



**Diseño e implementación de aplicación web para el control y gestión de órdenes en la
cafetería – heladería “San Ignacio” de la ciudad de Quito, mediante la ejecución de
servicios localhost.**

Uchuari Vélez, Kevin Leonel

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Trabajo de Integración Curricular, Previo a la Obtención del Título de Tecnólogo Superior en
Redes y Telecomunicaciones

Msc. Tintín Perdomo, Verónica Paulina

18 de Agosto del 2023

Latacunga

Copyleaks

Plagiarism report

Trabajo de Integracion Curricular Kev...

Scan details

Scan time:
August 14th, 2023 at 17:4 UTC

Total Pages:
52

Total Words:
12947

Plagiarism Detection



Types of plagiarism		Words
Identical	1.1%	148
Minor Changes	1.6%	208
Paraphrased	5%	651
Omitted Words	0%	0

AI Content Detection



Text coverage
 AI text
 Human text

Plagiarism Results: (54)

<p> TESIS_CI_ISO_01_2020_ET190336.pdf 1.1%</p> <p>https://bibliotecaunapec.blob.core.windows.net/tesis/tesis_ci...</p> <p>TEJADA GUZMAN, ISMELKA DECANATO DE INGENIERÍA E INFORMÁTICA ESCUELA DE INFORMÁTICA PROYECTO DE TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE:...</p>
<p> TESIS_CI_ISO_10_2019_ET190121.pdf 0.8%</p> <p>https://bibliotecaunapec.blob.core.windows.net/tesis/tesis_ci...</p> <p>TEJADA GUZMAN, ISMELKA DECANATO DE INGENIERÍA E INFORMÁTICA ESCUELA DE INFORMÁTICA PROYECTO DE TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE:...</p>
<p> ¿Qué es el Javascript? 0.8%</p> <p>https://latecnologia.online/que-es-el-javascript/</p> <p>Saltar al contenido Menú Menú Inicio Tecnología Inform...</p>

Firma:

Tintín Perdomo, Verónica Paulina

C. C.: 1802928398



Departamento de Eléctrica, Electrónica y Telecomunicaciones
Carrera de Tecnología Superior en Redes y Telecomunicaciones

Certificación

Certifico que el trabajo de integración curricular: "Diseño e implementación de aplicación web para el control y gestión de órdenes en la cafetería – heladería "San Ignacio" de la ciudad de Quito, mediante la ejecución de servicios localhost." fue realizada por el señor Uchuari Vélez Kevin Leonel, la misma que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisada y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se la sustente públicamente.

Latacunga, 18 de agosto de 2023

Firma:

Tintín Perdomo, Verónica Paulina

C. C.: 1802928398



Departamento de Eléctrica, Electrónica y Telecomunicaciones
Carrera de Tecnología Superior en Redes y Telecomunicaciones

Responsabilidad de Autoría

Yo, **Uchuari Vélez, Kevin Leonel**, con cédula de ciudadanía 1753023843, declaro que el contenido, ideas y criterios del trabajo de integración curricular: **Diseño e implementación de aplicación web para el control y gestión de órdenes en la cafetería – heladería “San Ignacio” de la ciudad de Quito, mediante la ejecución de servicios localhost.** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 18 de agosto de 2023

Firma:

Uchuari Vélez, Kevin Leonel

C.C.: 1753023843



Departamento de Eléctrica, Electrónica y Telecomunicaciones
Carrera de Tecnología Superior en Redes y Telecomunicaciones

Autorización de Publicación

Yo **Uchuari Vélez, Kevin Leonel** completos, con cédula de ciudadanía 1753023843, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Diseño e implementación de aplicación web para el control y gestión de órdenes en la cafetería – heladería “San Ignacio” de la ciudad de Quito, mediante la ejecución de servicios localhost.** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 18 de agosto de 2023

Firma:

Uchuari Vélez, Kevin Leonel

C.C.: 1753023843

Dedicatoria

El presente trabajo está dedicado especialmente para mis padres Segundo y Blanca, por ellos y para ellos, sus consejos y su apoyo tanto económico como emocional fueron una clave fundamental en mi desarrollo profesional, a mis hermanos para que vean un ejemplo a seguir y comprendan que con perseverancia y disciplina todo lo que uno se propone lo cumple.

Uchuari Vélez, Kevin Leonel

Agradecimiento

En primer lugar, doy gracias a Dios, por darme vida, salud e inteligencia en todo mi proceso académico, quiero agradecer a mis padres por su apoyo incondicional y sus palabras de aliento que día tras día fueron forjando mi camino, gracias a los profesores que me supieron enseñar mucho más allá de la materia, también quiero agradecer a la Universidad ya que sin su infraestructura, equipos y laboratorios los conocimientos adquiridos no serían los mismos.

Uchuari Vélez, Kevin Leonel

ÍNDICE DE CONTENIDOS

Carátula.....	1
Reporte de Verificación de Contenido	2
Certificación	3
Responsabilidad de Autoría	4
Autorización de Publicación	5
Dedicatoria.....	4
Agradecimiento.....	7
Índice de Contenidos	8
Índice de Tablas	12
Índice de Figuras	13
Resumen	15
Abstract.....	16
Capítulo I: El Problema	17
Antecedentes	17
Planteamiento del Problema	18
Justificación.....	19
Objetivos	20
<i>Objetivo General.....</i>	20
<i>Objetivos Específicos.....</i>	20
Alcance	20
Capítulo II: Marco Teorico.....	21

Arquitectura Cliente / Servidor	21
<i>Ventajas de una Arquitectura Cliente / Servidor</i>	<i>21</i>
<i>Separación de las Funciones en Capas</i>	<i>22</i>
Aplicaciones Web.....	22
<i>Definición</i>	<i>22</i>
<i>Tecnologías Web.....</i>	<i>23</i>
HTML.....	25
JavaScript	26
CSS	27
CSS y HTML.	28
<i>Frameworks</i>	<i>28</i>
<i>Frameworks de JavaScript.....</i>	<i>28</i>
Next.js.....	29
<i>Frameworks Front-End.....</i>	<i>29</i>
Bootstrap.....	29
Servidores Web	29
<i>Alojamiento de Archivos (Hosting).....</i>	<i>30</i>
Base de Datos.....	31
<i>Sistema de Gestión de Bases de Datos (SGBD)</i>	<i>31</i>
<i>Objetivos de un Sistema Gestor de Base de Datos</i>	<i>32</i>
MySQL	32

	10
Características de MySQL.....	33
Prestaciones	33
Limitaciones.....	34
Modelo Relacional.....	34
Metodologías	35
Metodología Ágil	35
Metodología SCRUM.....	36
Scrum vs Otras Metodologías	36
Fases de Scrum.....	37
Capítulo III: Desarrollo del Tema.....	40
Diseño e Implementación de la Aplicación	40
Estructuración del Product Backlog.....	40
Reunión y Planificación del Product Backlog.	41
Sprints del Product Backlog.....	41
Diseño de la Aplicación.....	50
Interfaces de Usuario.....	50
Implementación de la aplicación.....	60
Instalación y configuración del servidor local	60
Fase de pruebas	68
Desarrollo de Pruebas.....	69
Capítulo IV: Conclusiones y Recomendaciones.....	73

Conclusiones	73
Recomendaciones	74
Bibliografía	75
Anexos	78

ÍNDICE DE TABLAS

Tabla 1 <i>Plantilla para el registro de tareas a ejecutarse en el Product Backlog</i>	40
Tabla 2 <i>Sprint N. °1 Diseño y desarrollo página de inicio</i>	41
Tabla 3 <i>Sprint N. °2 Diseño y desarrollo página nueva orden</i>	42
Tabla 4 <i>Sprint N. °3 Diseño y desarrollo tabla registro de órdenes</i>	43
Tabla 5 <i>Sprint N. °4 Diseño y desarrollo categorías</i>	43
Tabla 6 <i>Sprint N. °5 Diseño y desarrollo sección productos</i>	44
Tabla 7 <i>Sprint N. °6 Diseño y desarrollo sección escoger sabor</i>	44
Tabla 8 <i>Sprint N. °7 Diseño y desarrollo sección sabores</i>	45
Tabla 9 <i>Sprint N. °8 Diseño y desarrollo carrito de compras</i>	46
Tabla 10 <i>Sprint N. °9 Diseño y desarrollo sección detalle venta</i>	46
Tabla 11 <i>Sprint N. °10 Diseño y desarrollo página ordenes pendientes</i>	47
Tabla 12 <i>Sprint N. °11 Diseño y desarrollo página ordenes por cobrar</i>	48
Tabla 13 <i>Sprint N. °12 Diseño y desarrollo página libro de órdenes</i>	48
Tabla 14 <i>Sprint N. °13 Diseño y desarrollo sección detalle de fecha</i>	49
Tabla 15 <i>Prueba N. °1</i>	69
Tabla 16 <i>Prueba N. °2</i>	69
Tabla 17 <i>Prueba N. °3</i>	70
Tabla 18 <i>Prueba N. °4</i>	70
Tabla 19 <i>Prueba N. °5</i>	71
Tabla 20 <i>Prueba N. °6</i>	71
Tabla 21 <i>Prueba N. °7</i>	72
Tabla 22 <i>Prueba N. °8</i>	72

ÍNDICE DE FIGURAS

Figura 1 <i>Página de inicio</i>	50
Figura 2 <i>Página Nueva Orden</i>	51
Figura 3 <i>Tabla de registro de pedidos y botón carrito de compras</i>	52
Figura 4 <i>Categorías de los productos</i>	52
Figura 5 <i>Catálogo de los producto</i>	52
Figura 6 <i>Botón Sabores</i>	53
Figura 7 <i>Sección Sabores</i>	53
Figura 8 <i>Botón guardar y botón cancelar</i>	54
Figura 9 <i>Producto escogido</i>	54
Figura 10 <i>Botón Sabores del producto</i>	54
Figura 11 <i>Campo de texto</i>	55
Figura 12 <i>Checkbox Llevar</i>	55
Figura 13 <i>Sección Escoger Sabor</i>	55
Figura 14 <i>Sabores escogidos</i>	56
Figura 15 <i>Botón carrito de compras</i>	56
Figura 16 <i>Sección Factura</i>	56
Figura 17 <i>Página Ordenes Pendientes</i>	57
Figura 18 <i>Sección Ordenes Pendientes</i>	57
Figura 19 <i>Página Ordenes por Cobrar</i>	58
Figura 20 <i>Sección Ordenes por Cobrar</i>	58
Figura 21 <i>Página Libro de Ordenes</i>	59
Figura 22 <i>Sección Detalle de Fecha</i>	59
Figura 23 <i>Equipo para la instalación del servidor local</i>	60

Figura 24 <i>Especificaciones técnicas del equipo.</i>	60
Figura 25 <i>Instalación del servidor Node Js en el equipo.</i>	61
Figura 26 <i>Verificación de la versión de Node instalada en el equipo.</i>	61
Figura 27 <i>Instalación del gestor de base de datos.</i>	62
Figura 28 <i>Modelo Relacional de la base de datos.</i>	62
Figura 29 <i>Script de la tabla productos.</i>	63
Figura 30 <i>Script de la tabla categorías.</i>	63
Figura 31 <i>Script de la tabla sabores.</i>	63
Figura 32 <i>Script de la tabla facturas.</i>	64
Figura 33 <i>Script de la tabla órdenes.</i>	64
Figura 34 <i>Creación de una aplicación Next Js.</i>	64
Figura 35 <i>Pool de conexión a la base de datos.</i>	65
Figura 36 <i>Código para la obtención de productos.</i>	65
Figura 37 <i>Código para filtrar los productos.</i>	66
Figura 38 <i>Código para registrar pedidos.</i>	66
Figura 39 <i>Código para registrar las facturas.</i>	67
Figura 40 <i>Código para obtener los sabores.</i>	68

Resumen

Para muchos negocios que ofrecen el servicio de alimentos, la atención y el servicio al cliente es una parte fundamental para garantizar una buena valoración por parte de las personas que los visitan. Mantener un servicio de primera y que sea en el menor tiempo posible ha hecho que muchos negocios de comida obtén por soluciones tecnológicas que ayude a todo su personal a cumplir este objetivo. Esto explica además la evolución y sofisticación que han seguido los protocolos de servicio al cliente, que se han ido adaptando al paso del tiempo y al avance de las tecnologías. Este nuevo punto de vista es el día a día de la cafetería – heladería “San Ignacio” que, como principal punto de unión y recreación familiar del sector, opto por un sistema que se encarga de recoger y gestionar cada una de las órdenes generadas por sus clientes para ofrecer un servicio mucho más rápido y de mejor calidad. Mantener todos sus datos y registros siempre a la mano y sin ninguna restricción a echo que este negocio opte por implementar un “Servidor Local” (Localhost) que alberga la base de datos de todos sus registros y la aplicación web que se encarga de gestionar todos estos registros. En resumen, un sistema que permita automatizar el servicio al cliente haciendo que este sea mucho más rápido y de igual o mejor calidad, beneficiando tanto a los clientes que visitan el local como a sus trabajadores.

