



**UNIVERSIDAD NACIONAL DE INGENIERÍA
RECINTO UNIVERSITARIO “SIMÓN BOLÍVAR”
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**

MONOGRAFIA

**Sistema web de gestión de inventario y control de ventas
de la microempresa “Panadería hermanos Brenes”.**

**PARA OPTAR AL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

**ELABORADO POR:
Sofía Alejandra Miranda Valle**

**TUTOR:
MSc. Gabriel Rafael Lacayo Saballos**

**MANAGUA, NICARAGUA
MARZO 2023**

Dedicatoria

Dedico este trabajo monográfico principalmente a Dios, ya que gracias a él he logrado concluir mis estudios académicos, y a todas las personas que me han apoyado durante mi trayectoria universitaria.

Agradecimientos

Esta monografía fue un proceso de aprendizaje y constante motivación, es por ello que al finalizar este trabajo quiero expresar mi gratitud a mi familia, en especial a mi madre, hermana, a mis amigos, en particular a Eliezert Bolaños por tu invaluable compañerismo, finalmente y no menos importante a un tutor MSc. Gabriel Lacayo, quien por medio de sus enseñanzas y orientaciones ha permitido el desarrollo de este trabajo monográfico.

Índice de contenido

1. Introducción.....	1
2. Antecedentes	3
3. Justificación.....	4
4. Objetivos	6
5. Marco teórico	7
<i>5.1 Conceptualización del sistema</i>	7
5.1.1 Sistema web	7
5.1.2 inventario	7
5.1.3 Gestión de inventario	8
5.1.4 Sistemas de gestión de inventario	8
5.1.5 Control de ventas	9
<i>5.2 Herramientas de desarrollo</i>	9
5.2.1 Lenguaje de Programación	9
5.2.2 Framework	10
5.2.3 Programación estructurada.....	10
5.2.4 Lenguaje orientado a objetos.....	10
5.2.5 Entorno de desarrollo.....	11
5.2.6 Modelo de interfaz.....	11
5.2.7 UML	11
5.2.8 Lenguaje PHP	12
5.2.9 HTML	12
5.2.10 Bootstrap.....	13
5.2.11 CSS.....	13
5.2.12 JavaScript	13
5.2.13 Sublime Text	14
5.2.14 AJAX	14
<i>5.3 Modelo de bases de datos</i>	14
5.3.1 Bases de datos	14
5.3.2 MySQL/MariaDB	15
5.3.3 Programa Cliente	15
<i>5.4 Metodologías agiles</i>	15
5.4.1 Metodología XP	16
5.4.2 Roles	17
5.4.3 Fases del modelo	18
<i>5.5 Modelo vista controlador</i>	20
5.5.1 Modelo	21
5.5.2 Controlador	21
5.5.3 Vistas	22

5.6 <i>Trello</i>	22
5.7 <i>Modelado de procesos BPM (Business Process Management)</i>	22
5.8 <i>Bizagi</i>	23
6. Marco metodológico	23
6.1 <i>Metodología XP</i>	23
6.2 <i>Fases de la metodología XP</i>	24
6.2.1 <i>Fase de Análisis</i>	24
6.2.3 <i>Fase de Diseño</i>	24
6.2.4 <i>Fase de desarrollo</i>	25
6.2.5 <i>Fase de Prueba</i>	25
7. Análisis y presentación de los resultados.....	26
7.1 <i>Historias de usuario</i>	26
7.2 <i>Plan de entrega del proyecto</i>	31
7.3 <i>Primera Iteración</i>	32
7.3.1 Diseño de base de datos	32
7.3.2 Tareas de ingeniería 1ra iteración.....	34
7.3.3 Tarjetas CRC	39
7.3.4 Codificación	42
7.3.5 Bocetos	45
7.3.6 Capturas de pantalla del Sistema Informático.....	48
Ilustración 17 Captura de pantalla Vista agregar categoría	50
7.3.7 Pruebas de aceptación	50
7.3.8 Resultados de la primera iteración.....	56
7.4 <i>Segunda iteración</i>	56
7.4.1 Tareas de ingeniería iteración 2.....	57
7.4.2 Tarjetas CRC	62
7.4.3 Codificación	67
7.4.4 Bocetos	69
7.4.5 Capturas de pantalla	71
7.4.6 Pruebas de aceptación	74
7.5 <i>Tercera iteración</i>	77
7.5.1. Tareas de Ingeniería tercera iteración	77
7.5.2 Tarjetas CRC	83
7.5.3 Codificación	84
7.5.4 Bocetos	86
7.5.5 Capturas de pantalla	87
7.5.6 Pruebas de aceptación	89

8. Métricas.....	90
8.1 Definición de métricas	91
8.2 Definición de los instrumentos.....	92
8.3 Resultado de la prueba.....	93
8.3.2 Resultado de caso de prueba 2	94
8.4 Resultados de la encuesta	95
9. Conclusiones.....	97
10. Recomendaciones.....	98
11. Referencias bibliográficas	99
12. Anexos	102
Anexo A: Estudio de factibilidad.....	102
Anexo B: Ingeniería de requerimientos	120
Anexo C: Manual de usuario.....	124
Anexo D: Diccionario de datos.....	154

Índice de ilustraciones

Ilustración 1 Fases de la metodología XP	18
Ilustración 2 Patrón MVC en una aplicación Web	21
Ilustración 3 Diagrama de bases de datos	33
Ilustración 4 Codificación conexión de base de datos.....	42
Ilustración 5 Codificación ingresar categoría	44
Ilustración 6 Boceto acceso al sistema	45
Ilustración 7 Boceto agregar usuario.....	45
Ilustración 8 Boceto visualizar usuarios	46
Ilustración 9 Boceto página inicio.....	46
Ilustración 10 Boceto usuario especial.....	47
Ilustración 11 Boceto añadir categoría	47
Ilustración 12 Boceto administrar categoría	48
Ilustración 13 Captura de pantalla Vista login	48
Ilustración 14 Captura de pantalla Vista administrar usuarios	48
Ilustración 15 Captura de pantalla Vista Panel de control	49
Ilustración 16 Captura de pantalla Vista panel de control usuario E	49
Ilustración 17 Captura de pantalla Vista agregar categoría	50
Ilustración 18 Captura de pantalla Vista visualizar categoría	50
Ilustración 19 Codificación crear cliente Parte I	68
Ilustración 20 Codificación crear cliente Parte II	68
Ilustración 21 Boceto registro de cliente	69
Ilustración 22 Boceto administrar clientes.....	69
Ilustración 23 Boceto de inventario de productos existentes	70
Ilustración 24 Boceto registro cliente	70
Ilustración 25 Boceto gestión de ventas.....	70
Ilustración 26 Captura de pantalla registro de ventas	71
Ilustración 27 Captura de pantalla dataset clientes.....	72
Ilustración 28 Captura de pantalla añadir productos existentes	72

Ilustración 29 Captura de pantalla productos existentes	73
Ilustración 30 Captura de pantalla crear ventas	73
Ilustración 31 Codificación administración de ventas	85
<i>Ilustración 32 Generación reporte de ventas</i>	85
Ilustración 33 Boceto reporte de ventas	86
Ilustración 34 Boceto administrar ventas	86
Ilustración 35 Captura de pantalla administración ventas	87
Ilustración 36 Captura de pantalla generar reportes parte	87
I Ilustración 37 Captura de pantalla generar reportes parte II	88
Ilustración 38 Gráficos de la encuesta	96
Ilustración 39 Descripción del sistema manual	107
Ilustración 40 Modelado creación de ventas sistema manual	118
Ilustración 41 Modelado de ventas sistema informático implementado	119
Ilustración 47 Vista General del sitio web	125
Ilustración 48 Acceso	126
Ilustración 49 Inicio sesión	126
Ilustración 50 Vendedor especial	127
Ilustración 51 Inicio	127
Ilustración 52 Usuarios	128
Ilustración 53 Usuario Administrador	129
Ilustración 54 Usuario vendedor	129
Ilustración 55 Usuario especial	130
Ilustración 56 Agregar nuevo usuario	131
Ilustración 57 Mensaje de confirmación del gestor añadir usuario	131
Ilustración 58 Vista usuarios	132
Ilustración 59 Modificar un usuario	132
Ilustración 60 Eliminar un usuario	133
Ilustración 61 Revisión de usuario eliminado	133
Ilustración 62 Agregar categoría	134
Ilustración 63 Mensaje de éxito de agregar categoría	134
Ilustración 64 Administrar categoría	135

