Chimbo
----------------------------	--------------

Versión

Versión	Fecha	Descripción	Autor
1.0	02-07-2022	Entrega y revisión de SLA	Emily Chimbo

Aprobación

Revisor	Rol	Firma	Fecha de Aprobación
Alejandro Cabrera	Propietario de El fogón de Coz		02-07-2022

Tabla de Contenidos

1.	Resumen del Acuerdo	3
2.	Metas y Objetivos	3
3.	Partes interesadas	3
4.	Revisión periódica	4
5.	Acuerdo de Servicio	4
5.1.	Alcance del Servicio	4
5.2.	Responsabilidades del cliente	5
5.3.	Responsabilidades del proveedor	5
5.4.	Supuestos de servicio	5
6.	Gestión de servicios	6
6.1.	Disponibilidad del servicio	6

1. Resumen del Acuerdo

Este Acuerdo representa un Acuerdo de Nivel de Servicio ("SLA" o "Acuerdo") entre *El fogón de Coz*, Emily Chimbo y Angello Ordoñez para el aprovisionamiento de los servicios de TI necesarios para respaldar y mantener RespedApp.

Este acuerdo sigue siendo válido hasta que sea reemplazado por un acuerdo revisado respaldado mutuamente por las partes interesadas. Este acuerdo describe los parámetros de todos los servicios de TI cubiertos tal como los entienden mutuamente las principales partes interesadas. Este acuerdo no reemplaza los procesos y procedimientos actuales a menos que se indique explícitamente en este documento.

2. Metas y Objetivos

El propósito de este acuerdo es garantizar que se implementen los elementos y compromisos adecuados para brindar soporte y entrega de servicios de TI consistentes a el fogón de Coz por parte de los proveedores de servicios.

Los objetivos de este Acuerdo son:

- Proporcionar una referencia clara a la propiedad del servicio, la rendición de cuentas, los roles y/o las responsabilidades.
- Presentar una descripción clara, concisa y medible de la prestación del servicio al cliente.
- Hacer coincidir las percepciones de la provisión de servicios esperada con el soporte y la entrega de servicios reales.

3. Partes interesadas

Los siguientes proveedores de servicios y clientes se utilizarán como base del Acuerdo y representan a las principales partes interesadas asociadas con este SLA:

Proveedor(es) de servicios de TI: Emily Chimbo y Angello Ordoñez

Cliente(s) de TI: El fogón de Coz

4. Revisión periódica

Este acuerdo es válido a partir de la fecha de vigencia descrita en este documento y es válido hasta nuevo aviso.

El propietario del documento es responsable de facilitar las revisiones periódicas de este documento. El contenido de este documento puede modificarse según sea necesario, siempre que se obtenga un acuerdo mutuo de las principales partes interesadas y se comunique a todas las partes afectadas.

El propietario del documento incorporará todas las revisiones posteriores y obtendrá acuerdos/aprobaciones mutuos según sea necesario.

Periodo de revisión: 1 vez al año

Fecha de revisión anterior: 02-07-2022

Fecha de próxima revisión: 02-07-2023

5. Acuerdo de Servicio

Los siguientes parámetros de servicio detallados son responsabilidad del proveedor de servicios en el soporte continuo de este acuerdo.

5.1. Alcance del Servicio

Los siguientes servicios están cubiertos por este acuerdo:

o Soporte telefónico atendido

o Soporte de correo electrónico monitoreado

o Asistencia remota

o Asistencia de emergencia (se aplican costes adicionales)

o Comprobación mensual del estado del sistema

5.2. Responsabilidades del cliente

Las responsabilidades del cliente en apoyo de este acuerdo incluyen:

- Pago de todos los costos por uso de servicios en Azure y de asistencia del grupo de desarrollo
- Disponibilidad al momento de resolver un incidente o solicitud relacionada con el servicio.

5.3. Responsabilidades del proveedor

Las responsabilidades del proveedor de servicios en apoyo de este acuerdo incluyen:

- Cumplimiento de tiempos de respuesta asociados con el servicio.
- Notificación adecuada al cliente de todos los mantenimientos programados.

5.4. Supuestos de servicio

Los supuestos relacionados con los servicios y/o componentes incluidos en el alcance incluyen:

- Los cambios en los servicios se comunicarán y documentarán a todas las partes interesadas.