Palabras clave: Servidor local, Automatización de servicios, Aplicación Web

Abstract

For many businesses that offer food service, attention and customer service are a fundamental part to ensure a good rating from the people who visit them. Maintaining a first-class service and doing it in the shortest time possible has led many food businesses to opt for technological solutions that help all their staff to meet this objective. This also explains the evolution and sophistication that customer service protocols have followed, which have been adapting to the passage of time and the advancement of technologies. This new point of view is the daily life of the "San Ignacio" café-ice cream shop, which, as the main point of union and family recreation in the sector, opted for a system that collects and manages each of the orders generated by its customers to offer a much faster and higher quality service. Keeping all their data and records always at hand and without any restriction has led this business to opt for implementing a "Local Server" (Localhost) that hosts the database of all its records and the web application that manages all these records. In summary, a system that allows automating customer service making it much faster and of equal or better quality, benefiting both the customers who visit the premises and their workers.

Keywords: Localserver, Service Automation, Web Application.

Capítulo I

El Problema

Antecedentes

Muchos locales de comida hoy en día no poseen un sistema que les permita tomar la orden de un cliente y enviársela automáticamente al área de cocina para su preparación, utilizan hojas de papel o libretas para anotar la orden de un cliente, luego deben entregar esa orden al área de cocina, área que mucha de las veces está saturada de pedidos y con varios retrasos ya que la mayoría de estas órdenes estas ilegibles o se extravían entre todos los pedidos que están pendientes, una vez preparado y entregado el pedido el área de cocina envía esa orden a la caja donde se registra la orden y el total a cancelar, todo esto de forma manual. Esto se ha convertido en un problema para muchos negocios ya que poseen registros desactualizados y poco veraces, que no aportan resultados reales, de igual forma los conflictos y retrasos en la entrega de los pedidos hace que se genere cierto disgusto por parte de los clientes que muchas de las veces se quejan por el mal servicio o ya no regresan.

Alba Fuentes (2013) en su proyecto de investigación cuyo tema es “Diseño e implementación de una aplicación Android para la gestión de pedidos en restaurantes”, manifiesta que es esencial que los restaurantes implementen una aplicación que les permita gestionar las ordenes de los clientes.

Jonathan Hernández y Juan Ferro (2017) en su proyecto de investigación con el tema “Desarrollo de un Aplicativo Web para la Facturación, Control de Kardex y Apoyo al Manejo de Gastos de la Empresa EMPANADAS DE MI TIERRA Y PRODUCTOS S.A.S.” manifiestan que las aplicaciones web son percibidas como una opción de tecnología de la información que se ajusta a las necesidades específicas, siendo idealmente utilizadas como una herramienta de negocios que facilita la comunicación digital y brinda un servicio a sus usuarios o clientes. Además, las aplicaciones web permiten la automatización y mejora de procesos, simplificando así la vida de muchas empresas, organizaciones e instituciones.

Planteamiento del Problema

Para los negocios de comida satisfacer las necesidades de los clientes, es sin duda, la misión más importante, pero al igual que cualquier empresa de ámbito comercial, se ven afectados por un sin número de problemas que requieren de soluciones rápidas y eficientes que satisfagan sus necesidades y las de sus clientes, en la actualidad la tecnología se ha convertido en una herramienta fundamental en cualquier negocio, grande o pequeño, gracias a los avances de la tecnología y a su fácil accesibilidad, ha hecho que muchos negocios de comida opten por soluciones tecnológicas que los ayuden a automatizar sus procesos de facturación y servicio al cliente, entregando siempre la mejor experiencia al cliente con una atención de primera e innovadora.

La cafetería – heladería “San Ignacio” se creó en el año 2017 con la misión de brindar un lugar acogedor, deportivo y familiar, en donde las personas puedan reunirse y pasar un momento agradable acompañados de sus exquisitos helados de paila. Su visión, convertirse en el principal punto de reunión familiar y deportivo del sector.

La cafetería – heladería “San Ignacio”, posee de un variado menú que busca abarcar todo lo que una cafetería y heladería debe poseer, para ofrecer a sus clientes varias opciones a elegir, sin embargo en los últimos meses han tenido un incremento de clientes y para su personal se ha complicado llevar un control estricto en la contabilidad y las órdenes de los clientes ya que cada orden de un cliente se genera de forma manual, escribiéndola en un libro, en donde se calcula el valor total de la orden y se da aviso al área de cocina para su preparación, el personal de cocina debe revisar constantemente el libro de pedidos para asegurarse que los productos son los correctos, muchas de las veces las órdenes están ilegibles y no se pueden distinguir los pedidos o no se pierden entre todas las demás órdenes; lo que genera confusiones o retrasos en el área de cocina, hacen que el cliente espere por más tiempo generando disgustos por parte de este y de igual forma al momento de realizar la contabilidad diaria al final del día, los valores muchas de la veces no coinciden.

Por lo cual, frente al aumento de clientes, retrasos en pedidos e inconsistencias en los registros contables que se ha presentado los últimos meses en la cafetería – heladería “San Ignacio” de la ciudad de Quito, se propone la implementación de una aplicación web que ayude al control, registro y contabilidad de cada una de las órdenes registradas por los clientes, mediante la ejecución de un servidor local con el fin de llevar un mejor control en sus registros y que su servicio sea más rápido, de mejor calidad y todo el personal sea más eficiente.

Justificación

El presente trabajo tiene la finalidad de automatizar y facilitar los procesos para la gestión y el registro de las órdenes de clientes, mediante una aplicación web que se ejecuta en un servidor local (localhost), que permitirá controlar y organizar cada una de las órdenes generadas por los clientes, para ofrecer un mejor servicio por parte del personal de la cafetería - heladería “San Ignacio”, en la actualidad “San Ignacio” no cuenta con sistema que le permita gestionar y llevar un registro contable de las órdenes de sus clientes, lo que genera retrasos en la entrega de los pedidos e inconsistencias en los valores facturados.

La importancia de implementar un sistema que permita controlar y gestionar las órdenes de los clientes de “San Ignacio” mediante la ejecución de servicios localhost, es que cada una de las órdenes podrá ser visualizada por todo el personal, agilizando así la preparación de los pedidos, se podrá obtener un registro permanente y legible de todas las órdenes, que ayudará a llevar una administración y control de los pedidos e ingresos del negocio.

Los beneficiarios de este proyecto serán el personal de servicio y de cocina, ya que por medio de la implementación de la aplicación podrán tomar las órdenes de los clientes sin la necesidad de avisarle al área de cocina, área que tiene acceso y puede visualizar todas las órdenes que se generan, de igual forma al realizar la contabilidad diaria el sistema ofrece el valor total de los ingresos obtenidos en el día, lo que ayuda a llevar un registro diario de todas las ventas.

Objetivos

Objetivo General

Diseñar e implementar una aplicación web que controlé y gestioné las órdenes de la cafetería – heladería “SAN IGNACIO” de la ciudad de Quito, mediante la ejecución de servicios localhost, para brindar al cliente un servicio más eficiente y de mayor calidad, en el menor tiempo posible.

Objetivos Específicos

- Integrar la aplicación de control y gestión de pedidos al área de producción de alimentos para que las órdenes sean visualizadas y preparadas de la misma forma en la que fueron registradas, evitando el retraso de los pedidos.
- Automatizar el registro permanente de las órdenes de los clientes con la utilización de una base de datos, para que puedan ser consultadas en cualquier momento.
- Establecer servicios para albergar y registrar las órdenes de la aplicación web utilizando la red interna del local, con el fin de poder acceder a los registros de las órdenes sin necesidad de tener acceso a internet.

Alcance

El presente proyecto tiene como propósito realizar el diseño y desarrollo de una aplicación web que ayude a controlar y gestionar las órdenes de los clientes de la cafetería - heladería “San Ignacio”. La aplicación permitirá a los empleados de San Ignacio realizar las siguientes tareas:

- Realizar el registro de nuevas órdenes.
- Generación de reporte de ventas.
- Registrar ordenes cobradas o pendientes por cobrar.
- Visualizar ordenes pendientes por entregar y por cobrar.
- Eliminar ordenes pendientes por entregar y por cobrar.

Capítulo II

Marco Teórico

Arquitectura Cliente / Servidor

El cliente es la parte de la aplicación que se ejecuta en el dispositivo del usuario y se encarga de presentar los datos y proporcionar la interfaz de usuario. Por otro lado, el servidor es el componente central que aloja la información y realiza los procesos necesarios para cumplir con las solicitudes del cliente.

Según Lujan (2002) en la arquitectura de red cliente-servidor, cada equipo, dispositivo o proceso conectado a la red dependiendo de su rol es designado como cliente o servidor, esto dependiendo de la función que lleva a cabo en la red. Además, encontró que generalmente, los servidores son equipos con una capacidad de procesamiento superior a la de un equipo convencional, estos equipos están destinados a manejar unidades de almacenamiento, impresoras, tráfico de red, información o aplicaciones. En cambio, los clientes son equipos básicos para realizar tareas cotidianas, son dispositivos que la mayoría de personas posee, que acceden y utilizan los recursos que brindan los servidores como por ejemplo las redes sociales, correo electrónico, plataformas de streaming, etc.

Lujan indica que en una aplicación cliente-servidor, el cliente envía una solicitud al servidor y este responde con los datos requeridos o con una indicación de error en caso de que no pueda satisfacer la solicitud. La arquitectura cliente-servidor es ampliamente utilizada en la web, donde los navegadores web funcionan como clientes y los servidores web alojan los sitios web y responden a las solicitudes de los clientes.

Ventajas de una Arquitectura Cliente / Servidor

Flores (1995) describe algunas de las ventajas de la arquitectura cliente servidor:

- Ahorro en dinero.
- Incremento de la productividad.
- Tienen mejor escalabilidad.

- Flexibilidad.
- La arquitectura cliente/servidor es modular.
- Utilización de recursos.
- Control centralizado.
- Sistemas Abiertos.

Separación de las Funciones en Capas

Flores menciona que la arquitectura cliente/servidor nos permite separar las funciones en tres niveles o capas:

- Capa de presentación; es la encargada de comunicar la aplicación con el usuario. Sus tareas principales son recibir la información del cliente, enviarla a la capa de acceso a datos para que la información sea procesada, recibir los resultados, y mostrarlos al usuario.
- Capa de aplicación; la capa de aplicación se encarga de administrar los datos en la fase de procesamiento. Es el enlace entre el cliente y los datos. Sus funciones principales son recibir los datos desde la capa de presentación, enviarla a la capa de acceso a datos para cumplir con las reglas de negocio de la aplicación (como por ejemplo registro de usuarios, facturación, inventario, etc.), y enviar todos los resultados a la capa de presentación.
- Capa de acceso a datos; es la encargada de gestionar los datos en el nivel de almacenamiento. Sus funciones incluyen guardar la información, recuperarla, mantenerla y resguardar su integridad.

Aplicaciones Web

Definición

Según Lerma-Blasco Murcia y Talón (2013) la arquitectura de las aplicaciones web se basa en el modelo cliente-servidor con máquinas interconectadas a través de Internet o una red interna. Surgió en la década de 1990 con la Web 1.0, permitiendo el acceso a la web desde dispositivos conectados mediante conexiones conmutadas; las páginas web se construyen utilizando HTML,

CSS y JavaScript para agregar estilos y funcionalidades interactivas en el lado del cliente. En la Web 1.5, se introdujeron animaciones Flash. Las páginas web dinámicas, escritas en lenguajes de programación para el servidor como PHP, ASP.net o JSP, generan código HTML de forma dinámica en el servidor, personalizando el contenido según la solicitud o consultas a bases de datos. La arquitectura ha evolucionado hacia la Web 2.0 y tecnologías más avanzadas.

Al ser accesibles desde cualquier lugar y en cualquier momento, las aplicaciones web permiten a los usuarios ahorrar tiempo y aumentar su productividad. Además, debido a su naturaleza distribuida, las aplicaciones web ofrecen una gran flexibilidad y escalabilidad, lo que las hace adecuadas para una amplia variedad de usos, desde pequeñas empresas hasta grandes corporaciones.

Tecnologías Web

Las tecnologías web continúan evolucionando y desempeñando un papel importante en la forma en que las personas interactúan con la información y realizan sus tareas diarias. Sin duda, su impacto continuará siendo importante en el futuro.

Según Capdevila (2001) las tecnologías Web son ampliamente utilizadas para acceder a recursos de conocimiento en internet o intranet a través de un navegador. Esto se debe a su facilidad de uso, flexibilidad y capacidad de imitar la forma en que las personas se relacionan. Además, encontró que estas tecnologías también pueden proporcionar recursos estratégicos personalizados a través de sistemas de Gestión del Conocimiento (GC) propietarios de la empresa; la World Wide Web (WWW) permite a los usuarios acceder a una amplia gama de información y navegar fácilmente de un recurso a otro. También señala que, dentro de estas tecnologías, se incluyen agentes inteligentes, chat, motores de búsqueda, navegadores y tecnologías push.

- Agentes inteligentes; los agentes inteligentes son programas que buscan información relevante para nosotros en Internet y otras fuentes, y nos la presentan en un formato fácil de usar para una utilización directa e inmediata.