Ilustración 65 Modificar categoría	135
Ilustración 66 Guardar cambios de categoría.....	136
Ilustración 67 Eliminar categoría	136
Ilustración 68 Vista general de productos	137
Ilustración 69 Agregar un producto	137
Ilustración 70 Seleccionar un producto	138
Ilustración 71 Registrar la venta.....	138
Ilustración 72 Mensaje de confirmación de la venta registrada.....	139
Ilustración 73 Vista general de administrar productos.....	139
Ilustración 74 Modificar un producto	140
Ilustración 75 Eliminar un producto	140
Ilustración 76 Advertencia	141
Ilustración 77 Vista general clientes.....	141
Ilustración 78 Agregar cliente.....	142
Ilustración 79 Clientes agregados	142
Ilustración 80 Modificar un cliente	143
Ilustración 81 Eliminar un cliente.....	143
Ilustración 82 Vista general ventas	144
Ilustración 83 Agregar una venta	144
Ilustración 84 Seleccionar cliente.....	145
Ilustración 85 Método de pago parte I	146
Ilustración 86 Método de pago parte II	146
Ilustración 87 Crear venta	147
Ilustración 88 Mensaje de confirmación de un registro de venta	147
Ilustración 89 Impresión de factura	148
Ilustración 90 Formato de la factura del cliente.....	148
Ilustración 91 Modificar venta.....	149
Ilustración 92 Eliminar ventas	150
Ilustración 93 Reportes de ventas parte I.....	150
Ilustración 94 Reporte de ventas parte II	151
Ilustración 95 Reporte de ventas parte III	151

Ilustración 96 Reporte de ventas parte IV	152
Ilustración 97 Reporte de ventas parte V	152
Ilustración 98 Archivo Excel del reporte de ventas.....	153
Ilustración 99 Procesos en el panel de control del sistema.....	153
Ilustración 100 Diccionario de datos	155

Índice de tablas

Tabla 1 Plantilla de historia de usuario	26
Tabla 2 Historia de usuario acceso al sistema	27
Tabla 3 Historia de usuario Registro de usuarios.....	27
Tabla 4 Historia de usuario Registro de categorías	28
Tabla 5 Historia de usuario de inventario de productos existentes	28
Tabla 6 Historia de usuario registro cliente	29
Tabla 7 Historia de usuario gestión de ventas	29
Tabla 8 Historia de usuario administración de ventas.....	30
Tabla 9 Historia de usuario generación de reportes.....	30
Tabla 10 Plan de entrega de proyecto	31
Tabla 11 Plan de entrega de proyecto primera iteración.....	32
Tabla 12 Tareas de Ingeniería primera iteración.....	34
Tabla 13 Tarea de Ingeniería 1 Iteración 1 Diseño de login.....	35
Tabla 14 Tarea de Ingeniería 2 Iteración 1 Almacenamiento usuarios	36
Tabla 15 Tarea de Ingeniería 3 Iteración 1 Roles de usuario	36
Tabla 16 Tarea de Ingeniería 4 Iteración 1 Validación de usuarios	36
Tabla 17 Tarea de Ingeniería 5 Iteración 1 Diseño interfaz admin.....	36
Tabla 18 Tarea de Ingeniería 6 Iteración 1 Diseño de interfaz usuario E	37
Tabla 19 Tarea de Ingeniería 7 Iteración 1 Diseño categoría	37
Tabla 20 Tarea de Ingeniería 8 Iteración 1 Diseño gestión categoría.....	37
Tabla 21 Tarea de Ingeniería 9 Iteración 1 Validación gestión categoría	38
Tabla 22 Tarea de Ingeniería 10 Iteración 1 Validación pestaña categoría	38
Tabla 23 Tarea de Ingeniería 11 Iteración 1 Visualización de los registros	38
Tabla 24 Plantilla de tarjetas CRC	39
Tabla 25 Tarjeta CRC Tarea de Ingeniería 1,2,3 Login	39
Tabla 26 Tarjeta CRC Tarea de Ingeniería 4 Usuarios en BD	40
Tabla 27 Tarjeta CRC Tarea de Ingeniería 6 Diseño interfaz admin.....	40
Tabla 28 Tarjeta CRC Tarea de Ingeniería 6 Diseño interfaz usuario.....	40

Tabla 29 Tarjeta CRC Tarea de Ingeniería 7,8 Diseño interfaz categoría	41
Tabla 30 Tarea CRC Tarea de Ingeniería 9,10,11 Validación categoría.....	41
Tabla 31 Plantilla prueba de aceptación	51
Tabla 32 Pruebas de aceptación Iteración 1	52
Tabla 33 Prueba de aceptación acceso al sistema	53
Tabla 34 Prueba de aceptación almacenamiento de usuario	53
Tabla 35 Prueba de aceptación definición de roles.....	53
Tabla 36 Prueba de aceptación validación tipos de usuario	54
Tabla 37 Prueba de aceptación Diseño de interfaz admin.....	54
Tabla 38 Prueba de aceptación diseño de interfaz usuario especial	55
Tabla 39 Prueba de aceptación diseño de interfaz categoría	55
Tabla 40 Plan de entrega de proyecto Iteración 2.....	56
Tabla 41 Tareas de Ingeniería iteración 2.....	57
Tabla 42 Tarea de Ingeniería 1 Iteración 2 Módulos productos existentes	58
Tabla 43 Tarea de Ingeniería 2 Iteración 2 Visualización dataset.....	58
Tabla 44 Tarea de Ingeniería 3 Iteración 2 Modulo clientes	59
Tabla 45 Tarea de Ingeniería 4 Iteración 2 Validación del dataset clientes	59
Tabla 46 Tarea de Ingeniería 5 Iteración 2 Modulo creación de ventas.....	60
Tabla 47 Tarea de Ingeniería 6 Iteración 2 Validación del dataset ventas	60
Tabla 48 Tarea de Ingeniería 7 Iteración 2 Automatización gestión ventas	61
Tabla 49 Tarea de Ingeniería 8 Iteración 2 Validación gestión ventas	61
Tabla 50 Tarjeta CRC Tarea de Ingeniería 1 Modulo productos existentes.....	62
Tabla 51 Tarjeta CRC Tarea de Ingeniería 2 Validación dataset productos	63
Tabla 52 Tarjeta CRC Tarea de Ingeniería 3 Modulo clientes	63
Tabla 53 Tarjeta CRC Tarea de Ingeniería 4 Validación dataset clientes	64
Tabla 54 Tarjeta CRC Tarea de Ingeniería 5 Modulo ventas	65
Tabla 55 Tarjeta CRC Tarea de Ingeniería 6 Validación dataset ventas	66
Tabla 56 Tarjeta CRC Tarea de Ingeniería 7,8 Gestión de ventas	66
Tabla 57 Pruebas de aceptación segunda iteración	74
Tabla 58 Prueba de aceptación inventarios de productos existentes	74
Tabla 59 Prueba de aceptación registro clientes	75

Tabla 60 Prueba de aceptación gestión de ventas	76
Tabla 61 Plan entrega de proyecto Iteración 3.....	77
Tabla 62 Tareas de Ingeniería Tercera Iteración	77
Tabla 63 Tarea de Ingeniería 1 Iteración 3 Diseño de interfaz de ventas	78
Tabla 64 Tarea de Ingeniería 1 Iteración 3 Codificación procesos de venta.....	79
Tabla 65 Tarea de Ingeniería 3 Iteración 3 Validación procesos ventas	79
Tabla 66 Tarea de Ingeniería 4 Iteración 3 Diseño factura	79
Tabla 67 Tarea de Ingeniería 5 Iteración 3 Impresión recibos	80
Tabla 68 Tarea de Ingeniería 6 Iteración 3 Stock clientes	80
Tabla 69 Tarea de Ingeniería 7 Iteración 3 Histograma	81
Tabla 70 Tarea de Ingeniería Iteración 3 Diseño interfaz reportes	81
Tabla 71 Tarea de Ingeniería 9 Iteración 3 Reportes en Excel	81
Tabla 72 Tarea de Ingeniería 10 Iteración 3 Codificación subprocessos ventas	82
Tabla 73 Tarea de Ingeniería 11 Iteración 3 Validar subprocessos ventas	82
Tabla 74 Tarjeta CRC administración de ventas.....	83
Tabla 75 Tarjeta CRC reporte de ventas	84
Tabla 76 Pruebas de aceptación de la tercera iteración	89
Tabla 77 Descripción de la prueba administración de ventas	89
Tabla 78 Descripción de la prueba de reportes.....	90
Tabla 79 Definición de los indicadores de evaluación.....	92
Tabla 80 Resultados de caso de prueba 1.....	93
Tabla 81 Resultado caso de prueba 2	94
Tabla 82 Resultado de la encuesta.....	95
Tabla 83 Aspectos técnicos del hardware.....	103
Tabla 84 Especificaciones del software	105
Tabla 85 Salario del personal involucrado en el sistema manual.....	108
Tabla 86 Costos mensuales sistema manual.....	108
Tabla 87 Costos totales sistema manual	109
Tabla 88 Salario del personal involucrado en el sistema informático.....	110
Tabla 89 Descripción de los costos de recursos	110
Tabla 90 Costos sistema manual.....	112

Tabla 91 Costos sistema informático	113
Tabla 92 Comparación de costos de ambos sistemas.....	114
Tabla 93 Símbolos de los diagramas de flujo	116
Tabla 94 Requerimientos funcionales del sistema	120
Tabla 95 Requerimientos no funcionales del sistema	122

1. Introducción

La industria alimentaria es cada vez más amplia y multidisciplinaria, el pan es de vital importancia en la gastronomía, por lo tanto, las panaderías juegan un rol fundamental en la sociedad con un rubro de trabajo tan antiguo como el ser humano mismo. Dentro del sector laboral tiene gran importancia en el mercado, en términos económicos generan empleo aumentando potencialmente las perspectivas de crecimiento para Nicaragua, es por ello que mediante la incorporación de herramientas tecnológicas se pretende implementar un sistema web que facilite a la microempresa panadería Hermanos Brenes llevar un control más eficiente de los servicios que ofrecen.