6. Gestión de servicios

El soporte efectivo de los servicios dentro del alcance es el resultado de mantener niveles de servicio consistentes. Las siguientes secciones brindan detalles relevantes sobre la disponibilidad del servicio, el monitoreo de los servicios dentro del alcance y los componentes relacionados.

6.1. Disponibilidad del servicio

Los parámetros de cobertura específicos para la aplicación son los siguientes:

- Atención del restaurante: 12:00 p.m. a 10:00 p. m. Lunes a Domingo

La aplicación debe estar disponible las 24 horas todos los días, incluso en horarios no laborables debido a que los clientes pueden hacer reservas durante todo el día. Esto se resume a cumplir alrededor de 168 horas de disponibilidad por semana. Esta disponibilidad podría llegar a ser menor por los únicos motivos de:

- despliegues y/o fallos del servidor donde se encuentra alojado.

ANEXO D: ACTA DE CIERRE DE PROYECTO

ANEXO D. ACTA DE CIERRE DE PROYECTO

**Acta de Cierre de
Proyecto**

RespedApp

Fecha: 15/07/2022

ANEXO D. ACTA DE CIERRE DE PROYECTO

Tabla de contenido

Información del Proyecto	3
Datos	3
Patrocinador / Patrocinadores.....	3
Razón de cierre.....	3
Aceptación de los productos o entregables	3
Aprobaciones	5

ANEXO D. ACTA DE CIERRE DE PROYECTO

Información del Proyecto

Datos

Empresa / Organización	El fogón de Coz
Proyecto	RespedApp
Fecha de preparación	14/07/2022
Cliente	Chef Alejandro Cabrera
Patrocinador principal	Chef Alejandro Cabrera
Gerente de Proyecto	Ing. Gloria Arcos

Patrocinador / Patrocinadores

Nombre	Cargo	Departamento / División
Chef Alejandro Cabrera	Gerente Propietario	Gerencia

Razón de cierre

Por medio de la presente, se da cierre formal al proyecto, por las razones especificadas en la siguiente ficha:

Marcar con una "X" la razón de cierre:

Entrega de todos los productos de conformidad con los requerimientos del cliente.	<input checked="" type="checkbox"/>
Entrega parcial de productos y cancelación de otros de conformidad con los requerimientos del cliente.	<input type="checkbox"/>
Cancelación de todos los productos asociados con el proyecto.	<input type="checkbox"/>

Aceptación de los productos o entregables

A continuación, se establece cuales entregables de proyecto han sido aceptados:

Entregable	Aceptación (Si o No)	Observaciones
Landing Page	Si	Ninguna

ANEXO D. ACTA DE CIERRE DE PROYECTO

Login	Si	Ninguna
Registro	Si	Ninguna
Sección Reservas	Si	Ninguna
Menú	Si	Ninguna
Carrito	Si	Ninguna
Sección Pedidos Locales	Si	Ninguna
Sección Pedidos Domicilio	Si	Ninguna
Sección Pedidos Reserva	Si	Ninguna
Dashboard Restaurante	Si	Ninguna
Dashboard Cliente	Si	Ninguna

Para cada entregable aceptado, se da por entendido que:

- El entregable ha cumplido los criterios de aceptación establecidos en la documentación de requerimientos y definición de alcance.
- Se ha verificado que los entregables cumplen los requerimientos.
- Se ha validado el cumplimiento de los requerimientos funcionales y de calidad definidos.
- Se ha realizado la transferencia de conocimientos y control al área operativa.
- Se ha concluido el entrenamiento que se definió necesario.
- Se ha entregado la documentación al área operativa.

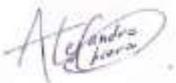
Se autoriza al Gerente de Proyecto a continuar con el cierre formal del proyecto o fase, lo cual deberá incluir:

- Reunión de Cierre
- Manual técnico del producto de software
- Manual de usuario del producto de software
- Cierre de todos los procesos de desarrollo

Una vez concluido el proceso de cierre, el Patrocinador (Sponsor) del proyecto deberá ser notificado para que el Gerente de Proyectos sea liberado y reasignado.