- Chats; el chat es una tecnología de conversación en tiempo real a través de texto, audio o video entre usuarios conectados a Internet. Las discusiones se llevan a cabo en canales temáticos y su principal ventaja es la compartición y distribución de conocimiento.
- Correo electrónico; el correo electrónico es una herramienta crítica para la comunicación y el trabajo en equipo, ampliamente utilizada en todas las organizaciones. Hay sistemas de correo electrónico que permiten enviar mensajes de voz y también hay sistemas que convierten texto en mensajes de voz para ser escuchados en dispositivos móviles.
- Motores de búsqueda; los motores de búsqueda son herramientas que ayudan a localizar documentos en una gran cantidad de información, pudiendo hacer búsquedas simples o complejas. En la actualidad, estos motores son capaces de indexar y buscar en más de 2 mil millones de páginas en un abrir y cerrar de ojos.
- Navegadores Web; los navegadores son software creados para visualizar páginas web y tener acceso a Internet con una interfaz gráfica que presenta texto, gráficos, audio, vídeo e incluso olores; no se consideran una herramienta de importancia en sí mismos para la Gestión de la información, sino más bien son una tecnología básica que proporciona el medio para la creación de otras tecnologías como las intranets, el correo electrónico y el chat.
- Tecnología Push; es una herramienta que proporciona información de manera anticipada en lugar de requerir que el usuario la busque. El usuario especifica qué tipo de información desea recibir y el software encuentra y notifica al usuario con una señal. Esto transforma la participación del usuario de ser activo en la búsqueda de información a recibir información de fuentes que son proactivas.

Entre las tecnologías más importantes utilizadas para el desarrollo de aplicaciones web se encuentran HTML, CSS y JavaScript; estas tecnologías se las utiliza para el diseño y la estructuración de aplicaciones web, en cambio Python, Ruby on Rails y PHP se utilizan para el

desarrollo de aplicaciones web. También, existen muchas plataformas y frameworks de desarrollo, como Next, React, Angular y Vue, que han logrado simplificar y agilizar el proceso de desarrollo.

HTML. Según Vertice (2009) HTML es un lenguaje fundamental en el desarrollo de páginas web y es esencial para cualquier persona que quiera crear o trabajar en el desarrollo de aplicaciones en línea; HTML es un lenguaje de programación que los computadores pueden interpretar y está diseñado para que los desarrolladores escriban instrucciones que los navegadores pueden ejecutar para mostrar una página web; las siglas HTML significan Hyper Text Markup (lenguaje marcado de hipertexto). En una computadora el hipertexto posee referencias como hipervínculos, link o parámetros. El hipertexto puede estar formado de texto, tablas, videos, archivos u otros elementos. Este concepto surgió por la necesidad de un método que permita procesar sistemáticamente la información entre computadoras conectadas a la red.

Vertice señala también, que HTML es un lenguaje que nos permite estructurar el documento por ejemplo el título, párrafos, etiquetas, imágenes, tablas, etc., pero no ayuda a describir la apariencia o los diseños del documento, sino que ofrece todas las herramientas para dar un formato al documento, según las capacidades de procesamiento del servidor web en el cual esta albergada la página web y las capacidades del navegador. Además, encontró que por esta razón no se debe tratar de dar diseños a la página desde el lenguaje de marcado, es mejor centrarse en estructurar con claridad y precisión el documentó para que resulte fácil de leer y entender. Además, muestra que HTML como lenguaje de programación tiene dos grandes ventajas que lo hacen imprescindible a la hora de desarrollar una aplicación web:

- Compatibilidad; es ejecutable en todos los navegadores web.
- Fácil aprendizaje; debido a su reducido número de etiquetas, es muy intuitivo y fácil de aprender.

JavaScript. Según Capdevila (2001) el lenguaje de programación JavaScript se utiliza principalmente para el desarrollo de páginas y aplicaciones web dinámicas. Menciona que una página o aplicación web dinámica es aquella que ofrece efectos en la paginas, animaciones, cualquier acción que se ejecuta al presionar un botón o las ventanas emergentes de aviso y de alerta; se señala también que JavaScript es un lenguaje de programación interpretado, por esta razón no necesita ser compilado para ejecutar los programas; es decir, las aplicaciones desarrolladas en JavaScript pueden ser probados directamente en cualquier navegador sin la necesidad de algún proceso intermedio. También aclara que JavaScript es un lenguaje de programación de alto nivel que se utiliza ampliamente en el desarrollo web para agregar interactividad y dinamismo a las páginas web. A continuación, nos muestra algunas de las principales ventajas de JavaScript:

- Interactividad en la página web; ayuda a los desarrolladores agregar elementos interactivos a las páginas web, como formularios, juegos, animaciones, etc.
- Validación de formularios en el lado del cliente; permite validar los formularios en el lado del cliente antes de enviarlos al servidor, lo que mejora la experiencia de usuario y reduce la carga en el servidor.
- Mejora de la accesibilidad; JavaScript puede utilizarse para mejorar la accesibilidad de un sitio web, como agregar descripciones de audio a imágenes o mejorar la navegación con teclado.
- Interacción con el usuario; permite a los desarrolladores interactuar con los usuarios a través de diálogos y alertas, lo que mejora la comunicación y la experiencia de usuario.
- Mejora de la experiencia de usuario; JavaScript permite a los desarrolladores crear páginas web más atractivas y dinámicas, lo que mejora la experiencia de usuario y aumenta la retención de visitantes en un sitio web

Capdevila (2001) también señala que JavaScript es ampliamente adoptado en el desarrollo web debido a su carácter de lenguaje de programación de alto nivel; Tiene la capacidad de añadir interactividad, validar formularios, mejorar la accesibilidad, interactuar con los usuarios y mejorar la experiencia de usuario, lo que lo convierte en una herramienta valiosa para cualquier desarrollador web. Además, su compatibilidad con una amplia variedad de navegadores y plataformas lo hace fácil de usar y accesible; en general, JavaScript es un componente esencial para cualquier sitio web moderno y su importancia continuará en el futuro del desarrollo web.

CSS. Durango (2015) expresa que; CSS define los estilos para los elementos HTML, como colores, fuentes, bordes, tamaños de texto y otros aspectos visuales; los estilos se aplican a los elementos HTML mediante reglas CSS, que se escriben en un archivo externo o dentro de la misma página HTML; con CSS, los desarrolladores pueden modificar el diseño de una página web de manera rápida y eficiente, sin tener que cambiar el contenido de la página. nos indica que CSS por sus siglas “Cascading Style Sheets” (hojas de estilo en cascada), es el formato recomendado para dar estilos de diseño a las paginas y aplicaciones web, este estándar fue publicado por el Worl Wide Web Consortium (W3C). Además, encontró que la implementación de este estándar en el desarrollo web nos permite ahorrar tiempo, proporciona coherencia y simplifica significativamente la creación de páginas web; Además, señala que CSS ofrece varias ventajas en el desarrollo web, incluyendo:

- Desvinculación entre estilos y contenido; CSS simplifica el mantenimiento y la adaptación de un sitio web al separar la apariencia de su contenido.
- Incremento de la accesibilidad; CSS posibilita el mejoramiento de la accesibilidad de un sitio web al brindar la capacidad de ajustar el tamaño de la tipografía según las exigencias de un usuario.
- Consistencia en la apariencia; CSS permite aplicar estilos coherentes a través de todo un sitio web, lo que mejora la apariencia y la experiencia de usuario.

- Reducción del tamaño del archivo HTML; al separar la presentación de la estructura, se puede reducir el tamaño de los archivos HTML, lo que mejora el rendimiento y la velocidad de carga del sitio web.
- Mejora de la compatibilidad entre navegadores; CSS permite mejorar la compatibilidad entre navegadores, lo que significa que un sitio web puede tener una apariencia consistente en diferentes navegadores

Durango (2015) concluye que CSS es una herramienta esencial en el desarrollo de páginas web modernas y permite a los desarrolladores crear páginas atractivas y profesionales con una gran variedad de opciones de estilo; al separar la información de contenido y presentación, CSS también facilita la mantenibilidad y escalabilidad de los sitios web.

CSS y HTML. Tinoco (2014) muestra que para entender mejor las funciones clave de HTML y CSS, imaginemos que estamos construyendo un edificio. Las columnas y paredes del edificio representarían la estructura de una página web desarrollada con HTML, mientras que la presentación del edificio simbolizaría la separación del diseño del contenido en la página web, lo que hace que sea más sencillo mantener y manipular el sitio web; también, señala que HTML y CSS son lenguajes complementarios que se utilizan juntos para crear páginas web; HTML proporciona la estructura y el contenido de una página web, mientras que CSS se encarga de su presentación y diseño.

Frameworks

Sánchez Acosta, Tuesta Monteza y Mejía Cabrera (2015) indican que un framework es un conjunto de herramientas y componentes preestablecidos que proporcionan un marco de trabajo para desarrollar software. Sirve como base para crear aplicaciones, organizar el código y acelerar el proceso de desarrollo. Los frameworks ofrecen una estructura predefinida y un conjunto de reglas que los desarrolladores pueden utilizar para crear aplicaciones más eficientemente.

Frameworks de JavaScript

Next.js. Next.js es un marco de JavaScript de código abierto para el desarrollo de aplicaciones web de servidor. Se basa en React y ofrece una serie de características avanzadas para el desarrollo de aplicaciones web escalables y de alto rendimiento. Kinsta (2022) aclara que Next.js es un framework de desarrollo que te permite desarrollar aplicaciones web completas y amigables integrando poderosas herramientas de JavaScript. Además, encontró que este framework es utilizado por algunas de las empresas más grandes del mundo y es conocido por brindar la mejor experiencia a los desarrolladores al crear aplicaciones listas para producción con todas las funciones necesarias.

Frameworks Front-End

Bootstrap. Escobar, Alvarez y Alvarez (2019) indican que Bootstrap es un marco de diseño de código abierto para desarrollar aplicaciones web y móviles. Fue creado por Twitter y es uno de los marcos más populares y ampliamente utilizados en el desarrollo web, además, señalan que Bootstrap es una librería de código abierto para diseño web que ofrece plantillas y herramientas para HTML, CSS y JavaScript. También encontraron que Bootstrap incluye elementos como tipografía, formularios, botones, tablas, menús de navegación y otros componentes de diseño web. Indican que a diferencia de otros frameworks, Bootstrap se enfoca solo en el desarrollo del front-end de una aplicación o sitio web.

Servidores Web

Corporation's (2022) indica que un servidor web es un programa informático o un dispositivo de hardware que concede entrada a material web por medio de internet. Cuando un usuario ingresa a un sitio web, envía una petición a un servidor web, el cual a su vez responde con el contenido de la página en formato HTML al navegador del usuario.

- Desde el punto de vista del hardware, un servidor web es un ordenador que guarda el software de servidor web y los ficheros que forman un sitio web (como documentos HTML, imágenes, hojas de estilo CSS y archivos JavaScript); este servidor web conectado a

internet se encarga de mantener la comunicación de datos con otros dispositivos conectados a la red.

- Desde el punto de vista del software, un servidor web consta de varios componentes que controlan cómo los usuarios de la web acceden a los archivos alojados en él. Estos componentes incluyen, al menos, un servidor HTTP. Un servidor HTTP es un software que puede comprender URLs (direcciones web) y HTTP (el protocolo que utiliza tu navegador para obtener páginas web). Este servidor HTTP se puede acceder a través de los nombres de dominio de los sitios web alojados en él y proporciona el contenido de esos sitios web a los dispositivos de los usuarios finales

Corporation's (2022) indica que para la publicación de una página o aplicación web es necesario implementar un servidor web estático o dinámico.

- Un servidor web estático, también conocido como "pila", es una combinación de hardware en forma de una computadora y software en forma de un servidor HTTP; este tipo de servidor se le llama "estático" porque simplemente envía los archivos almacenados sin realizar ningún tipo de modificación a su navegador.
- Un servidor web dinámico se compone de un servidor web estático con software adicional, que normalmente incluye una aplicación servidor y una base de datos; este tipo de servidor es llamado "dinámico" debido a que la aplicación servidor actualiza los archivos alojados antes de enviarlos a través del servidor HTTP al navegador

Alojamiento de Archivos (Hosting)

Corporation's (2022) indica que el hosting es un servicio que permite a los dueños de sitios web alojar sus páginas en internet para que puedan ser accedidas por los usuarios a través de un navegador web. También, señala que técnicamente, es posible almacenar todos esos archivos en una computadora, pero es más cómodo guardarlos en un servidor web dedicado que:

- Está en funcionamiento en todo momento.

- Está conectado a Internet en todo momento.
- Tiene la misma dirección IP todo el tiempo.
- Es mantenido por un proveedor externo.

A causa de estos elementos, localizar un proveedor de hosting confiable se convierte en un componente fundamental en la creación de su página web. Es fundamental examinar las distintas alternativas proporcionadas por las empresas y seleccionar aquella que concuerde más con las necesidades y recursos disponibles, ya que los servicios varían desde alternativas gratuitas hasta tarifas que sobrepasan los miles de dólares cada mes.

Base de Datos

Pisco Gómez (2017) define a una base de datos como una colección organizada y estructurada de datos que representan una realidad objetiva y están organizados de manera independiente de las aplicaciones. Esto significa que pueden ser utilizados y compartidos por diferentes usuarios y aplicaciones. En otras palabras, una base de datos puede considerarse una colección de datos que pueden variar con el tiempo; también indica que los programas de aplicación funcionan con los datos guardados en la base de datos a través de las herramientas que brindan los sistemas de gestión de bases de datos. Estos sistemas, en la mayoría de los casos, cuentan con lenguajes específicos para manipular la información que hacen más sencillo el trabajo de los usuarios.