El presente documento tiene como finalidad exponer la propuesta del desarrollo, un sistema web para el establecimiento Panadería Hermanos Brenes, dicho sistema a rasgos generales garantizará un alto nivel de calidad en la atención de los consumidores a la hora de solicitar cualquier servicio a la panadería, esto mediante la funcionalidad de operaciones productivas presentes en el sistema web, de tal forma que se aumente el rendimiento por la demanda de pan en el área de negocio.

Dicho documento se ha seccionado en tres partes fundamentales:

- **Anteproyecto:**

Esta sección contiene una breve presentación de la empresa, la problemática que presenta y las causas que impulsaron a desarrollar el proyecto informático como solución al problema presentado, por otra parte, le acompaña un sustancial marco teórico y un estudio de factibilidad, esto con el propósito de introducir al estudio preliminar del proyecto y la viabilidad del mismo.

- **Análisis y Diseño del sistema:**

Esta parte se identifican los procesos y procedimientos que estarán presentes durante el desarrollo del sistema informático, así como la metodología implementada para agilizar, organizar y garantizar el

resultado esperado de cada interacción de entrega entre la empresa y el grupo de desarrollo, las cuales se van ajustando a la visión del producto final, simultáneamente se incluyen la elaboración de los diagramas, casos de uso y diseño de bases de datos.

- **Implementación del sistema:**

Finalmente se realizan pruebas del sistema y se describen una serie de actividades, tareas y procesos a realizar mediante manuales de usuarios y manuales técnicos, para el uso correcto del sistema de información.

Esta propuesta de sistema web incorpora todos los procedimientos aplicados en cuanto de atención al cliente se trata; procesos tales como: gestión de ventas, categorías de productos, stock de productos existentes, gestión de usuarios, gestión de clientes, reportes de ventas. Se propone un sistema informático con el propósito de satisfacer las necesidades actuales del establecimiento, reemplazando modos de trabajo manuales al automatizar tareas fundamentales presentes en la panadería, beneficiando de esta manera tanto a los clientes como al negocio.

2. Antecedentes

Panadería Hermanos Brenes fue fundada en 1980 por el patriarca Robert Ernesto Brenes Velásquez, con más de 40 años al servicio de la población masatepina y sus alrededores. En su principio era un negocio familiar, pequeño, con un personal limitado, el cual se encontraba conformado por tan solo cuatro trabajadores. La panadería recibió gran aceptación por parte de los consumidores, año tras año el negocio crecía, así como la necesidad de encontrar un método para archivar la información de este.

En 1990 se implementó como método de tratamiento de la información de la empresa un libro de cuentas, en este libro la persona encargada de atender las ventas procedía a anotar cada movimiento que realizaban los trabajadores en la panadería, posteriormente migraron este libro de cuentas a un pequeño sistema de escritorio codificado en Access. Con el paso de los años el sistema fue inoperable debido a la falta de compatibilidad de la aplicación con las nuevas versiones de sistemas operativos y frameworks de desarrollo, regresando nuevamente a utilizar el libro manual de cuentas.

Actualmente Hermanos Brenes cuenta con un personal de veinticinco trabajadores y no poseen un método sistematizado, digitalizado o informático que les permita llevar un control de sus ganancias y servicios que ofrece la panadería. Es relevante mencionar que desde sus inicios la Panadería Hermanos Brenes se ha mostrado interesada en gestionar el área de inventario: registro de clientes, facturación, cobros y en un intento de llevar el control de servicios de manera manual han hecho un registro de facturas membretadas que posteriormente son archivadas.

3. Justificación

La gran competencia en el sector panadero y las exigencias del mercado hacen imprescindible estar en constante actualización respecto a los avances tecnológicos. Absolutamente todas las organizaciones requieren sistemas de información durante toda la existencia empresarial que estas tengan. Presentamos esta propuesta al notar la ineficiencia que tiene la panadería Hermanos Brenes al momento de realizar cualquier operación o procesamiento de datos de algún servicio ya que al ser un local concurrido necesita un método lo suficientemente eficaz al momento de manipular datos.

La aplicación a desarrollar es un sistema web que gestione todas aquellas funciones necesarias para la agilización del tratamiento de información mediante la automatización de procesos estratégicos para la empresa, dichos procesos cubren todo el ciclo de vida de cada operación desde una venta realizada hasta el registro de un nuevo ingrediente ya sea añadido o dado de baja del stock. A través del nuevo registro se mantendrá un control fijo de la información, o sea estará disponible en todo momento, de tal forma que se pueda acceder a la información cuando sea pertinente, igualmente se pretende optimizar datos y procesos que actualmente no se encuentran informatizados.

Disponer de un sistema web permite manipular mejor la información agilizando la toma de decisiones dentro de la organización, minimizando el tiempo de elaboración de informes pertinentes. Se opta por un SIW (Sistema de información web) como respuesta a la necesidad por la que estaba atravesando la empresa, como la más viable solución ante esta problemática. Existen otras soluciones como por ejemplo utilizar un sistema de planificación de recursos empresariales, sin embargo, se descartó esta opción debido a que la panadería tiene procesos propios con lo que el sistema ERP no cuenta.

La implementación de este sistema de gestión de inventario generará a futuro múltiples beneficios, como la automatización de los procesos de la panadería, tales como: facturación, generación de informes, estados de cuentas y servicios, además permitirá concentrar toda la información de la panadería con el propósito de ser resguardada de manera segura.

4. Objetivos

Objetivo General

Desarrollar un sistema web para la gestión de inventario de productos existentes y control de ventas de la microempresa panadería Hermanos Brenes.

Objetivos específicos

- Analizar los requerimientos y alcances del sistema web para la gestión de ventas panadería Hermanos Brenes.
- Diseñar un modelo de interfaz a partir de los requerimientos iniciales usando una herramienta de diseño UML.
- Codificar en base al diseño todos los procesos y funciones que estarán presentes dentro del sistema web, utilizando el lenguaje de programación PHP y MySQL server.
- Implementar el sistema web de gestión de ventas realizando las pruebas de aceptación para garantizar el correcto funcionamiento.

5. Marco teórico

El marco teórico es el punto de partida para un proyecto de investigación, es el "mapa" que guía la información y los datos que se utilizan a lo largo del desarrollo de la tesis.

5.1 Conceptualización del sistema

En esta sección se definen cada uno de los conceptos relevantes utilizados durante el desarrollo de este proyecto.

5.1.1 Sistema web

Se denomina sistema informático o sistema web a aquellas aplicaciones en donde los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador [1], en otras palabras, es un conjunto de datos que interactúan entre sí con un fin común, dando como resultado una estructura organizada de datos que sirve de apoyo a actividades comerciales, educativas, industriales, etc.

Los sistemas informáticos son los sistemas encargados de recibir, guardar y procesar información para posteriormente entregar resultados a partir de ello [2], en otras palabras ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización. debido a lo práctico del navegador web como Cliente ligero, a la independencia del Sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

5.1.2 Inventario

La clave es lograr un equilibrio entre las salidas (ventas) y las entradas (compras) de material [2], con respecto a lo anterior podemos decir que un inventario es la solución más eficaz para potenciar cualquier negocio

siendo este una es una herramienta clave y eficaz para aumentar las ganancias de una empresa. Un inventario consiste en el registro de todos los productos que se tienen en existencia. Si no se tiene un inventario, no se puede vender nada, son de gran importancia y utilidad para realizar previsiones de ventas, análisis de los datos, control de suministros existentes y monitoreo de los productos.

5.1.3 Gestión de inventario

El objetivo principal de una gestión de inventario es garantizar la existencia de los productos así como la disponibilidad de estos, la reserva y la financiación [2], dicho lo anterior se define como gestión de inventario a una serie de procesos que permiten administrar correctamente los productos existentes de modo que se regula el flujo entre las entradas y las salidas de existencias de los productos de una empresa. La forma de regular el flujo de entrada es variando la frecuencia y el tamaño de los pedidos que se realicen a los proveedores.