ANEXO D. ACTA DE CIERRE DE PROYECTO

Aprobaciones

Patrocinador	Fecha	Firma
Chef Alejandro Cabrera	15/07/2022	

Manual de la aplicación RespedApp

2022

Autor: AngeloOZ & MilyChimbo

ANEXO E: MANUAL DE USUARIO DE RESPEDAPP

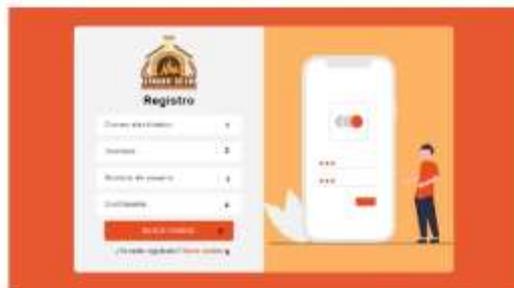
1	Iniciar sesión	3
2	Registrarse como cliente	3
3	Realizar una reservación	3
4	Realizar un pedido.....	4
4.1	Pedido en local.....	6
4.2	Pedido a domicilio	6
4.3	Pedido de reserva	8
4.4	Confirmar orden.....	9
4.5	Cancelar orden	9
5	Perfil del administrador	9
5.1	Resumen de la vista administrador	9
5.2	Ordenes activas	10
5.3	Resumen de reservas del día	10
5.4	Pedidos.....	10
5.5	Reservas	11
5.6	Productos y categorías	11
5.7	Administrador de usuarios	12

1 Iniciar sesión



1. Nombre de usuario
2. Contraseña
3. Botón de iniciar sesión
4. Enlace para registrarse

2 Registrarse como cliente

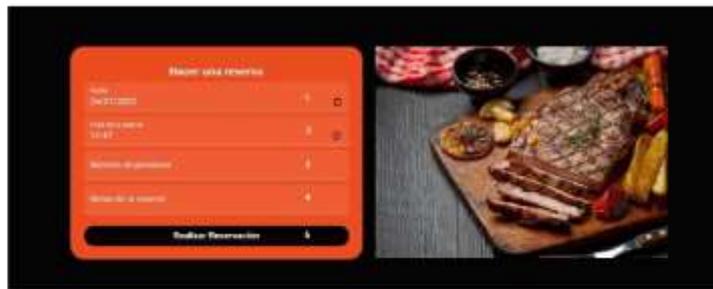


1. Correo electrónico
2. Teléfono
3. Nombre de usuario
4. Contraseña
5. Botón de registrarse
6. Enlace para iniciar sesión

3 Realizar una reservación

Para crear una reservación inicialmente se debe iniciar sesión, dado que, para registrar la reserva, se tiene que saber a qué usuario le pertenecerá la reservación, en el caso no exista una sesión activa el sistema redirigirá usuario a la pantalla de inicio de sesión. Otro aspecto que tomar en cuenta es que todos los campos están validados, es decir la fecha no puede ser menor a la actual, la hora será solo durante el horario de atención, el número de personas no puede ser mayor al aforo.

ANEXO E: MANUAL DE USUARIO DE RESPEDAPP



1. Fecha para cuando se desea la reservación (no debe ser menor a la fecha actual)
2. Horario de la reserva (debe funcionar en el horario de atención)
3. Número de personas (debe ser menor al aforo máximo)
4. Notas para la reservación (es opcional, aquí puede indicar algunos preparativos al restaurante)
5. Botón para realizar la reservación

Una vez que realizada la reservación, se muestra una alerta de confirmación, una vez que se cierra, se muestra otra alerta donde se le pregunta si desea agregar un pedido a la reservación actual, el proceso de pedidos se lo mostrará más adelante.



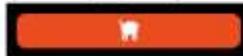
4 Realizar un pedido

Para realizar un pedido se parte desde la ruta de menú, donde se puede observar el listado de productos que se encuentran activos, en la siguiente pantalla que se muestra a continuación se explicará todas las funcionalidades de la pantalla de menú, los productos, las categorías, entre otras cosas. Así mismo como las reservaciones para continuar con el pedido se debe tener una sesión existente caso contrario el sistema redirigirá al usuario a la pantalla de iniciar sesión.