Sistema de Gestión de Bases de Datos (SGBD)

Marqués (2011) define a un sistema de gestión de bases de datos como una herramienta que permite a los usuarios establecer, crear y mantener la base de datos, proporcionando también un acceso regulado a los datos. Este sistema está compuesto por la base de datos, el sistema de gestión de bases de datos y los programas de aplicación que brindan servicios a la empresa u organización. Además, aclara que el enfoque utilizado en los sistemas de bases de datos es similar al usado en la programación orientada a objetos actual, en la que se divide en una

implementación interna de un objeto y una definición externa separada. Los usuarios del objeto solo ven su descripción externa y no tienen que preocuparse por cómo se lleva a cabo su implementación interna.

Objetivos de un Sistema Gestor de Base de Datos

Cabello (2010) señala que el SGBD es crucial para garantizar la seguridad y la integridad de los datos. Para lograr esto, proporciona a los usuarios diversas herramientas, incluyendo:

- Herramientas para definir y establecer la estructura de la base de datos y especificar los datos en ella.
- Herramientas para gestionar y crear la estructura de almacenamiento necesaria en los dispositivos de almacenamiento.
- Herramientas para el manejo y control de la información almacenada en las bases de datos, incluyendo la capacidad de agregar, cambiar, eliminar o buscar datos.
- Herramientas para restaurar los datos en caso de un evento desastroso.
- Herramientas para la generación de copias de respaldo.
- Herramientas para administrar y coordinar las conexiones y las transferencias de información con la base de datos.
- Herramientas para desarrollar aplicaciones basadas en los diseños externos de los datos.
- Herramientas para la configuración y puesta en marcha de la base de datos.
- Herramientas para la transferencia de datos hacia y desde la base de datos

MySQL

Arias (2014) indica que MySQL es una base de datos relacional que utiliza el lenguaje SQL (Lenguaje de Consulta Estructurado). Es un Sistema de Gestión de Bases de Datos de código abierto lanzado en 1995 y posteriormente adquirido por Sun Microsystems en 2008, y posteriormente por Oracle en 2009. También señala que MySQL es uno de los sistemas de gestión de bases de datos más populares y ampliamente utilizados en todo el mundo,

especialmente en aplicaciones web y aplicaciones empresariales. Además, encontró que MySQL permite almacenar, organizar y acceder a grandes cantidades de información de manera eficiente y segura. También proporciona una amplia gama de herramientas de administración y consulta para ayudar a los desarrolladores a trabajar con sus datos de manera efectiva.

Características de MySQL

Casillas, Ginestá y Mora (2014) describen las características que hacen destacar a este Sistema Gestor de Bases de Datos (SGBD), junto con las áreas en las que presenta problemas de diseño, limitaciones o partes del estándar que aún no han sido incorporadas.

Prestaciones. Casillas, Ginestá y Mora (2014) expresan que MySQL ha ganado popularidad debido a una serie de características atractivas:

- Está diseñado en C/C++.
- Se ofrecen ejecutables para alrededor de diecinueve plataformas diferentes.
- La interfaz de programación de aplicaciones (API) está disponible en los lenguajes Java, Perl, PHP, Python, C, C++, Eiffel, Ruby y TCL.
- Está diseñado para funcionar de manera optimizada en equipos con múltiples procesadores.
- Su tiempo de respuesta es particularmente impresionante.
- Puede ser usado tanto como un sistema cliente-servidor o incorporado en aplicaciones.
- Tiene una amplia variedad de tipos de datos disponibles.
- Ofrece varias opciones de almacenamiento para las tablas, cada una con sus propias características y rendimiento, para que se pueda adaptar a cada situación específica.
- Su gestión se realiza a través de la asignación de usuarios y permisos.
- Se han documentado casos de MySQL manejando grandes volúmenes de información, incluyendo 50 millones de registros, 60,000 tablas y 5 millones de columnas.

- Hay diferentes formas de conectarse a MySQL, incluyendo a través de TCP/IP, sockets UNIX y sockets NT, y también cuenta con un soporte completo para ODBC.

Limitaciones. Al conocer los conceptos detrás de su diseño, se puede entender mejor por qué algunas de sus limitaciones existen.

- No cuenta con la capacidad de manejar procedimientos almacenados (que estarán disponibles en la versión 5.0 próxima a lanzarse).
- El soporte de disparadores no está presente actualmente en MySQL, pero se espera que sea añadido en la próxima versión 5.0.
- No se incluyen vistas en la versión actual, pero están previstas para ser incluidas en la próxima versión 5.0.
- No posee capacidades de objetos como tipos de datos personalizados, herencia, entre otros.

Modelo Relacional

Según Ricardo (2009) INGRES fue uno de los primeros proyectos de investigación sobre el modelo relacional que fue desarrollado en la Universidad de California en Berkeley alrededor de la misma época que el proyecto System R. La investigación resultó en una versión universitaria de INGRES y también en un producto comercial. ORACLE fue creado y comercializado utilizando muchos de los resultados obtenidos en System R. Algunos de los primeros sistemas de gestión de bases de datos relacionales basados en microcomputadoras incluyen dBase, R:base y Paradox. Access de Microsoft, que utiliza el modelo relacional, es ahora el sistema de gestión de bases de datos basado en microcomputadora más ampliamente utilizado.

Actualmente, los sistemas de gestión de bases de datos empresariales más populares que utilizan el modelo relacional son Oracle, DB2, Informix, Sybase y SQL Server de Microsoft; el modelo relacional es un enfoque para la representación y gestión de datos en bases de datos, en este modelo, los datos se organizan en tablas con filas y columnas, y las relaciones entre las

tablas se establecen mediante claves foráneas este enfoque permite a los desarrolladores trabajar de manera eficiente con grandes cantidades de información y realizar consultas complejas para extraer datos relevantes.

Metodologías

Según Maida y Pacienza (2015) “Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo.”

Maida y Pacienza (2015) señalan que el proceso para crear software es un enfoque organizado y controlado para llevar a cabo un proyecto con una alta probabilidad de éxito; también indican que la metodología para el desarrollo de software incluye los pasos necesarios para concebir, aplicar y mantener un producto de software desde la identificación de la necesidad hasta el cumplimiento de su propósito original. Además, resaltan que, en el contexto de la ingeniería de software, es posible destacar que una metodología:

- Optimiza el proceso y el producto software.
- Métodos que guían en la planificación y en el desarrollo del software.
- Define qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto.

Metodología Ágil

Rivas, Corona, Gutiérrez y Hernández (2015) señalan que las empresas actualmente operan en un entorno global en constante cambio, y deben ser flexibles para adaptarse a nuevas oportunidades de mercado, cambios en las condiciones económicas y el surgimiento de productos y servicios nuevos y competitivos; para lograr esto, es esencial utilizar computadoras y tecnologías computacionales, lo que hace que el software sea un componente clave en casi todas las actividades empresariales. Por lo tanto, el desarrollo de software debe ser ágil para responder de manera oportuna y de calidad a todas las necesidades.

Orjuela y Rojas (2008) mencionan que las metodologías ágiles surgieron en los 90 como una respuesta a las crecientes expectativas y necesidades de los usuarios, producto de la masificación del uso de computadoras personales. Estas metodologías buscan lograr resultados más rápidos en el desarrollo de software sin comprometer su calidad, y han sido efectivas en resolver los problemas que surgen en este contexto.

La metodología Ágil es una forma efectiva y flexible de abordar el desarrollo de software que se enfoca en la entrega continua de software de alta calidad. Con su enfoque en la iteración y la adaptabilidad, la metodología Ágil es una elección popular para muchos equipos de desarrollo que buscan mejorar la eficiencia y la calidad de sus proyectos de software.

Metodología SCRUM

Según Maida y Pacienza (2015) los principios básicos de la metodología de trabajo de Scrum surgieron en la década de 1980, cuando fue desarrollada para abordar los procesos de reingeniería por parte de Goldratt, Takeuchi y Nonaka; la idea detrás de Scrum proviene de los nuevos métodos de producción utilizados en productos exitosos tanto en Japón como en los Estados Unidos; los equipos que crearon estos productos partían de requisitos amplios e innovadores, y tenían que lanzarlos al mercado en un plazo mucho más corto que antes. Estos equipos seguían patrones similares en la ejecución de sus proyectos. Se estudió la forma de trabajo de estos equipos altamente productivos y multidisciplinarios y se comparó con la colaboración entre los jugadores de rugby y su formación en un Scrum, del cual tomó su nombre.

Scrum vs Otras Metodologías. Según Rodríguez y Rubén Dorado (2015) Scrum, que es una metodología ágil, sigue muchos de los principios de estas, como la entrega temprana de productos funcionales, la retroalimentación constante en el equipo, la capacidad de adaptarse a los cambios y trabajar en conjunto con el cliente. También mencionan algunas características distintivas que son importantes resaltar.

- El tiempo; Scrum, que es una metodología ágil, se diferencia de otras metodologías como eXtreme Programming (XP) en cuanto a su marco de tiempo de iteración. Mientras que en XP se utilizan marcos de tiempo de una o dos semanas, en Scrum se trabaja con marcos de dos a cuatro semanas.
- El proceso; Scrum, como metodología, no dicta qué herramientas deben ser utilizadas en la aplicación de su proceso. La decisión sobre qué herramientas usar la toma la organización o el Scrum Master. Sin embargo, otras metodologías de desarrollo ágil si definen las herramientas como parte integral de su proceso estándar, como sucede con el desarrollo orientado a pruebas en XP.
- Los cambios; en la iteración Scrum, se recomienda mantener los compromisos adquiridos sin hacer cambios, mientras que otras metodologías de desarrollo ágil son más flexibles y permiten realizar ajustes durante el proceso de desarrollo de la iteración.
- El orden; en Scrum, el equipo está permitido para basarse en la prioridad establecida por el cliente y decidir en qué comprometerse para cada iteración en términos de desarrollo. En cambio, en XP, el equipo debe seguir las directrices proporcionadas por el cliente.
- La retroalimentación; Scrum recomienda brindar retroalimentación al final de cada sprint, mientras que, en otras metodologías de desarrollo ágil como XP, se sugiere brindar retroalimentación temprana a medida que se realiza la entrega.

Scrum cambió la forma en que el cliente percibía el trabajo, ya que, al ser parte del equipo, comprendió el trabajo que se estaba desarrollando, participó en el proceso y realizó revisiones tempranas, lo que resultó en mejoras en las entregas que eran de alta calidad y satisfacían sus necesidades.

Fases de Scrum. Arías Becerra y Durango Venegas (2018) Una de las ventajas clave de la metodología de desarrollo ágil Scrum es la capacidad de ser ágil y flexible en la interacción entre los distintos actores involucrados en el proceso de desarrollo. Además, Scrum

permite una rápida reacción a los cambios en el Product Backlog. Estos intercambios se dividen en fases, y es en estas donde se implementan los cambios significativos propuestos en este enfoque.

Mariño y Alfonso (2014) muestran las fases de SCRUM aplicadas a un proyecto de software:

- Gestión de requerimientos; consiste en una lista de tareas que lleva a la creación de un producto tecnológico, conocido como Product Backlog.
- Product Backlog; el Product Backlog es una lista de características, requisitos y tareas pendientes que deben ser desarrollados o mejorados en un producto o proyecto. Es una herramienta central en el desarrollo ágil, especialmente en el marco de trabajo Scrum. La lista de características y requisitos se prioriza y se revisa continuamente en función de las necesidades y objetivos del producto, lo que permite a los equipos de desarrollo enfocarse en lo más importante. La lista se mantiene viva y se adapta a los cambios en el mercado y los requisitos del usuario.
- Planificación del Sprint; durante esta fase, se asignan y secuencian las actividades que cada uno de los miembros del equipo realizará durante el Sprint, con el objetivo principal de crear un plan para almacenar los objetivos planteados para el Sprint, de acuerdo con el Product Backlog.
- Sprint; el Sprint es un período de tiempo limitado en el que se desarrolla el trabajo. El equipo de desarrollo trabaja en las tareas asignadas en la planificación del Sprint para lograr los objetivos específicos establecidos.
- Revisión del Sprint; esta fase es una oportunidad para que el equipo de desarrollo presente los resultados del Sprint a sus colegas y al cliente. La revisión del Sprint es una oportunidad para obtener retroalimentación y para mejorar el proceso en el futuro.

- Retrospectiva del Sprint; durante esta fase, el equipo de desarrollo reflexiona sobre el Sprint anterior y discute cómo pueden mejorar en el futuro. La retrospectiva es una oportunidad para identificar problemas y soluciones y para hacer ajustes en el proceso.

Estas son las fases básicas de la metodología Scrum. Sin embargo, es importante tener en cuenta que la metodología Scrum es un proceso iterativo, por lo que cada fase se repite constantemente durante el ciclo de vida del proyecto.

Capítulo III

Desarrollo del Tema

Diseño e Implementación de la Aplicación

Estructuración del Product Backlog

Un Product Backlog es una lista ordenada de características, funcionalidades o tareas necesarias para completar un proyecto de software. Para el desarrollo del Product Backlog de la aplicación, se utilizó una plantilla que se describe a continuación.