5.1.4 Sistemas de gestión de inventario

Los sistemas de gestión de inventario también conocidos como "sistemas administrativos" o "aplicaciones administrativas" son aquellas que están creadas, instaladas y finalmente alojadas en un servidor en internet o sobre en una intranet (red local). Su aspecto es muy similar a páginas web que vemos normalmente, pero en realidad los 'sistemas web' tienen funcionalidades muy potentes que brindan respuestas a casos particulares, dichos sistemas se pueden utilizar en cualquier navegador web (Chrome, Firefox, Internet Explorer, etc.) sin importar el sistema operativo que posea el computador [3]. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor en donde se aloja el sistema, este tipo de diferencias se ven reflejada en los costos, la rapidez de obtención de la

información, la optimización de las tareas por parte de los usuarios y en alcanzar una gestión estable.

5.1.5 Control de ventas

El control de ventas es un registro que asegura la productividad de la fuerza de ventas de una empresa, asociado a los productos y los vendedores. Este control permite que un negocio se mantenga alerta y coherente en sus acciones, así como vender más, siendo las ventas parte fundamental en cualquier negocio [2], es por ello que llevar un control de estas representa una herramienta de gran utilidad para al empresario , administrador u operador permitiendo obtener una amplia visibilidad sobre los procesos de ventas, además mantiene la transparencia de todas las actividades de un vendedor con el propósito de eliminar acciones mal intencionadas como desviaciones negativas; en caso de presentarse, estas pueden analizarse y corregirse a tiempo y adecuadamente.

5.2 Herramientas de desarrollo

Los seres humanos, usando sus habilidades analíticas y de resolución de problemas, inventaron los algoritmos (grupo limitado de instrucciones que trabajan con una serie de datos de entrada para producir un resultado) [5] esto se logra utilizando cualquier herramientas de desarrollo. Las herramientas web constituyen parte fundamental al momento de codificar cualquier sistema informático. A continuación, se definen las herramientas requeridas al momento de desarrollar el sistema web.

5.2.1 Lenguaje de Programación

Es un modelo computacional el cual posee una colección de valores y operaciones [4] , hay que mencionar además que los lenguajes de programación proporcionan al programador la capacidad de escribir una serie de instrucciones con el fin de controlar el comportamiento físico y lógico de un sistema informático.

En términos generales, un lenguaje de programación es una herramienta que permite desarrollar software o programas para computadora. Los lenguajes de programación son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora [7], en resumidas palabras, un lenguaje de programación es la comunicación e iteración entre ser humano y computadora.

5.2.2 Framework

Se define frameworks como un marco de trabajo que se utiliza para desarrollar un software determinado [5] . Si se lleva este concepto al mundo de la creación de páginas web, se puede decir que un frameworks es un entorno de trabajo que hace mucho más fácil la programación o desarrollo de una web, cuya función es proporcionar soluciones y herramientas específicas reutilizables de un código envuelto con una interfaz de programación.

5.2.3 Programación estructurada

La programación estructurada es aquella que se divide en bloques [4] dichos bloques son una serie de procedimientos y funciones a los que se le atribuyen el control secuencial de cada sección e iteración permitiendo reutilizar código programado. La programación estructurada está basada en módulos funcionales bien marcados, generalmente jerarquizados de acuerdo al tipo específico del problema. [9]

5.2.4 Lenguaje orientado a objetos

Es un paradigma en el cual los programas se ven formados por entidades llamados objetos [6], es decir, los datos interactúan entre si y son capaces de reaccionar y cambiar su estado interno en función de las peticiones que reciben. La programación orientada a objetos proporciona conceptos y

herramientas que permiten a los programadores representar el mundo real tan fielmente como sea posible. [11]

5.2.5 Entorno de desarrollo

Un entorno de desarrollo es una plataforma diseñada para facilitar la labor del programador de software, también se conoce como Integrated Development Environment IDE, se traduce entorno de desarrollo integrado [7] , por lo general un IDE combina varias herramientas que permiten automatizar una buena parte de las tareas y fases de programación, como el análisis, diseño de arquitectura, codificación, pruebas, validación, gestión y mantenimiento. En algunos lenguajes un IDE puede funcionar como un sistema en tiempo de ejecución en donde permite utilizar lenguajes de programación en forma interactiva, sin necesidad de trabajo orientado a archivo de texto.

5.2.6 Modelo de interfaz

Un modelo de una Interfaz es una representación visual [8] en donde se le intenta mostrar al cliente cómo interactuará el usuario con la aplicación, y cómo responderá ésta. Las técnicas de modelado de las interacciones pueden describir objetos, tareas y diálogos.

5.2.7 UML

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas [9]. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene cabe destacar que UML no es un lenguaje de programación, pero existe en herramientas que se pueden usar para generar código en diversos

lenguajes usando los diagramas UML guardando una relación directa con el análisis y el diseño orientados a objetos.

5.2.8 Lenguaje PHP

PHP es el lenguaje de scripting, multipropósito que está situado especialmente para el desarrollo de páginas web [10]. Su claridad en el diseño, módulos bien organizados y mejor mantenimiento de tecnologías, lo hacen el lenguaje más popular en la industria actual. Su popularidad y credibilidad puede estar relacionada al hecho de que organizaciones reputadas como la universidad de Harvard o la red social Facebook, están basadas en PHP [11]. Esto es posible porque los sitios PHP pueden ser fácilmente mantenidos, mejorados y actualizados de vez en cuando. Utilizaremos el lenguaje PHP por su estilo basado en servidores además que con esta herramienta crearemos nuestra página de inicio de sesión con usuario y contraseña además de los detalles de formulario con el haremos la conexión con nuestra base de datos.

5.2.9 HTML

HTML es un estándar que sirve como referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros [12]. Se utilizará HTML debido a las mejoras en la creación de la estructura del código web y en el manejo óptimo de las etiquetas web. De esta manera, se trabaja con un estándar mucho más versátil, que permitirá realizar una interacción mucho más poderosa y simple, mejorando la experiencia de uso por parte del usuario y facilitando la depuración del código web.

5.2.10 Bootstrap

Bootstrap es uno de los frameworks CSS de código abierto más conocidos y más utilizados en el mundo del desarrollo web. Este framework facilita enormemente la maquetación de páginas web, ya que permite crear una interfaz muy limpia y completamente responsive [13], es decir, adaptable a cualquier tamaño de pantalla. Además, ofrece tantas herramientas y funcionalidades que permite crear una web desde cero muy fácilmente.

5.2.11 CSS

CSS3 es una tecnología que ha tenido una evolución en el tiempo, actualmente se encuentra en su versión 3, como su propio nombre indica. Es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado [14]. Se implementará CSS3 porque incorpora mejoras mayormente en el ámbito visual. El concepto en el que se basa este software es el modelo de cajas flexibles (flexbox), el cual básicamente es un modo de diseño que nos permitirá colocar los elementos de una página para que se comporten de forma predecible cuando el diseño de la página deba acomodarse a diferentes tamaños de pantalla y diferentes dispositivos.

5.2.12 JavaScript

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios [15]. Se utiliza JavaScript porque es muy versátil, puesto que es muy útil para desarrollar páginas dinámicas y es soportado por los navegadores más populares.

5.2.13 Sublime Text

Sublime Text es un software de tipo editor de código; sofisticado y multiplataforma, una herramienta concebida para programar sin distracciones, navegar por el código, y moverse por el código de forma rápida [16]. Además, destaca por su kit de herramientas simultáneamente con su interfaz de usuario, es por esta razón que se utilizará el (IDE) Sublime Text como editor de código ya que por la velocidad que posee lo posiciona en uno de los editores más eficientes de utilizar.

5.2.14 AJAX

Es una técnica de desarrollo web para crear aplicaciones asíncronas, válida para múltiples plataformas [17], como se ha dicho Ajax es un método el cual consiste en mantener una comunicación asíncrona entre el servidor y el navegador, ejecutado por el cliente mediante la interactuación con el servidor sin necesidad de recargar la página web, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

5.3 Modelo de bases de datos

Un modelo de base de datos es una estructura lógica [18] utilizada por una base de datos, incluidas las relaciones y restricciones que determinan cómo se almacenan, organizan y acceden a los datos. Asimismo, el modelo de base de datos define qué tipos de operaciones se pueden realizar sobre los datos, es decir, también determina cómo se manipulan los datos y proporciona la base para diseñar un lenguaje de consulta.

5.3.1 Bases de datos

Se entiende como base de datos a un determinado conjunto de datos estructurados y almacenados de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede por lo tanto constituirse de cualquier tipo de datos como datos numéricos, datos alfanuméricos, datos espaciales [19].

Los elementos clave de la base de datos son esa estructuración y sistematicidad pues ambas son las responsables de las características que hacen que las bases de datos tengan un enfoque superior a la hora de gestionar datos.

5.3.2 MySQL/MariaDB

MySQL es un sistema gestor de bases de datos (SGBD) [18] cabe señalar que es el más utilizado, por excelencia, constituido por un conjunto completo de funciones avanzadas, herramientas de administración y soporte técnico para el tratamiento de registros

cuyos principales objetivos son lograr los niveles más altos de escalabilidad, seguridad, confiabilidad y tiempo de actividad en MySQL. MariaDB es una bifurcación de MySQL cuya función principal es funcionar como server binario de MySQL dentro de la herramienta phpMyAdmin/Xampp.