ANEXO E: MANUAL DE USUARIO DE RESPEDAPPP



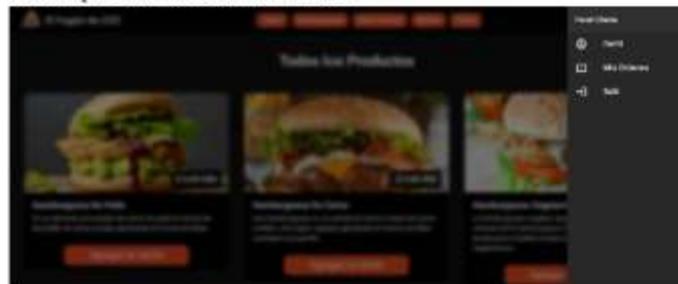
1. Enlace a la página inicial
2. Enlaces de las categorías registradas del menú
3. Botón que agrega el producto al carro, el cual tiene una animación, que indica que se agregó un producto



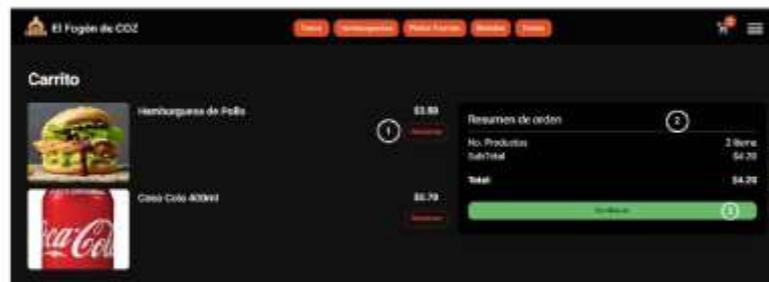
4. Botón que lleva al carrito de productos, aquí también se muestra la cantidad de ítems que se encuentra en el carrito de compras,



5. Botón que visualiza el menú lateral



La pantalla del carrito de compras tiene la siguiente descripción



1. Botón para remover ítems del carrito de compras
2. Resumen de la orden

ANEXO E: MANUAL DE USUARIO DE RESPEDAPP

3. Botón para continuar, las acciones de este botón dependerán, del rol del usuario en el caso de los meseros este redirige directo a pedidos locales, en el caso de cliente, se muestra un modal donde puede elegir si es de local o domicilio y en el caso de reservación lo redirige directo al pago de domicilio.

4.1 Pedido en local

En el caso de que sea cliente o administrador se le preguntará donde se encuentra ubicado



Para el resumen final de local se presenta una pantalla similar al resumen del carro de compras, la diferencia es que aquí se debe seleccionar una mesa y de ser necesario agregar una nota al pedido.



1. Número de orden
2. Resumen de la orden
3. Seleccionar mesa
4. Notas para la orden
5. Pago en línea (Se lo mostrará más adelante)
6. Confirmar orden
7. Cancelar orden

4.2 Pedido a domicilio

Para el caso de domicilio, esta acción no puede realizarse desde un usuario de tipo mesero, si es otro usuario de igual forma se pregunta donde se encuentra ubicado, de ahí este será redirigido hacia una pantalla para seleccionar las direcciones.

ANEXO E: MANUAL DE USUARIO DE RESPEDAPPP



A continuación, se presenta la pantalla donde se puede seleccionar direcciones de un listado del usuario, además, también puede agregar nuevas direcciones



1. Este icono identifica la dirección que está marcado como predeterminada
2. Este botón permite seleccionar la dirección de envío
3. Es el listado de direcciones que pertenecen al usuario registrado
4. Fomulario para agregar una nueva dirección
5. Botón para registrar la nueva dirección

Una vez que se selecciona la dirección se redirige al resumen final, donde se mostrará la dirección, número de orden, notas, etc.



1. Número de orden
2. Botón para editar la dirección actual
3. Resumen de orden
4. Notas para la orden
5. Pago en línea
6. Confirmar orden
7. Cancelar orden

4.3 Pedido de reserva

Para la reserva se parte desde la modal donde se le pregunta si se desea agregar un pedido a la reserva



Al igual que en los pedidos normales, se parte desde carrito de compras, al dar clic al botón continuar, se redirige al resumen final de la reserva.