Tabla 1

Plantilla para el registro de tareas a ejecutarse en el Product Backlog.

Tarea	
ID:	Estimación:
Nombre:	Prioridad:
Responsable:	Estado:
Dependencias:	
Descripción:	

Nota. Estructura básica de un Product Backlog.

- ID: Se escribe un número o código único para identificar cada elemento del backlog.
- Nombre: Se escribe título corto y descriptivo que indica de qué se trata el elemento.
- Prioridad: Es un indicador de la importancia o urgencia del elemento en relación con los demás elementos del backlog.
- Estimación: Es una estimación del tiempo o esfuerzo requerido para completar el elemento.
- Descripción: Se escribe una descripción detallada del elemento, incluyendo requisitos, restricciones y detalles técnicos.
- Responsable: Se escribe el nombre del miembro del equipo que es responsable de completar el elemento.
- Dependencias: Es una lista de otros elementos del backlog que deben ser completados antes de poder trabajar en este elemento.

- Estado: Es un indicador del progreso del elemento, como "pendiente", "en progreso" o "completado".

Reunión y Planificación del Product Backlog.

En las reuniones de trabajo con el cliente, conforme se iba actualizando los requerimientos para el desarrollo de la aplicación web, el cliente explico que necesitaba una aplicación web que le ayude a gestionar las ordenes de su negocio, de igual forma necesitaba que todos sus registros de ventas se registraran automáticamente en una base de datos y por último requería que todo sea implementado en un servidor local (localhost).

Ya entendido el sistema de negocio al cual iba ser implementada la aplicación, junto con el cliente se procedió a discutir las fases del desarrollo de la aplicación, también se le explico al cliente sobre la metodología de la cual se estaría haciendo uso para el desarrollo e implementación de la aplicación en este caso la metodología Scrum Lite, ya que la comunicación entre usuario final y el desarrollador debe ser constante, para garantizar un producto de calidad y actualizaciones continuas.

A continuación, se detallan las tareas que se registraron en el Product Backlog durante las reuniones de trabajo, mediante la metodología Scrum Lite las tareas del Product Backlog se actualizaron constantemente para conseguir un producto final amigable con el usuario e intuitivo para todo el personal.

Sprints del Product Backlog

Tabla 2

Sprint N. °1 Diseño y desarrollo página de inicio.

		Tarea	
ID:	1	Estimación:	3 días
Nombre:	Diseño y desarrollo de interfaz de usuario para la página de inicio	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	Ninguna		

Tarea

- Descripción:**
- Color de fondo #AD2653.
 - Los botones deben ocupar el 75% del ancho de la página en dispositivos móviles, en computadoras de escritorio deben ocupar el 50% y deben tener bordes redondeados de color #000000.
 - El color de fondo de los botones debe ser #AD2653 y el texto #000000.
 - La página principal debe contar con 5 botones con las siguientes opciones:
 1. Nueva Orden icono:
(<i className="bi bi-journal-plus"></i>).
 2. Ordenes Pendientes icono:
(<i className="bi bi-journal-medical"></i>).
 3. Ordenes por Cobrar icono:
(<i className="bi bi-cash-coin"></i>).
 4. Libro de Ordenes icono:
(<i className="bi bi-journal-text"></i>).
 5. Nuevo Producto icono:
(<i className="bi bi-journal-medical"></i>).
 - Al posar el puntero del mouse sobre un botón, cambiar el color de fondo de los botones por #000000 y el texto por #ffffff.
-

Nota. Diseños y funcionalidades de la página de inicio.

Tabla 3

Sprint N. °2 Diseño y desarrollo página nueva orden.

Tarea			
ID:	2	Estimación:	15 días
Nombre:	Diseño y desarrollo de interfaz de usuario para la página de Nueva Orden	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Color de fondo #AD2653. ● Botón para regresar al menú principal en la esquina superior derecha con icono (<i className="bi bi-house-fill"></i>). ● Botón Home con fondo #000000 y texto #ffffff. ● La página se divide en tres secciones en forma vertical: <ol style="list-style-type: none"> 1. La primera sección debe abarcar el 25% del alto de toda la pantalla y está designada para la tabla donde se van a ir registrando las ordenes de los clientes. 2. La segunda sección también debe abarcar el 25% del lato de toda la pantalla y esta sección está designada para las categorías de los productos. 3. Y la tercera sección debe abarcar el 50% del alto de toda la pantalla y en esta sección es donde se van a mostrar todos los productos que ofrece el negocio. 		
Descripción:			

Nota. Diseño y estructuración de la página Nueva Orden.

Tabla 4

Sprint N. °3 Diseño y desarrollo tabla registro de órdenes.

Tarea			
ID:	3	Estimación:	8 días
Nombre:	Diseño y desarrollo de tabla para registrar las órdenes.	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Página Nueva Orden 		
Descripción:	<ul style="list-style-type: none"> ● Cabeceras de la tabla: <ul style="list-style-type: none"> ❖ CANT. ❖ Producto ● Color de fondo de la tabla #000000 y texto #ffffff. ● La tabla debe estar en un container con la clase "overflow-auto", con el fin de que exista un scroll en la tabla. ● La tabla se divide en 3 columnas: <ol style="list-style-type: none"> 1. Cantidad (CANT.). 2. Nombre del Producto (Producto). 3. Botón Sabores (Sabores). ● En la columna CANT. se debe visualiza por medio de una etiqueta (label) la cantidad total del producto elegido, esta se debe acompañar con otra etiqueta menos (-) que sirva para restar en -1 la cantidad total. ● En la columna Producto se debe visualizar el nombre del producto escogido. ● En la última columna debe existir un botón Sabores con la clase "btn btn-primary btn-sm", este botón deberá dirigir hacia el apartado sabores. 		

Nota. Diseño, desarrollo y estructuración de la tabla de registro de pedidos.

Tabla 5

Sprint N. °4 Diseño y desarrollo categorías.

Tarea			
ID:	4	Estimación:	3 días
Nombre:	Diseño y desarrollo de las categorías de los productos	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Página Nueva Orden ● Color de fondo #AD2653. 		
Descripción:	<ul style="list-style-type: none"> ● Botones centrados, en un orden de 3 x 2, se debe utilizar la clase "btn btn-success btn-sm". ● La sección cuenta con 6 botones que se dividen en las siguientes categorías: 		

Tarea	
	<ol style="list-style-type: none"> 1. Todo 2. Menú 3. Bebidas 4. Snacks 5. Porciones 6. Otros <ul style="list-style-type: none"> ● Al momento de presionar en cualquiera de las categorías, se filtrarán todos los productos pertenecientes a esa categoría.

Nota. Diseño, estructuración y desarrollo de la sección de las categorías de los productos.

Tabla 6

Sprint N. °5 Diseño y desarrollo sección productos.

Tarea			
ID:	5	Estimación:	4 días
Nombre:	Diseño y desarrollo de la sección de productos	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Página Nueva Orden ● Color de fondo #AD2653. ● El nombre de cada producto se debe visualizar mediante tarjetas. ● El ancho de la tarjeta debe ser el 22% del container, y el alto debe ser de 4em. 		
Descripción:	<ul style="list-style-type: none"> ● El Color de fondo de la tarjeta debe ser #000000 y el texto #ffffff ● El orden de la sección de productos debe ser de 4 columnas. ● Toda a sección debe estar contenida en un container con scroll. ● Al momento de seleccionar un producto, automáticamente este se añade a la tabla de pedidos. ● Si se vuelve a seleccionar un producto la cantidad del producto en la tabla de pedidos se incrementa +1. 		

Nota. Diseño, desarrollo y estructuración de la sección productos.

Tabla 7

Sprint N. °6 Diseño y desarrollo sección escoger sabor.

Tarea			
ID:	6	Estimación:	4 días
Nombre:	Diseño y desarrollo de la sección escoger sabores.	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Página Nueva Orden ● Tabla de registro de pedidos 		

Tarea	
Descripción:	<ul style="list-style-type: none"> ● Para el color de fondo utilizar la clase “bg-dark”. ● En la parte superior, deben existir 2 botones: <ol style="list-style-type: none"> 1. Botón Guardar: Debe ser instanciado con la clase “btn btn-success”, es el encargado de guardar los sabores escogidos. 2. Botón Cancelar: Debe ser instanciado con la clase “btn btn-success”, este botón se encarga de cancelar y cerrar la sección de sabores. ● Mediante tarjetas se debe mostrar: <ul style="list-style-type: none"> ❖ Nombre del producto. ❖ Índice del producto. ❖ Un área de texto. ❖ Un botón con la clase “btn btn-dark” que diga “Sabor”. ❖ Un checkbox con una etiqueta que diga “Llevar”. ● En el área de texto debe existir un placeholder que muestre “Nota...”, este campo sirve para que el personal de servicio escriba alguna nota adicional al pedido. ● Al momento de hacer clic en el botón “Sabor”, se debe desplegar la sección sabores.

Nota. Requerimientos para la sección escoger sabores.

Tabla 8

Sprint N. 07 Diseño y desarrollo sección sabores.

Tarea			
ID:	7	Estimación:	4 días
Nombre:	Diseño y desarrollo de la sección sabores	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Página Nueva Orden ● Tabla de registro de pedidos ● Sección escoger sabores ● Para el color de fondo utilizar la clase “bg-dark”. ● En la parte superior, debe existir un encabezado H1 con el título “Sabores”. ● Para cada sabor se debe utilizar un botón. ● Cada botón de los sabores, debe tener un color de fondo, color que debe ser acordado con el cliente. 		
Descripción:	<ul style="list-style-type: none"> ● Al momento de hacer clic en un sabor, este debe registrarse en un arreglo, si el producto posee más de un sabor, se debe poder seguir escogiendo los demás sabores. ● Una vez escogidos todos los sabores, automáticamente se cierra la sección sabores y se debe visualizar la sección escoger sabores. 		

Nota. Requerimientos para la sección sabores

Tabla 9

Sprint N. °8 Diseño y desarrollo carrito de compras.

Tarea			
ID:	8	Estimación:	5 días
Nombre:	Diseño y desarrollo del carrito de compras.	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Página Nueva Orden ● Tabla de registro de pedidos 		
Descripción:	<ul style="list-style-type: none"> ● Para el carrito de compras se debe instanciar un botón con la clase "btn btn-success". ● El botón debe estar posicionado en la esquina superior derecha a un lado del botón home de la página de nueva orden. ● El botón debe tener un icono de carrito de supermercado. ● El botón del carrito se debe visualizar solo si existe algún producto en la tabla de registro de pedidos. ● Al momento de hacer clic en el botón carrito de compras se debe visualizar la sección factura. 		

Nota. Especificaciones para el desarrollo de la sección carrito de compras.

Tabla 10

Sprint N. °9 Diseño y desarrollo sección detalle venta.

Tarea			
ID:	9	Estimación:	4 días
Nombre:	Diseño y desarrollo de la sección detalle venta.	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Página Nueva Orden ● Tabla de registro de pedidos ● Botón carrito de compras ● Para el color de fondo utilizar la clase "bg-light". ● En la parte superior, debe existir un encabezado H1 donde se muestre "Orden N. °" y el número de orden a la que corresponde la factura. 		
Descripción:	<ul style="list-style-type: none"> ● Para separar el encabezado con el detalle de la factura, debe existir una línea. ● Para detallar los productos de la orden, se debe utilizar etiquetas. ● En el detalle de venta se debe detallar el producto y el valor del 		

Tarea
<p>producto.</p> <ul style="list-style-type: none"> ● Para los productos que se marcaron para llevar se debe mostrar después del nombre del producto, un badge que diga llevar. ● Al final de la lista de todos los productos de la orden, se debe poder visualizar el total de la orden. ● En la parte posterior de la pantalla deben instanciarse 4 botones: <ol style="list-style-type: none"> 1. Orden Cobrada: Marca el estado de la orden como cobrada y regresa a la página de inicio, se debe instanciar con la clase "btn btn-success". 2. Orden por Cobrar: Marca el estado de la orden como pendiente por cobrar y regresa a la página de inicio, se debe instanciar con la clase "btn btn-light". 3. Regresar: Cierra la sección Factura y debe permitir seguir registrando más productos a la orden, se debe instanciar con la clase "btn btn-primary". 4. Cancelar: Cancela la orden y regresa a la página de inicio.

Nota. Estructura para el desarrollo de la sección factura.

Tabla 11

Sprint N. °10 Diseño y desarrollo página ordenes pendientes.

Tarea			
ID:	10	Estimación:	4 días
Nombre:	Diseño y desarrollo de la página Ordenes Pendientes	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Para el color de fondo utilizar "#AD2653". ● En la parte superior, deben existir un encabezado H1 en donde muestre "Ordenes Pendientes": ● En la parte inferior se debe instanciar un botón con la clase "btn btn-primary", el botón debe decir "Regresar". ● Al hacer clic en el botón "Regresar", debe regresar a la página de inicio. ● En el centro de la página se debe instanciar un contenedor con un tamaño del 75% y con la clase "overflow-auto". ● En el contenedor mediante tarjetas Se deben mostrar las ordenes que están pendientes por entregar. ● Cada tarjeta debe mostrar: <ul style="list-style-type: none"> ❖ Numero de Orden. ❖ Nombre del cliente. ❖ Cantidad del producto. ❖ Nombre del producto. ❖ Sabores del producto. ❖ Botón "Entregado". 		