5.3.3 Programa Cliente

Un programa tipo cliente son aquellos programas o máquinas que acceden de forma concurrente a un gestor que gestionaría un sistema de base de datos que habilita y controla las comunicaciones de acceso a aquellos que tuvieran los permisos adecuados [20].

5.4 Metodologías agiles

Las metodologías son aquellas que se encargan de adaptar el proceso del trabajo a las circunstancias y contexto en el que se encuentra, para que, si ocurre algún inconveniente o cambio inesperado en el panorama, los procedimientos en la empresa puedan adaptarse con facilidad y de manera inmediata, así el proyecto no se vea afectado negativamente de ninguna manera, [21], Funcionan como una serie de principios y valores que tienen como principal característica realizar entregas rápidas y continuas de software funcionando [22] es decir es un marco de trabajo en donde deben entregarse en cortos plazos partes funcionales del proyecto

cuyo objetivo es desarrollar productos y servicios de calidad que respondan a las necesidades de unos clientes, partiendo de esto se optó por utilizar la metodología ágil XP para poder gestionar el desarrollo del sistema web con mayor eficacia y eficiencia impulsando la productividad de las actividades que se realizaran durante el proceso de desarrollo.

Como principales ventajas se tienen:

- Mejora la calidad: Minimiza los errores en los entregables y mejora la experiencia y las funcionalidades para el cliente.
- Mayor compromiso: Mejora la satisfacción del empleado y genera conciencia de equipo.
- Rapidez: Acorta los ciclos de producción y minimiza los tiempos de reacción y toma de decisiones.
- Aumento de la productividad: Al asignar mejor los recursos, y de forma más dinámica, mejora la producción según las prioridades que tenga la empresa.

5.4.1 Metodología XP

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos [23]. Extreme Programming se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo. Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes, razones principales por la cual se optó por la implementación de ella, se basa en tres elementos para funcionar correctamente: los valores, los principios y las prácticas. El proceso de trabajo es circular y empieza con la planificación, seguido por el diseño, desarrollo y las pruebas, entre los principales principios se tienen:

- Realimentación rápida
- Asumir simplicidad

- Cambio incremental
- Abrazar el cambio
- Trabajo de calidad

5.4.2 Roles

Se define como rol a una determinada función que desempeña una persona en un lugar o situación. Dentro de la metodología Xp existen distintos tipos de roles con el fin de definir un encargado para tarea en específico, las tareas pueden ser rotativas, individuales y algunas combinadas desempeñadas por una o más personas.

- Tracker: es el encargado de seguimiento. Proporciona retroalimentación al equipo. Debe verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.
- Cliente: Es quien escribe las historias de usuario y las pruebas funcionales para validar su implementación. Asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar el mayor valor de negocio.
- Programador: Es el que escribe las pruebas unitarias y el código del sistema.
- Coach: es responsable del proceso global y se encarga de guiar a los miembros del equipo para seguir el proceso correctamente.
- Tester: ayuda al cliente a escribir las pruebas funcionales. Ejecuta pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- Big Boss: es el dueño del establecimiento, representa un vínculo entre clientes y programadores. Su labor esencial es la coordinación.
- Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto ayuda al equipo a resolver un problema específico y puede que no siempre haya un consultor como parte del equipo.

- Manager: agenda las reuniones y se asegura de que el proceso de juntas sea seguido, registra los resultados de las reuniones para futuros reportes para el Tracker. Asiste a las reuniones y trae información importante con el propósito de mantener al equipo de trabajo productivo.
- Doomsayer: se asegura de que todos los miembros conozcan los riesgos del proyecto, siendo responsable de los cambios al proyecto gracias a los riesgos.

5.4.3 Fases del modelo

El modelo de desarrollo XP sigue una serie de etapas de forma sucesiva, la etapa siguiente empieza cuando termina la etapa anterior. Las fases que componen el modelo son las siguientes:

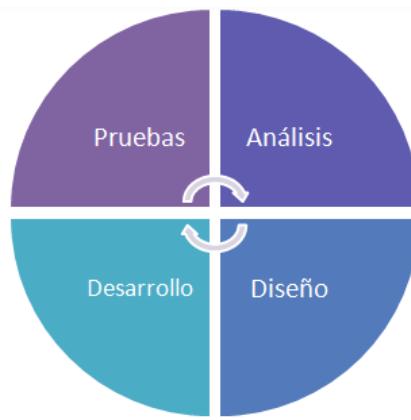


Ilustración 1 Fases de la metodología XP

Fuente: Metodología XP, la metodología de desarrollo más exitosa (2016), <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/>

5.4.3.1 Análisis

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas

semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. [22]

5.4.3.2 Diseño

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

Una entrega debería obtenerse en no más de tres meses y 12 iteraciones [23]. Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El plan de entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio).

5.4.3.3 Desarrollo

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase [23]. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de prueba).

5.4.3.4 Pruebas

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente [23]. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

5.4.3.5 Documentación

Esta última fase no pertenece a las etapas de iteración del modelo XP, sino es la fase final, es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema [23]. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

5.5 Modelo vista controlador

El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que utilizando 3 componentes (Vistas, Modelos y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación [24], este patrón es aplicado en muchos lenguajes de programación, uno de ellos es PHP.

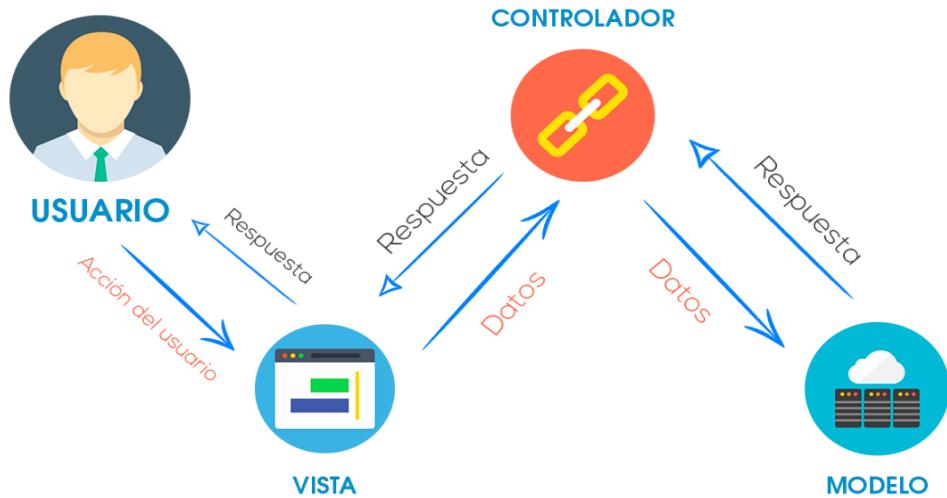


Ilustración 2 Patrón MVC en una aplicación Web

Fuente: TecnologySoft, Precognis (2022),

<https://es.linkedin.com/pulse/modelo-vista-controlador-precognis>

El funcionamiento es simple, el usuario realiza una acción desde un formulario de la aplicación ya sea selección, registro, eliminación o actualización de datos que será enviada desde la vista hacia el controlador, este se encargara de enviar los datos necesarios al modelo el cual se encargara de procesar los datos en la base de datos. El modelo enviará una respuesta de dicha acción al controlador el cual se encargará de regresar esa respuesta a la vista donde fue enviada la petición por parte del usuario [25].

5.5.1 Modelo

Es la parte que se encarga de interactuar directamente con la base de datos, es decir, todo el código PHP que hará las inserciones, selecciones, actualizaciones o eliminaciones de datos [24].

5.5.2 Controlador

En esta parte se encuentra toda la interfaz gráfica de la aplicación con la cual el usuario interactúa. Acá se encuentra el código CSS, HTML y

JavaScript, además todos los recursos como las imágenes, fuentes tipográficas, iconos , etc. [24].

5.5.3 Vistas

Es el que interactúa como intermediario entre el modelo y la vista. Es decir, es el que recibe las peticiones enviadas desde la vista para enviárselas al modelo para luego enviar la respuesta del modelo hacia la vista. [24]. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista. Utilizar el modelo MVC es importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los frameworks modernos utilizan MVC (o alguna adaptación del MVC) para la arquitectura.

5.6 Trello

En definitiva, la gestión de proyectos y equipos puede parecer una tarea abrumadora, por fortuna existen diferentes herramientas que facilitan la administración del flujo de trabajo. Trello es una aplicación de administración de proyectos que permite organizar y gestionar actividades, estableciendo prioridades de tareas [26] con el fin de mejorar el desempeño del equipo durante el desarrollo del proyecto. Todas las actividades relacionadas con el proyecto se muestran en una única interfaz que es visible para todos los miembros del equipo que forman parte del proyecto, esto se lleva a cabo mediante el uso de tableros o plantillas en donde se organiza visualmente el proyecto, posteriormente se dividen y subdividen las tareas.