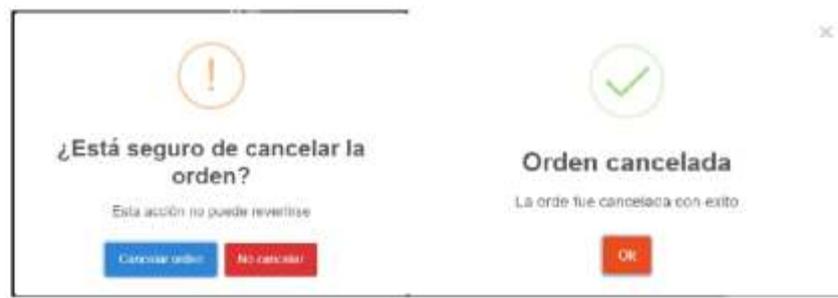
1. Número de orden
2. Resumen de la reservación
3. Resumen de la orden
4. Notas para la orden
5. Pago en línea
6. Confirmar orden
7. Cancelar orden

4.4 Confirmar orden



4.5 Cancelar orden

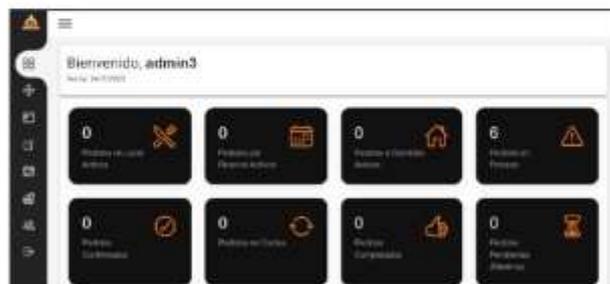
Modales que indican la cancelación de la orden



5 Perfil del administrador

5.1 Resumen de la vista administrador

En esta pantalla el administrador puede ver un resumen general de las reservaciones activas, canceladas, en proceso, de igual formas los pedidos, en proceso, finalizados, en preparación, entre otras cosas más.



5.2 Ordenes activas

En esta pantalla se puede ver todas las ordenes activas que se van registrando en tiempo real, esta dividido en contenedores según sus estados, los cuales pueden ir cambiando según su preparación, este proceso lo puede realizar el personal de cocina arrastrando y soltando los ítems.



5.3 Resumen de reservas del día

En esta pantalla el administrador puede visualizar todas las reservaciones que estén registradas para el día actual.



5.4 Pedidos

En esta pantalla el administrador puede ver todos los pedidos, los cuales están divididos por los tipos, en este caso: locales, domicilio, reservas y los de estado finalizado, además, en esta pantalla también el administrador puede cambiar el estado de los pedidos los cuales están definidos:

- En proceso
- Confirmado
- Preparando

ANEXO E: MANUAL DE USUARIO DE RESPEDAPP



Para el caso de productos de igual forma el administrador puede agregar los productos, que pertenezcan a una categoría, de igual forma el administrador puede agregar, editar o eliminar un producto.



5.7 Administrador de usuarios

En esta pantalla el administrador puede gestionar todos los usuarios del sistema, existen tres tipos de usuarios: Administrador, Meseros y Clientes, en esta pantalla el administrador puede agregar, editar y eliminar los usuarios de tipo mesero y administrador, en el caso de clientes solo puede eliminarlos, no podrá realizar ninguna otra acción.





esPOCH

Dirección de Bibliotecas y
Recursos del Aprendizaje

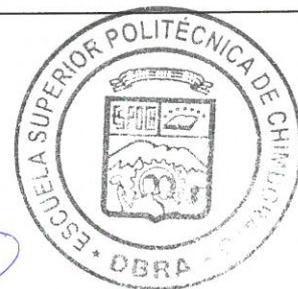
UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y
DOCUMENTAL

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 15 / 12 / 2022

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: Emily Micaela Chimbo Encalada, Angello Rafael Ordoñez Zapata
INFORMACIÓN INSTITUCIONAL
Facultad: Informática y Electrónica
Carrera: Ingeniería de Software
Título a optar: Ingeniero en Software
f. responsable: Ing. Cristhian Fernando Castillo Ruiz


DBRA
Ing. Cristhian Fernando Castillo



2352-DBRA-UTP-2022