Nota. Requisitos para el desarrollo de la página Ordenes Pendientes.

Tabla 12

Sprint N. °11 Diseño y desarrollo página ordenes por cobrar.

Tarea			
ID:	11	Estimación:	4 días
Nombre:	Diseño y desarrollo de la página Ordenes Por Cobrar	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Para el color de fondo utilizar "#AD2653". ● En la parte superior, deben existir un encabezado H1 en donde muestre "Ordenes Por Cobrar": ● En la parte inferior se debe instanciar un botón con la clase "btn btn-primary", el botón debe decir "Regresar". ● Al hacer clic en el botón "Regresar", debe regresar a la página de inicio. ● En el centro de la página se debe instanciar un contenedor con un tamaño del 75% y con la clase "overflow-auto". ● En el contenedor mediante tarjetas Se deben mostrar las ordenes que están pendientes por cobrar. ● Cada tarjeta debe mostrar: <ul style="list-style-type: none"> ❖ Numero de Orden. ❖ Nombre del cliente. ❖ Cantidad del producto. ❖ Nombre del producto. ❖ Sabores del producto. ❖ Total, de la orden. ❖ Botón "Cobrado". ● Al hacer clic en el botón "Cobrado", automáticamente se debe eliminar la orden de la página Ordenes Por Cobrar. 		

Nota. Requerimientos para el desarrollo de la página Ordenes por Cobrar.

Tabla 13

Sprint N. °12 Diseño y desarrollo página libro de órdenes.

Tarea			
ID:	12	Estimación:	4 días
Nombre:	Diseño y desarrollo de la página Libro de Ordenes	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Para el color de fondo utilizar "#AD2653". ● En la parte superior, deben existir un encabezado H1 en donde muestre "Libro de Ordenes": ● En la parte inferior se debe instanciar un botón con la clase "btn 		

Tarea
<ul style="list-style-type: none"> ● btn-primary”, el botón debe decir “Regresar”. ● Al hacer clic en el botón “Regresar”, debe regresar a la página de inicio. ● En el centro de la página se debe instanciar un contenedor con un tamaño del 75% y con la clase “overflow-auto”. ● En el contenedor mediante tarjetas Se deben mostrar el reporte de ventas diario. ● Cada tarjeta debe mostrar: <ul style="list-style-type: none"> ❖ Fecha en formato dd/mm/aa. ❖ Total, de ventas de esa fecha. ● La tarjeta debe estar instanciada con la clase “bg-dark”. ● Para la fecha utilizar la clase “text.warning”. ● El total se debe mostrar en etiqueta (label) y debe estar instanciada con la clase “text-light”. ● Debe registrarse todas la ordenes en el apartado libro de ordenes

Nota. Requerimientos para el desarrollo de la página Libro de Ordenes.

Tabla 14

Sprint N. °13 Diseño y desarrollo sección detalle de fecha.

Tarea			
ID:	13	Estimación:	4 días
Nombre:	Diseño y desarrollo de la sección detalle de fecha	Prioridad:	Alta
Responsable:	Uchuari Kevin	Estado:	Completado
Dependencias:	<ul style="list-style-type: none"> ● Página de inicio ● Página Libro de Ordenes 		
Descripción:	<ul style="list-style-type: none"> ● Para el color de fondo utilizar “#AD2653”. ● En la parte superior, deben existir un encabezado H1 en donde muestre “Ordenes de:” y la fecha del registro. ● En la parte inferior se debe instanciar un botón con la clase “btn btn-primary”, el botón debe decir “Regresar” y al lado derecho debe existir un recuadro donde se pueda visualizar los ingresos totales de esa fecha. ● Al hacer clic en el botón “Regresar”, debe regresar a la página Libro de Ordenes. ● En el centro de la página se debe instanciar un contenedor con un tamaño del 75% y con la clase “overflow-auto”. ● En el contenedor mediante tarjetas Se deben mostrar las todas las ordenes registradas en esa fecha. ● Cada tarjeta debe mostrar: <ul style="list-style-type: none"> ❖ Fecha en formato dd/mm/aa. ❖ Total de ventas de esa fecha. ● La tarjeta debe estar instanciada con la clase “bg-dark”. ● Para la fecha utilizar la clase “text.warning”. ● El total se debe mostrar en etiqueta (label) y debe estar 		

Tarea

- instanciada con la clase “text-light”.
 - Al momento de hacer clic en cualquier tarjeta, se debe abrir la sección detalle de fecha, en donde se deben mostrar todas las órdenes registradas en esa fecha.
 - Cada tarjeta debe mostrar:
 - ❖ Numero de Orden.
 - ❖ Nombre del cliente.
 - ❖ Cantidad del producto.
 - ❖ Nombre del producto.
 - ❖ Sabores del producto.
 - ❖ Total, de la orden.
-

Nota. Requisitos para el desarrollo de la sección detalle de fecha

Diseño de la Aplicación

Para la fase del diseño de la aplicación se utilizó los patrones de diseño proporcionados por la librería Bootstrap, con esta librería conseguimos que nuestra aplicación sea cien por ciento Responsive, gracias a esto logramos que nuestra aplicación se adapte a todas las pantallas.

La aplicación requirió ser responsive, ya que los empleados utilizan el teléfono móvil para realizar el registro de la ordenes de los clientes.

De igual forma el equipo encargado de cocina utiliza los teléfonos móviles para revisar si tienen ordenes pendientes de entregar, lo que hace que el servicio al cliente sea más rápido y eficaz.

La aplicación que el cliente requería, tenía que ser adaptable a dispositivos móviles, de esta forma se ahorrarían costes de implementación, ya que todo el personal cuenta con un teléfono inteligente y la aplicación estaría al alcance de todos, sin necesidad de adquirir equipos nuevos.

De esta forma logramos que la aplicación se cien por ciento adaptable a cualquier dispositivo, ya sea de escritorio o móvil.

Interfaces de Usuario

Figura 1

Página de inicio.



Nota. Interfaz de usuario de la página de inicio.

Figura 2

Página Nueva Orden.



Nota. Interfaz de usuario de la página Nueva Orden.

Figura 3

Tabla de registro de pedidos y botón carrito de compras.



Nota. La tabla de registros de pedidos y el botón carrito de compras solo se visualiza cuando se escoge algún producto.

Figura 4

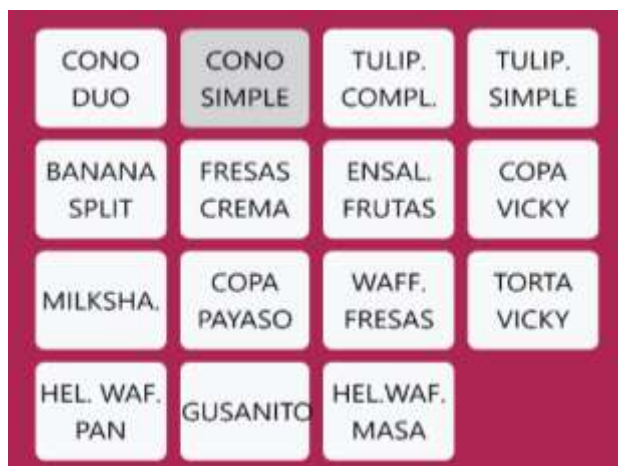
Categorías de los productos.



Nota. La imagen muestra las categorías de los productos con las que cuenta el negocio.

Figura 5

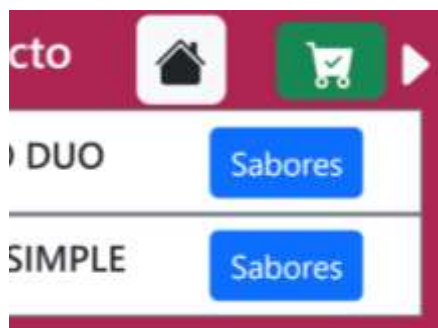
Catálogo de los productos.



Nota. Los productos se visualizan mediante tarjetas.

Figura 6

Botón Sabores



Nota. Es el encargado de abrir la sección sabores

Figura 7

Sección Sabores

A screenshot of the 'Sección Sabores' (Flavors Section) in a mobile application. At the top, there are two buttons: 'Guardar' (green) and 'Cancelar' (red). Below these are three product options, each in a separate colored box. The first box is light gray and labeled 'CONO DUO N°1:'. The second box is light orange and labeled 'CONO DUO N°2:'. The third box is light gray and labeled 'CONO DUO N°3:'. Each box contains a 'Sabores' button, a text input field labeled 'Nota...', and a checkbox labeled 'Llevar:'.

Nota. En la sección sabores se puede escoger los sabores para cada producto.

Figura 8

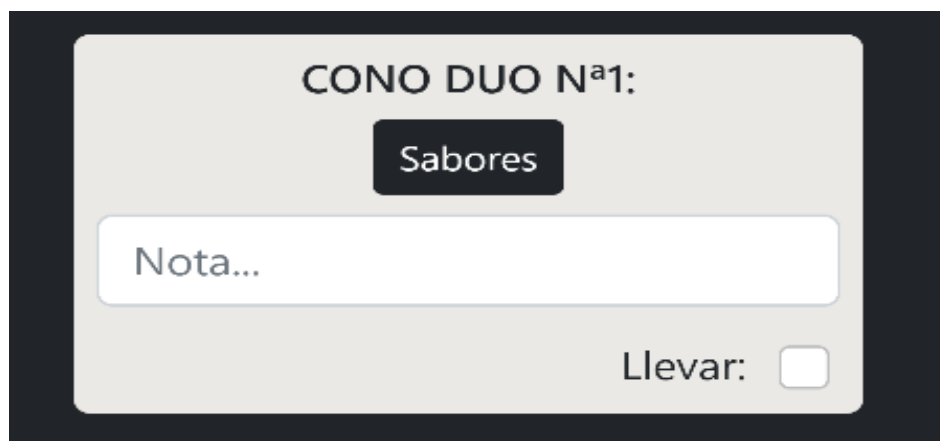
Botón guardar y botón cancelar



Nota. El botón guardar, salva los sabores escogidos y el botón cancelar, cierra la sección sabores.

Figura 9

Producto escogido.



Nota. La imagen muestra la tarjeta en la cual se debe mostrar el producto escogido.

Figura 10

Botón Sabores del producto.



Nota. El botón sabores es el encargado de abrir la sección Escoger Sabores.

Figura 11


Campo de texto.

A screenshot of a text input field with a light gray border and rounded corners. The text "Nota..." is displayed inside the field. The field is centered between two dark gray vertical bars.

Nota. En este campo el personal de servicio puede agregar un comentario al producto escogido.

Figura 12

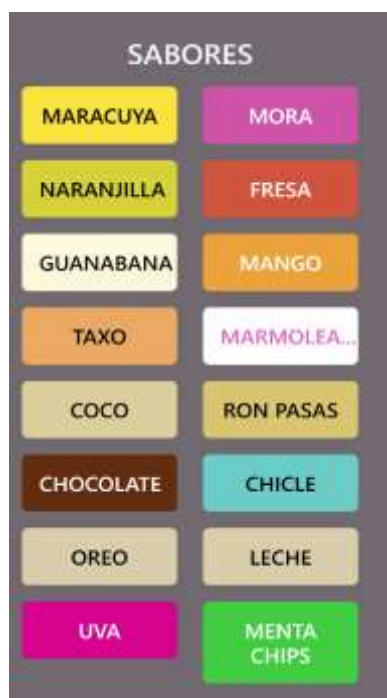
Checkbox Llevar.

A screenshot of a checkbox labeled "Llevar:". The checkbox is currently unchecked and is located on a light gray background with a dark gray shadow.

Nota. Este Checkbox ayuda al personal a saber si el producto es para llevar o para servir.

Figura 13

Sección Escoger Sabor.

A screenshot of a section titled "SABORES" (Flavors). It displays a grid of 16 buttons, each representing a different flavor. The buttons are arranged in two columns and eight rows. The flavors listed are: MARACUYA, MORA, NARANJILLA, FRESA, GUANABANA, MANGO, TAXO, MARMOLEA..., COCO, RON PASAS, CHOCOLATE, CHICLE, OREO, LECHE, UVA, and MENTA CHIPS. Each button has a unique background color.

Nota. En la sección Escoger Sabores los sabores se muestran mediante botones con diferentes colores de fondo.

Figura 14

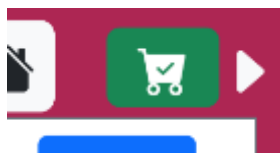
Sabores escogidos.



Nota. Los sabores escogidos en la sección Escoger Sabores, se muestran en vez del botón sabores.

Figura 15

Botón carrito de compras.



Nota. El botón carrito de compras abre la sección factura.

Figura 16

Sección Factura.



Nota. En la sección factura se muestran detallados los productos, sus valores y una serie de opciones para el control y gestión de la orden.

Figura 17

Página Ordenes Pendientes.



Nota. La imagen muestra la página Ordenes Pendientes, la cual se encarga de ayudar a todo el personal a saber que pedidos están pendientes de entregar.

Figura 18

Sección Ordenes Pendientes.



Nota. En esta sección se detallan en número de orden, el nombre del cliente, los productos de la orden, los sabores y comentarios.