5.7 Modelado de procesos BPM (Business Process Management)

El modelado de procesos BPM (*Business Process Management*) es un conjunto de actividades que deben seguirse a fin de permitir la creación de uno o más modelos para la representación, comunicación, análisis, diseño de un proyecto [27], cuyo principal objetivo es comprender el

funcionamiento interno de la organización de los procesos y mecanismos utilizados , para ello se utilizan diferentes herramientas graficas tales como diagramas , flujos de procesos.

5.8 Bizagi

Bizagi es un software de mapeo de procesos [28] cuya función es implementar modelados de procesos de tipo BMP (*Business Process Management*) , una herramienta para la elaboración de diagramas de procesos.

6. Marco metodológico

Se trata del proceso cuya finalidad es desarrollar productos o soluciones para un cliente o mercado en particular, teniendo en cuenta factores como los costos, la planificación, la calidad y las dificultades asociadas. A todo esto, es a lo que denominamos metodologías de desarrollo de software [21]. Es decir, se trata del proceso que se suele seguir a la hora de diseñar una solución o un programa específico. Tiene que ver, por tanto, con la comunicación, la manipulación de modelos y el intercambio de información y datos entre las partes involucradas. En esta sección se define los conceptos de la metodología a implementar.

6.1 Metodología XP

Se decidió implementar la metodología extreme mejor conocida como metodología XP durante el desarrollo del sistema web debido a que es una de las metodologías de desarrollo más exitosas, utilizada especialmente para proyectos de desarrollo a corto plazo que consiste en una programación rápida o extrema cuya particularidad es tener como parte del equipo al usuario final.

6.2 Fases de la metodología XP

Una fase implica tareas, estados y procesos, es decir una fase es realizar una serie de actividades en un periodo de tiempo específico y concreto, en donde cada tarea es dividida en etapas de desarrollo con la intención de mejorar la productividad hasta lograr el resultado deseado. A continuación, se describe cada una de las fases que constituyen la metodología XP.

6.2.1 Fase de Análisis

En esta fase se estudian las necesidades del cliente y se analiza la información proporcionada por este, posteriormente se define el problema; todo esto se realiza mediante ciclos de desarrollo cortos llamados iteraciones, los cuales al final generan entregables funcionales. Una vez definido el problema se determinan los requerimientos para llevar a cabo el proyecto simultáneamente se definen las entidades que estarán presentes en el sistema web a través de las historias de usuario. Las historias de usuario resumen la necesidad concreta de este al utilizar un producto o servicio, así como la solución que la satisface. Estipuladas las historias de usuarios se designa la prioridad que tendrá cada una de ellas y finalmente se especifica las entregas de los alcances de cada una.

6.2.3 Fase de Diseño

En esta fase se proyecta el diseño del sistema web, procurando que sea lo más sencillo posible, esto con el propósito de conseguir una aplicación fácilmente entendible y aceptada por el usuario final haciendo uso de metáforas. Las metáforas producen una descripción simple y entendida de lo que hay que construir.

6.2.4 Fase de desarrollo

En esta fase es en donde se codifica la solución, de ser necesario de modifica el código para optimizar el correcto funcionamiento de la aplicación web.

6.2.5 Fase de Prueba

Posterior a la codificación se realizan pruebas unitarias a cada una de las historias de usuario validando así la aceptación del cliente a cada una de las soluciones codificadas para cada historia de usuario. Una vez el cliente este conforme con todas las soluciones propuestas se presenta formalmente el sistema y se verifican las funcionalidades que este provea.

6.2.6 Fase de la documentación

Finalmente, en esta fase se elabora un documento final para todo el sistema, con el propósito de proporcionar información sobre todas las operaciones y procesos que están presentes en el.

7. Análisis y presentación de los resultados

El análisis de resultados es la parte final y conclusiva de cualquier investigación; en el que se procesar y presenta toda la información de manera ordenada y comprensible y finalmente se llega a las conclusiones que dichos análisis originan.

7.1 Historias de usuario

Las historias de usuario son una explicación en general de una función del software desde la perspectiva del usuario final, las historias de usuarios de la microempresa Panadería Hermanos Brenes son las siguientes:

- Acceso al sistema (credenciales autorizadas)
- Registro de usuarios
- Registro de categorías
- Inventario de productos existentes
- Registro de clientes
- Gestión de ventas
- Administración de ventas
- Generación de reportes

El modelo de la plantilla que se utilizó para la elaboración de las historias de usuario y sus componentes es la siguiente,

Tabla 1 Plantilla de historia de usuario

Historia de Usuario	
Número 1: Permite identificar a una historia de usuario.	Usuario: Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario
Nombre de la historia: Describe de manera general a una historia de usuario.	
Prioridad: Grado de importancia que el cliente asigna a una historia de usuario	Riesgo en desarrollo: Valor de complejidad que una historia de usuario representa al equipo de desarrollo.

Iteración asignada 1: Número de iteración, en que el cliente desea que se implemente una historia de usuario.
Programador: Persona encargada de codificar la historia de usuario
Descripción: Información detallada de una historia de usuario.
Observaciones: Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de una historia de usuario.

A continuación, en las tablas 2-9 se muestran las historias de usuario que fueron empleadas para llevar a cabo el desarrollo del sistema informático.

Tabla 2 Historia de usuario acceso al sistema

Historia de Usuario	
Numero 1	Usuario: Administrador, Vendedor, Especial
Nombre de la historia: Acceso al sistema	
Prioridad: Alta (Alta, media, baja)	Riesgo en desarrollo: Medio (Alto, medio, baja)
Iteración asignada 1	
Descripción: Los usuarios podrán acceder al sistema mediante credenciales previamente designadas por los desarrolladores: el perfil contara de un nombre y una contraseña	

Tabla 3 Historia de usuario Registro de usuarios

Historia de Usuario	
Numero 2	Usuario: Administrador
Nombre de la historia: Registro de usuario	
Prioridad: Alta (Alta, media, baja)	Riesgo en desarrollo: Alta (Alto, medio, baja)

Iteración asignada 1	
Descripción: El administrador será el encargado de asignar las credenciales a los demás operadores del sistema, así como también designará los permisos de acceso al sistema	
Observaciones: En dependencia del cargo que ejerzan en la Panaderia Hermanos Brenes será su tipo de usuario	

Tabla 4 Historia de usuario Registro de categorías

Historia de Usuario	
Numero 3	Usuario: Administrador, Especial
Nombre de la historia: Registro de categorías	
Prioridad: Alta (Alta, media, baja)	Riesgo en desarrollo: baja (Alto, medio, baja)
Iteración asignada 1	
Descripción: Los usuarios con los permisos definidos por el administrador tendrán el acceso a la pestaña donde se ingresan las categorías de pan existentes en la panadera, posteriormente se mantienen un registro de estas y se le asignan productos.	
Observaciones: Únicamente el administrador y el usuario especial pueden realizar esta acción	

Tabla 5 Historia de usuario de inventario de productos existentes

Historia de Usuario	
Numero 4	Usuario: Administrador, Especial
Nombre de la historia: Inventario de productos existentes	
Prioridad: Alta (Alta, media, baja)	Riesgo en desarrollo: alta (Alto, medio, baja)
Iteración asignada 2	
Descripción: Los usuarios con los permisos definidos por el administrador tendrán el acceso a la pestaña donde se ingresan los productos existentes en la Panaderia, a manera de inventario, a una categoría le corresponden una serie de productos	

Observaciones: Únicamente el administrador y el usuario especial pueden realizar esta acción

Tabla 6 Historia de usuario registro cliente

Historia de Usuario	
Numero 5	Usuario: Administrador, Vendedor
Nombre de la historia: Registro de clientes	
Prioridad: Alta (Alta, media, baja)	Riesgo en desarrollo: media (Alto, medio, baja)
Iteración asignada 2	
Descripción: Los usuarios con los permisos definidos por el administrador tendrán el acceso a la pestaña donde se registra la información básica de los clientes, tendrán acceso a la visualización de esta información y a la gestión de esta	
Observaciones: Únicamente el administrador y el vendedor pueden realizar esta acción	

Tabla 7 Historia de usuario gestión de ventas

Historia de Usuario	
Numero 6	Usuario: Administrador, Vendedor
Nombre de la historia: Gestión de ventas	
Prioridad: Alta (Alta, media, baja)	Riesgo en desarrollo: Alta(Alto, medio, baja)
Iteración asignada 2	
Descripción: Los usuarios con los permisos definidos por el administrador tendrán el acceso a la pestaña donde se registra cada venta, también tendrá acceso a la visualización, reporte y gestión de esta información	
Observaciones: Únicamente el administrador y el vendedor pueden realizar esta acción	

Tabla 8 Historia de usuario administración de ventas

Historia de Usuario	
Numero 7	Usuario: Administrador, Vendedor
Nombre de la historia: Administración de ventas	
Prioridad: Alta (Alta, media, baja)	Riesgo en desarrollo: alta (Alto, medio, baja)
Iteración asignada 3	
Descripción: Los usuarios con los permisos definidos por el administrador tendrán el acceso a la pestaña donde se visualiza y gestiona toda la información de los registros de ventas realizados	
Observaciones: Únicamente el administrador y el vendedor pueden realizar esta acción	

Tabla 9 Historia de usuario generación de reportes

Historia de Usuario	
Numero 8	Usuario: Administrador
Nombre de la historia: Generación de reportes	
Prioridad: Alta (Alta, media, baja)	Riesgo en desarrollo: alta (Alto, medio, baja)
Iteración asignada 3	
Descripción: El administrador obtiene un reporte en Excel o pdf de los registros de ventas por un rango de fecha, también puede gestionar la información de los registros de ventas y elegir como visualizar los registros	
Observaciones: Únicamente el administrador puede realizar esta acción	

7.2 Plan de entrega del proyecto

En base a las historias de usuario definidas anteriormente se elaboró el siguiente plan de entrega, el cual contiene las historias de usuario que se entregarán en cada iteración, teniendo en cuenta la prioridad y el orden cronológico de cada historia.