Figura 19

Página Ordenes por Cobrar.



Nota. La imagen muestra la página Ordenes por Cobrar, la cual se encarga de ayudar a todo el personal a saber que ordenes están pendientes por cobrar.

Figura 20

Sección Ordenes por Cobrar.



Nota. En esta sección al presionar el botón Orden Cobrada automáticamente la orden se elimina de la sección ordenes por cobrar.

Figura 21

Página Libro de Ordenes.



Nota. La imagen muestra la página libro de ordenes en donde se detallan los ingresos diarios que ha obtenido el negocio.

Figura 22

Sección Detalle de Fecha.



Nota. En esta sección se puede visualizar todas las órdenes registradas en una fecha en específico, también se muestra los totales de cada orden y el total de ingresos obtenidos en esa fecha.

Implementación de la aplicación

Para la implementación de la aplicación se utilizó tecnologías tales como un servidor Node, con el framework de desarrollo basado en React, Next Js; también se utilizó un gestor de base de datos, en este caso MySQL.

Instalación y configuración del servidor local

Para la instalación del servidor Node se utilizó un equipo COMPAQ, con el sistema operativo Windows 8.1, este equipo sería el encargado de albergar la aplicación web y la base de datos, así que se procedió a realizar la configuración de los servicios que debían ejecutarse.

Figura 23

Equipo para la instalación del servidor local.



Nota. La imagen muestra el equipo para la instalación del servidor local.

Figura 24

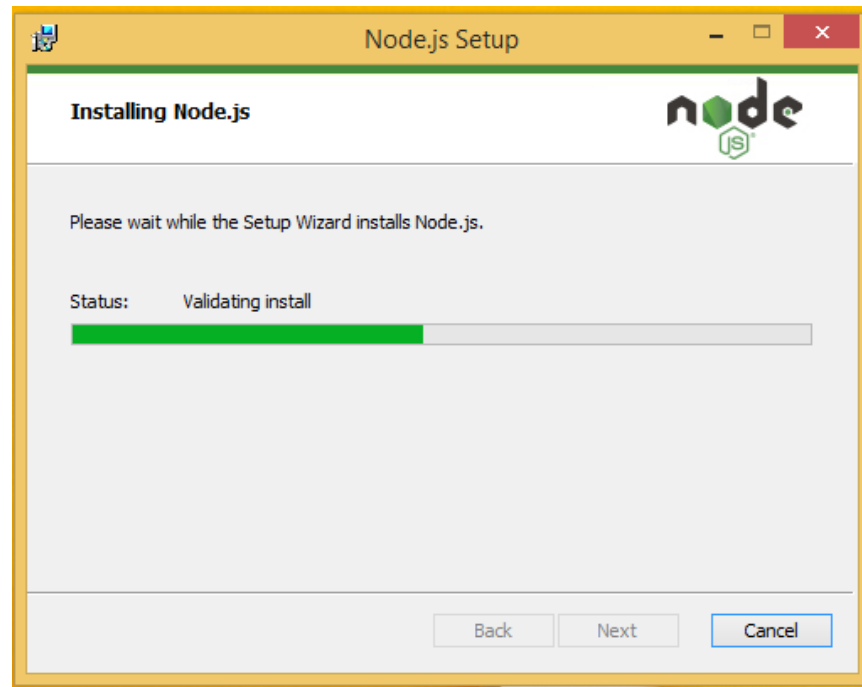
Especificaciones técnicas del equipo.

Edición de Windows	
Windows 8.1 Pro N	
© 2013 Microsoft Corporation. Todos los derechos reservados.	
	
Sistema	
Procesador:	AMD E-300 APU with Radeon(tm) HD Graphics 1.30 GHz
Memoria instalada (RAM):	8,00 GB (3,49 GB utilizable)
Tipo de sistema:	Sistema operativo de 32 bits, procesador x64
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponible para esta pantalla

Nota. En la imagen se puede visualizar las especificaciones del computador.

Figura 25

Instalación del servidor Node Js en el equipo.



Nota. La imagen muestra la instalación de Node Js.

Figura 26

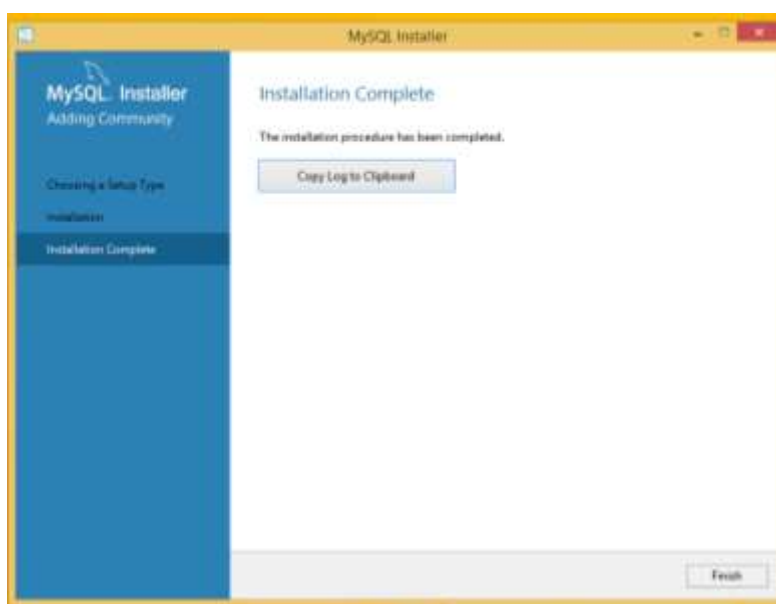
Verificación de la versión de Node instalada en el equipo.



Nota. Versión de Node instalada.

Figura 27

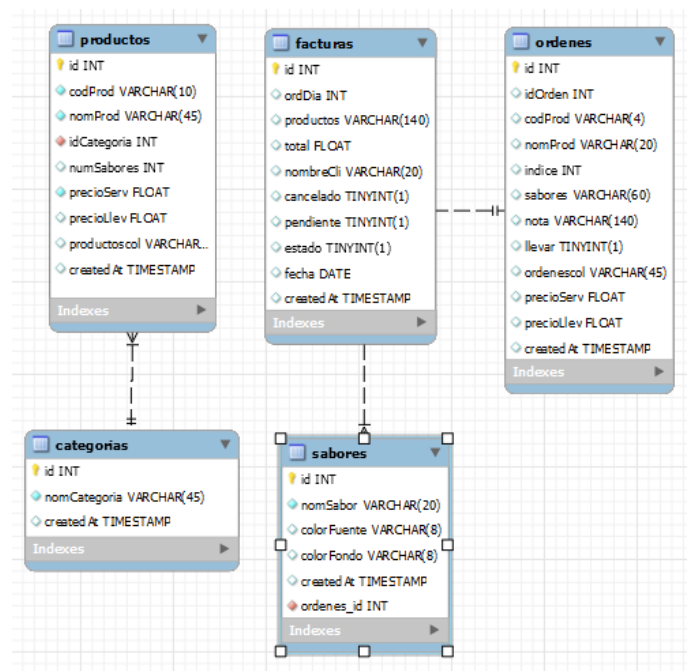
Instalación del gestor de base de datos.



Nota. La imagen muestra la instalación de MySQL en el equipo.

Figura 28

Modelo Relacional de la base de datos.



Nota. La figura muestra el modelo relacional de la base de datos para la aplicación.

Figura 29

Script de la tabla productos.

```

1  CREATE TABLE `heladeria`.`productos` (
2  `id` INT NOT NULL AUTO_INCREMENT,
3  `codProd` VARCHAR(10) NOT NULL,
4  `nomProd` VARCHAR(45) NOT NULL,
5  `idCategoria` INT NOT NULL,
6  `numSabores` INT NULL,
7  `precioServ` FLOAT NOT NULL,
8  `precioLev` FLOAT NULL,
9  `createdAt` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
10 PRIMARY KEY (`id`));
11

```

Nota. En la imagen se puede visualizar la creación de la tabla productos.

Figura 30

Script de la tabla categorías.

```

1  CREATE TABLE `heladeria`.`categorias` (
2  `id` INT NOT NULL AUTO_INCREMENT,
3  `nomCategoria` VARCHAR(15) NOT NULL,
4  `createdAt` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
5  PRIMARY KEY (`id`));
6

```

Nota. La imagen muestra la creación de la tabla categorías.

Figura 31

Script de la tabla sabores.

```

1  CREATE TABLE `heladeria`.`sabores` (
2  `id` INT NOT NULL AUTO_INCREMENT,
3  `nomSabor` VARCHAR(20) NULL,
4  `colorFuente` VARCHAR(8) NULL,
5  `colorFondo` VARCHAR(3) NULL,
6  `createdAt` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
7  PRIMARY KEY (`id`));
8

```

Nota. En la imagen se puede visualizar la creación de la tabla sabores.

Figura 35

Pool de conexión a la base de datos.

```
import { createPool } from 'mysql2/promise';

const pool = createPool({
  host: '127.0.0.1',
  user: 'root',
  password: '123456',
  port: 3306,
  database: 'heladeria'
})
export {pool};
```

Nota. En la imagen se muestra la conexión a la base de datos.

Figura 36

Código para la obtención de productos.

```
import { pool } from "../../database/db";
export default async function handler(req, res ){
  switch(req.method) {
    case 'GET':
      return await obtenerProductos(req, res);
    case 'POST':
      return await guardarProducto(req, res);
  }
}
const obtenerProductos = async (req, res) =>{
  const [result] = await pool.query('SELECT * FROM productos');
  console.log(result);
  return res.status(200).json(result);
}
const guardarProducto = async (req, res) =>{
  const {codProd, nomProd, idCategoria, numSabores, precioServ, precioLlev} =
req.body;
  const [result] = await pool.query('INSERT INTO productos SET ?',{
    codProd,
    nomProd,
    idCategoria,
    numSabores,
    precioServ,
    precioLlev
  })
  return res.status(204).json();
}
```

Nota. La imagen muestra el código para obtener los productos.

Figura 37

Código para filtrar los productos.

```
import { pool } from "../../database/db";
export default async function handler(req, res ){
  switch(req.method) {
    case 'POST':
      return await filtrarProducto(req, res);
  }
}
const filtrarProducto = async (req, res) =>{
  const { idCategoria } = req.body;
  const [result] = await pool.query('SELECT * FROM productos WHERE (idCategoria = ?)',[idCategoria])
  return res.status(200).json(result);
}
```

Nota. En la imagen se puede visualizar el código para filtrar los productos.

Figura 38

Código para registrar pedidos.

```
import { pool } from "../../database/db";
export default async function handler(req, res ){
  switch(req.method) {
    case 'GET':
      return await obtenerProductos(req, res);
    case 'POST':
      return await guardarProducto(req, res);
  }
}
const obtenerProductos = async (req, res) =>{
  const [result] = await pool.query('SELECT * FROM ordenes');
  console.log(result);
  return res.status(200).json(result);
}
const guardarProducto = async (req, res) =>{
  const {idOrden, codProd, nomProd, indice, sabores, nota, llevar, precioServ, precioLlev} = req.body;
  const [result] = await pool.query('INSERT INTO ordenes SET ?',{
    idOrden,
    codProd,
```

Nota. La imagen muestra el código para el registro de los pedidos de los clientes.

Figura 39

Código para registrar las facturas.

```
import { pool } from "../../../database/db";
export default async function handler(req, res ){
  switch(req.method) {
    case 'GET':
      return await obtenerFactura(req, res);
    case 'POST':
      return await guardarFactura(req, res);
  }
}
const obtenerFactura = async (req, res) =>{
  const [result] = await pool.query('SELECT * FROM facturas');
  console.log(result);
  return res.status(200).json(result);
}
const guardarFactura = async (req, res) =>{
  const {ordDia, productos, total, nombreCli, cancelado, pendiente, estado} =
req.body;
  const [result] = await pool.query('INSERT INTO facturas SET ? , fecha =
CURDATE()',{
    ordDia,
    productos,
    total,
    nombreCli,
    cancelado,
    pendiente,
    estado,
  })
  return res.status(204).json();
}
```

Nota. En la imagen se puede visualizar el código para registrar facturas.

Figura 40

Código para obtener los sabores.

```
import { pool } from "../../database/db";
export default async function handler(req, res ){
  switch(req.method) {
    case 'GET':
      return await obtenerSabores(req, res);
    case 'POST':
      return await guardarSabores(req, res);
  }
}
const obtenerSabores = async (req, res) =>{
  const [result] = await pool.query('SELECT * FROM sabores');
  console.log(result);
  return res.status(200).json(result);
}
const guardarSabores = async (req, res) =>{
  const {nomSabor} = req.body;
  const [result] = await pool.query('INSERT INTO productos SET ?',{
    nomSabor,
  })
  return res.status(204).json();
}
```

Nota. En la imagen se muestra el código para obtener los sabores que existen.

Fase de pruebas

La fase de pruebas en el desarrollo de software se refiere a un proceso en el que se evalúa la calidad, la funcionalidad y la confiabilidad de una aplicación antes de su lanzamiento al público. El objetivo principal de la fase de pruebas es identificar y corregir cualquier error o problema en la aplicación para garantizar una experiencia de usuario satisfactoria. La fase de pruebas puede incluir una variedad de técnicas, incluyendo pruebas unitarias, pruebas de integración, pruebas de aceptación y pruebas de rendimiento, entre otros. La fase de pruebas es un paso crucial en el ciclo de desarrollo de software y se considera una parte integral de la garantía de calidad del software.

Desarrollo de Pruebas

Tabla 15

Prueba N. °1

Evaluación de prueba	
Nombre:	Ingreso a la aplicación
Objetivos de la prueba:	Visualizar el tiempo de respuesta de la aplicación.
Procedimiento de la prueba:	<ul style="list-style-type: none"> • Ingreso mediante un navegador web. • Visualización de la página de inicio.
Resultados esperados:	Al ingresar a la IP del servidor se debe mostrar la página de inicio con todas las opciones principales de la aplicación.
Estado:	Completado exitosamente
Observaciones:	Ninguna

Nota. La tabla muestra la evaluación de prueba n. °1.

Tabla 16

Prueba N. °2

Evaluación de prueba	
Nombre:	Crear nueva orden
Objetivos de la prueba:	Registrar una nueva orden y añadir productos.
Procedimiento de la prueba:	<ul style="list-style-type: none"> • Presionar botón nueva orden. • Visualización de la página de nueva orden. • Registro y creación de nuevo número de orden.
Resultados esperados:	Al presionar el botón Nueva Orden automáticamente se crea un nuevo número de orden, este número de orden sirve para diferenciar la orden de las demás.
Estado:	Completado exitosamente
Observaciones:	El nuevo número de orden se obtiene mediante las consultas a la base de datos de las ordenes creadas, este nuevo número de debe enviar mediante parámetros de URL.

Nota. La tabla muestra la evaluación de prueba n. °2.

Tabla 17

Prueba N. °3

Evaluación de prueba	
Nombre:	Agregar productos
Objetivos de la prueba:	Registrar productos a la nueva orden. <ul style="list-style-type: none"> • Presionar la categoría que busca.
Procedimiento de la prueba:	<ul style="list-style-type: none"> • Visualizar todos los productos pertenecientes a la categoría escogida. • Presionar el producto deseado. • Agregarlo a la tabla de productos.
Resultados esperados:	Al presionar el botón de cualquier categoría, automáticamente se filtran todos los productos pertenecientes a esa categoría, al presionar en alguno de estos productos, el sistema añade el producto a la tabla de pedidos.
Estado:	Completado exitosamente
Observaciones:	Para añadir 2 o más productos del mismo tipo se debe presionar el mismo producto así se aumenta la cantidad del mismo producto.

Nota. La tabla muestra la evaluación de prueba n. °3.

Tabla 18

Prueba N. °4

Evaluación de prueba	
Nombre:	Personalizar producto
Objetivos de la prueba:	<ul style="list-style-type: none"> • Registrar los sabores de los productos escogidos. • Ingresar una corta descripción. • Escoger producto para llevar. • Presionar botón sabores.
Procedimiento de la prueba:	<ul style="list-style-type: none"> • Visualización de la sección de Escoger Sabores. • Presionar botón Sabor de cada producto. • Visualización de la sección Sabores. • Presionar el botón del sabor que se desea.
Resultados esperados:	Al presionar el botón sabores, se deberá visualizar la sección Escoger Sabores, cada producto se debe mostrar en tarjetas en donde al dar clic en el botón sabor, se despliegue una lista con todos los sabores que existen, de igual forma en la tarjeta debe existir un campo de texto en donde se pueda ingresar una pequeña descripción del producto.
Estado:	Completado exitosamente
Observaciones:	Cada producto tiene diferente número de sabores.

Nota. La tabla muestra la evaluación de prueba n. °4.

Tabla 19

Prueba N. °5

Evaluación de prueba	
Nombre:	Crear factura
Objetivos de la prueba:	Registrar el total de la orden.
Procedimiento de la prueba:	<ul style="list-style-type: none"> • Presionar botón carrito de compras. • Visualización de la sección factura. • Calcular el total de la orden. • Visualización de los productos y sus totales.
Resultados esperados:	Al presionar el botón carrito de compras el sistema realiza el caculo del total de la orden y permite la visualización de los productos y sus valores.
Estado:	Completado exitosamente
Observaciones:	En este apartado se debe visualizar si el producto es para llevar.

Nota. La tabla muestra la evaluación de prueba n. °5.

Tabla 20

Prueba N. °6

Evaluación de prueba	
Nombre:	Visualizar Ordenes Pendientes.
Objetivos de la prueba:	Visualizar las ordenes que están pendientes por entregar.
Procedimiento de la prueba:	<ul style="list-style-type: none"> • Presionar botón Ordenes Pendientes. • Visualización de la página de Ordenes Pendientes. • Presionar botón Orden entregada. • Eliminar Orden de ordenes pendientes.
Resultados esperados:	Al presionar el botón Ordenes Pendientes, se debe visualizar la página Ordenes Pendientes en donde se muestran las ordenes que están pendientes de entregar, cada orden se muestra en una tarjeta donde se detallan los productos, sabores, descripción y si el producto es para llevar o para servirse, cada tarjeta debe tener un botón para confirmar si la orden ha sido entregada.
Estado:	Completado exitosamente
Observaciones:	Ninguna.

Nota. La tabla muestra la evaluación de prueba n. °6.

Tabla 21

Prueba N. 07

Evaluación de prueba	
Nombre:	Visualizar Ordenes por Cobrar.
Objetivos de la prueba:	Visualizar las ordenes que están pendientes por cobrar. <ul style="list-style-type: none"> • Presionar botón Ordenes por Cobrar.
Procedimiento de la prueba:	<ul style="list-style-type: none"> • Visualización de la página de Ordenes por Cobrar. • Presionar botón Orden cobrada. • Eliminar Orden de ordenes por cobrar. <p>Al presionar el botón Ordenes por Cobrar, se debe visualizar la página Ordenes por Cobrar en donde se muestran las ordenes que están pendientes por cobrar, cada orden se</p>
Resultados esperados:	muestra en una tarjeta donde se detallan los productos, sabores, descripción y si el producto es para llevar o para servirse, cada tarjeta debe tener un botón para confirmar si la orden ha sido cobrada.
Estado:	Completado exitosamente.
Observaciones:	Ninguna.

Nota. La tabla muestra la evaluación de prueba n. 07.

Tabla 22

Prueba N. 08

Evaluación de prueba	
Nombre:	Visualizar Libro de Ordenes.
Objetivos de la prueba:	Visualizar todas las ordenes que se han registrado. <ul style="list-style-type: none"> • Presionar botón Libro de Ordenes.
Procedimiento de la prueba:	<ul style="list-style-type: none"> • Visualización de la página de Libro de Ordenes. • Presionar fecha para detallar. • Visualización de los detalles de esa fecha. <p>Al presionar el botón Libro de Ordenes, se debe visualizar la</p>
Resultados esperados:	página Libro de Ordenes en donde se muestran todos los días registrados y el total de ingresos obtenidos.
Estado:	Completado exitosamente
Observaciones:	Al presionar en cualquier día, se debe visualizar todas las ordenes registradas en ese día.

Nota. La tabla muestra la evaluación de prueba n. 08.

Capítulo IV

Conclusiones y recomendaciones

Conclusiones

La implementación de un sistema de gestión de órdenes basado en una aplicación web y un servidor local puede mejorar la eficiencia de una cafetería - heladería al permitir una gestión más rápida y efectiva de las órdenes de los clientes. La automatización de procesos de registro y gestión de órdenes puede reducir el tiempo de espera para los clientes y, en consecuencia, aumentar la satisfacción de los mismos.

Un sistema de gestión de órdenes basado en una aplicación web y un servidor local puede reducir la posibilidad de errores humanos en la toma de órdenes y su registro. La aplicación puede ser diseñada para incluir campos de entrada obligatorios y para validar la información ingresada por los usuarios, lo que puede reducir el riesgo de errores en el registro de órdenes.

La implementación de una aplicación web para la gestión de órdenes de una cafetería - heladería también puede hacer que el servicio sea más accesible para los clientes y el personal de servicio. Los clientes pueden realizar sus órdenes de manera más rápida y sencilla, y los empleados pueden acceder a la información de las órdenes desde cualquier dispositivo con acceso a la red local, lo que les permite ofrecer un mejor servicio a los clientes.

Recomendaciones

La interfaz de usuario de la aplicación debe ser fácil de usar y comprender, tanto para los clientes como para el personal de servicio. La aplicación debe ser diseñada de manera que los usuarios puedan navegar de manera fluida y sin confusiones entre las diferentes secciones de la aplicación, como la selección de productos, la configuración de la orden y la realización del pago.

La aplicación debe incluir las funcionalidades esenciales para la gestión de órdenes, como la selección de productos, la inclusión de notas de pedido y la integración con un sistema de pago seguro. También se recomienda incluir una sección de estadísticas y reportes para que el personal pueda analizar y hacer seguimiento de los datos de ventas y de los patrones de pedidos de los clientes.

La aplicación debe ser diseñada y configurada de manera que garantice la seguridad de los datos de los clientes, así como la disponibilidad y el respaldo de los datos de las órdenes. Se recomienda la implementación de medidas de seguridad, como la encriptación de datos y la autenticación de usuarios, y la realización de copias de seguridad regulares para garantizar la integridad y disponibilidad de los datos. Además, es importante tener un plan de contingencia en caso de fallas o interrupciones en la red o en el servidor local.

Bibliografía

- Arías Becerra, J. C., & Durango Venegas, C. E. (2018). Propuesta de un método para desarrollar Sistemas de Información Geográfica a partir de la metodología de desarrollo ágil - SCRUM -. *Cuaderno Activa*, 29-41.
- Arias, Á. (2014). *Bases de Datos con MySQL: 2ª Edición*. IT Campus Academy.
- Capdevila, J. P. (Noviembre de 2001). *Sociedad de la información*. Obtenido de <http://www.sociedadelainformacion.com/12/tecnologiasweb.pdf>
- Casillas Satillán, L. A., Ginestá, M. G., & Mora, Ó. P. (2014). *Bases de datos en MySQL*. Universitat oberta de Catalunya.
- Corporation's, M. (29 de Noviembre de 2022). *mdn web docs*. Obtenido de https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks
- Durango, A. (2015). *Diseño Web con CSS: 2ª Edición*. IT Campus Academy.
- Escobar, A., Alvarez, D., & Alvarez, J. (2019). *Twitter Bootstrap Web Development*. Packt Publishing.
- Flores, A. (1995). *UN ANÁLISIS REGIONAL DE LA ARQUITECTURA*. Monterrey: INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTEREY.
- Fuentes, A., & Camen. (2013). *Diseño e implementación de una aplicación Android para la gestión de pedidos en restaurantes*. España: Master's thesis.
- Hernandez Galvis, J. C., & Ferro Hernandez, J. C. (2017). *Desarrollo de un Aplicativo Web para la Facturación, Control de Kardex y Apoyo al Manejo de Gastos de la Empresa "Empanadas de mi tierra y productos S.A.S."*. Bogota D.C: Universitaria Agustiniiana.

- Kinsta. (19 de Diciembre de 2022). *Kinsta*. Obtenido de <https://kinsta.com/es/base-de-conocimiento/next-js/>
- Lerma-Blasco, R. V., Murcia, J. A., & Talón, E. M. (2013). *Aplicaciones web*. España: McGraw-Hill/Interamericana de España.
- Lujan, M. S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. España, Alicante: Editorial Club Universal.
- Maida, E. G., & Pacienza, J. (Diciembre de 2015). *Metodologías de desarrollo de software*. Obtenido de Universidad Católica Argentina:
<https://repositorio.uca.edu.ar/handle/123456789/522>
- Mariño, S. I., & Alfonzo, P. L. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. *Scientia Et Technica*, 413-418.
- Marqués, M. (2011). *Bases de Datos*. Catello de la Plana.
- Nevado Cabello, V. (2010). *Introducción a las bases de datos relacionales*. Madrid: Vision Netware S.L.
- Orjuela Duarte, A., & Rojas C., M. (2008). Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo. *Revista Avances en Sistemas e Informática*, 159-171.
- Pisco Gómez, À., Regalado Jaica, J. J., Gutiérrez García, J., Quimis Sánchez, O., Marcillo Parrales, K., & Marcillo Merino, J. (2017). *Fundamentos sobre la gestión de base de datos*. Diciembre: Área de innovación y Desarrollo, S.L.
- Ricardo, C. M. (2009). *Bases de Datos*. Santa Fe, México: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V.

Rivas, C. I., Corona, V. P., Gutiérrez, J. F., & Hernández, L. (2015). Metodologías actuales de desarrollo de software. *Revista de Tecnología e Innovación*, 2-5.

Rodríguez, C., & Rubén Dorado, V. (2015). ¿Por qué implementar Scrum? *Revista ONTARE*, 125-144.

Sánchez Acosta, C., Tuesta Monteza, V., & Mejía Cabrera, I. (2015). Análisis comparativo de frameworks para el desarrollo de aplicaciones web en java. *Ingeniería: Ciencia, Tecnología e Innovación*, 60-60.

Tinoco, E. E. (2014). *Programación web con CSS, JavaScript, PHP y AJAX*. Iván Soria Solís.

Vertice, E. (2009). *Diseño básico de páginas web em HTML*. Editorial Vertice.

Anexos