Numero de Historia de usuario	Iteración correspondiente	Prioridad	Fecha de Inicio	Fecha Final
1	1	Media	30/07/2021	15/08/2021
2	1	Alta	18/08/2021	04/09/2021
3	1	Baja	07/09/2021	19/09/2021
4	2	Alta	25/09/2021	15/10/2021
5	2	Media	17/10/2021	30/10/2021
6	2	Alta	02/11/2021	23/11/2021
7	3	Alta	25/11/2021	26/11/2021
8	3	Alta	06/01/2022	15/01/2022

Tabla 10 Plan de entrega de proyecto

7.3 Primera Iteración

A la iteración número 1 del plan de entrega de proyecto le corresponden las siguientes historias de usuario:

Tabla 11 Plan de entrega de proyecto primera iteración

Número	Nombre
1	Acceso al sistema
2	Registro de usuarios
3	Registro de categorías

Antes de comenzar el desarrollo de la primera iteración se realizó el diseño de base de datos, posteriormente se definen las tareas de ingeniería correspondientes a la primera historia de usuario, la cual tiene por nombre Acceso al sistema.

7.3.1 Diseño de base de datos

El diagrama # 3, diagrama de base de datos, ilustra el modelo de base de datos utilizado para almacenar la información del sistema web, simultáneamente se representan de forma gráfica las tablas contenidas en la base de datos, la estructura de estas y las relaciones entre sí.

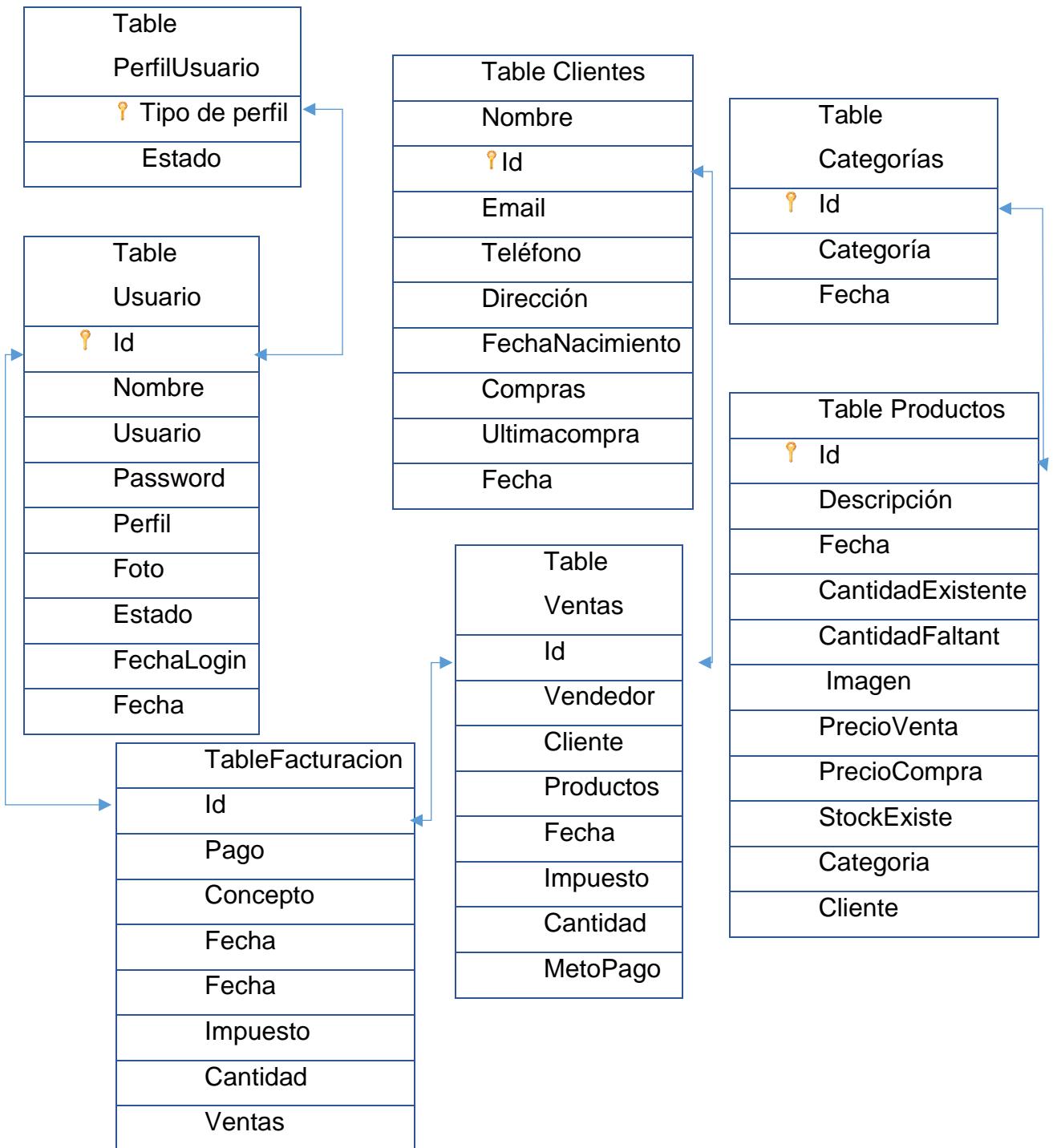


Ilustración 3 Diagrama de bases de datos

7.3.2 Tareas de ingeniería 1ra iteración

Las tareas de ingeniería son actividades que identifican, definen y controlan los requisitos que se entregarán en el sistema. Una vez diseñada la base de datos se detallan todas las tareas de ingeniería correspondiente a la primera iteración.

Tabla 12 Tareas de Ingeniería primera iteración

Numero de tarea	Numero de historias	Nombre de la tarea
1	1	Diseño de interfaz del Login
2	2	Almacenamiento de usuarios en BD
3	2	Definición de roles de los usuarios
4	2	Validación de tipos de usuarios
5	2	Diseño de interfaz para el administrador
6	3	Diseño de interfaz para el usuario especial
7	3	Diseño para el registro de la categoría

8	3	Diseño para la gestión de los registros de categoría
9	3	Validación para la gestión de la pestaña categoría
10	3	Validación para la visualización de la pestaña categoría
11	3	Validación de la visualización de los registros

Tabla 13 Tarea de Ingeniería 1 Iteración 1 Diseño de login

Tarea de Ingeniería	
Numero de tarea: 1	Numero de historia: 1
Nombre de la tarea: Diseño de la interfaz del login	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 05/08/2021	Fecha fin: 12/08/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se realiza el diseño de la interfaz de usuario, un login, en donde el usuario ingresará sus credenciales para acceder al sistema de información, también se realiza un diseño de interfaz de usuario para visualizar las credenciales que el administrador posteriormente le otorgará a los usuarios.	

Tabla 14 Tarea de Ingeniería 2 Iteración 1 Almacenamiento usuarios

Tarea de Ingeniería	
Numero de tarea: 2	Numero de historia: 1
Nombre de la tarea: Almacenamiento de los usuarios en Base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 05/08/2021	Fecha fin: 06/08/2021
Programador: Christopher Daniel Paniagua Martinez	
Descripción: Se conecta la base de datos con la interfaz del login para capturar los datos verificar el acceso y gestionar los usuarios en la interfaz de usuario	

Tabla 15 Tarea de Ingeniería 3 Iteración 1 Roles de usuario

Tarea de Ingeniería	
Numero de tarea: 3	Numero de historia: 1
Nombre de la tarea: Definición de los roles de los usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 06/08/2021	Fecha fin: 08/08/2021
Programador: Christopher Daniel Paniagua Martinez	
Descripción: Se define los 3 diferentes tipos de usuarios y en dependencia de sus funciones sus permisos, también se le otorga el control total de los usuarios al administrador	

Tabla 16 Tarea de Ingeniería 4 Iteración 1 Validación de usuarios

Tarea de Ingeniería	
Numero de tarea: 4	Numero de historia: 1
Nombre de la tarea: Validación de los tipos de usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 08/08/2021	Fecha fin: 08/08/2021
Programador: Christopher Daniel Paniagua Martinez	
Descripción: Se valida que las claves y contraseñas de los usuarios funcionen en conjunto con la interfaz y la base de datos correctamente	

Tabla 17 Tarea de Ingeniería 5 Iteración 1 Diseño interfaz admin

Tarea de Ingeniería	
Numero de tarea: 5	Numero de historia: 1
Nombre de la tarea: Diseño de interfaz para el administrador	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 09/08/2021	Fecha fin: 12/08/2021
Programador: Sofía Alejandra Miranda Valle	

Descripción: Se codifica un diseño de interfaz de lo que será el sistema, en este punto la interfaz de usuario debe de ser funcional y los otros módulos deben estar en un menú desplazable a manera de lectura.

Tabla 18 Tarea de Ingeniería 6 Iteración 1 Diseño de interfaz usuario E

Tarea de Ingeniería	
Numero de tarea: 6	Numero de historia: 1
Nombre de la tarea: Diseño de interfaz para usuario especial	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 12/08/2021	Fecha fin: 13/08/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se codifica un diseño de interfaz de lo que será el sistema orientado al usuario de tipo especial.	

Tabla 19 Tarea de Ingeniería 7 Iteración 1 Diseño categoría

Tarea de Ingeniería	
Numero de tarea: 7	Numero de historia: 1
Nombre de la tarea: Diseño para el registro de interfaz categoría	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 15/08/2021	Fecha fin: 18/08/2021
Programador: Christopher Daniel Paniagua Martinez	
Descripción: Se realiza un diseño de interfaz funcional en relación a la estructura de la tabla categoría previamente codificada en base de datos.	

Tabla 20 Tarea de Ingeniería 8 Iteración 1 Diseño gestión categoría

Tarea de Ingeniería	
Numero de tarea: 8	Numero de historia: 1
Nombre de la tarea: Diseño para la gestión de los registros de categoría	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 18/08/2021	Fecha fin: 23/08/2021
Programador: Christopher Daniel Paniagua Martinez	
Descripción: Se realiza un diseño de interfaz para la gestión de los procesos que estarán presentes en el sistema de información dentro del módulo categoría.	

Tabla 21 Tarea de Ingeniería 9 Iteración 1 Validación gestión categoría

Tarea de Ingeniería	
Numero de tarea: 9	Numero de historia: 1
Nombre de la tarea: Validación para la gestión de la pestaña categoría	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 24/08/2021	Fecha fin: 30/08/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se codifican , automatizan y validan los procesos de gestión que están presentes en el módulo de categoría.	

Tabla 22 Tarea de Ingeniería 10 Iteración 1 Validación pestaña categoría

Tarea de Ingeniería	
Numero de tarea: 10	Numero de historia: 1
Nombre de la tarea: Validación para la visualización de la pestaña categoría	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 30/08/2021	Fecha fin: 10/09/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se codifican los procesos y definen la información pertinente a ser visualizada en la pestaña principal del módulo de categoría	

Tabla 23 Tarea de Ingeniería 11 Iteración 1 Visualización de los registros

Tarea de Ingeniería	
Numero de tarea: 11	Numero de historia: 1
Nombre de la tarea: Validación de la visualización de los registros	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 10/09/2021	Fecha fin: 19/09/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: En base a la preferencia del cliente se realiza una validación	

7.3.3 Tarjetas CRC

Las tarjetas CRC (clase, responsabilidad, colaboración) son una herramienta visual que ilustra una serie de actividades a cumplirse, así como sus correspondientes colaboradores o responsables que participan en determinada actividad, a continuación, se muestran las tarjetas de Clase-Responsabilidad-Colaboración de cada tarea de Ingeniería, correspondiente a cada iteración.

El modelo de la plantilla que se utilizó para la elaboración de las tarjetas CRC y sus componentes es la siguiente:

Tabla 24 Plantilla de tarjetas CRC

Clase: Nombre de la tarea correspondiente	
Responsabilidad: Asignación del desarrollo de la clase	Nombre de la persona encargada
Colaboradores: usuarios que manipularan el sistema	Tipos de usuarios
Tarea: Actividades a desarrollar	

A continuación, en las tablas 25-30 se muestran las tarjetas CRC que fueron empleadas para llevar a cabo el desarrollo del sistema informático.

Tabla 25 Tarjeta CRC Tarea de Ingeniería 1,2,3 Login

Clase: Diseño de interfaz del Login	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, usuario especial, vendedor
Diseñar la interfaz del login	
Validar la interfaz del login	
Verificar las credenciales de los usuarios	
Designar los roles de usuarios	
Validar los roles de usuarios	
Autorizar el acceso al sistema	
Devolver un mensaje en caso que el usuario no exista	

Tabla 26 Tarjeta CRC Tarea de Ingeniería 4 Usuarios en BD

Clase: Almacenamiento de usuarios en Base de Datos	
Responsabilidad	Christopher Paniagua
Colaboración	Clientes, administrador
Realizar la conexión del login a BD	
Definir los tipos de usuario	
Conceder el acceso al sistema en dependencia del tipo de usuario	
Validar los usuarios en BD	

Tabla 27 Tarjeta CRC Tarea de Ingeniería 6 Diseño interfaz admin

Clase: Diseño Interfaz Admin	
Responsabilidad	Sofie Valle
Colaboración	Clientes, administrador
Diseñar una interfaz en base a los requerimientos del usuario	
Diseñar una interfaz de usuario con interacción flexible	
Diseñar procesos que sean reversibles	
Diseñar un menú de tipo dropdown	
Realizar un diseño consistente e intuitivo	

Tabla 28 Tarjeta CRC Tarea de Ingeniería 6 Diseño interfaz usuario

Clase: Diseño de Interfaz de usuario especial	
Responsabilidad	Sofie Valle
Colaboración	Clientes, administrador
Diseñar una interfaz que reduzca la necesidad que el usuario memorice	
Ocultar los tecnicismos internos al usuario ocasional	
Diseñar procesos que sean reversibles	
Diseñar formularios consistentes	

Diseñar una interfaz en donde permita tener el control al usuario	
Mostrar menú de operaciones	
Llamar a los gestores de menú correspondiente para cada proceso	

Tabla 29 Tarjeta CRC Tarea de Ingeniería 7,8 Diseño interfaz categoría

Clase: Diseño de Interfaz Categoría	
Responsabilidad	Christopher Paniagua
Colaboración	Clientes, administrador
Diseñar una interfaz que permita que la interacción del usuario sea intuitiva	
Diseñar procesos internos personalizables	
Diseñar una interfaz que proporcione interacción directa con los objetos en pantalla	
Definir atajos que sean intuitivos	
Diseñar una interfaz en donde permita tener el control al usuario	
Mostrar menú de operaciones	
Llamar a los gestores de menú correspondiente para cada proceso	
Revelar información de forma progresiva	

Tabla 30 Tarea CRC Tarea de Ingeniería 9,10,11 Validación categoría

Clase: Validación Categoría	
Responsabilidad	Christopher Paniagua, Sofie Valle
Colaboración	Clientes, administrador
Garantizar la captura de los datos	
Controlar la información a medida que avanza cada proceso	

Codificar procesos trazables	
Codificar atajos que sean intuitivos	
Agilizar el flujo de los procesos evitando retrasos y mitigando errores	
Codificar menú de operaciones	
Codificar a los gestores de formularios de menú correspondiente para cada proceso	
Codificar procesos automatizados	

7.3.4 Codificación

En esta sección se presentan los resultados de la etapa de escritura de la solución propuesta del sistema informático, en otras palabras, el código fuente de las instrucciones definidas en el lenguaje de programación.

```

1  <?php
2
3  <?php
4
5  <?php
6
7  <?php
8
9  <?php
10 <?php
11 <?php
12 <?php
13 <?php
14 <?php
15 <?php
16 <?php
17 <?php

```

Ilustración 4 Codificación conexión de base de datos

```

2
3 class ControladorUsuarios{
4
5     static public function ctrIngresoUsuario(){
6
7         if(isset($_POST["ingUsuario"])){
8
9             if(preg_match('/^[_a-zA-Z0-9]+$/i', $_POST["ingUsuario"])){
10
11                 $encriptar = crypt($_POST["ingPassword"], '$2a$07$asxx54ahjppf45sd87a5a4dDDGsystemdev$');
12
13                 $tabla = "usuarios";
14
15                 $item = "usuario";
16                 $valor = $_POST["ingUsuario"];
17
18                 $respuesta = ModeloUsuarios::MdlMostrarUsuarios($tabla, $item, $valor);
19
20                 if($respuesta["usuario"] == $_POST["ingUsuario"] && $respuesta["password"] == $encriptar){
21
22                     if($respuesta["estado"] == 1){
23
24                         $_SESSION["iniciarSesion"] = "ok";
25                         $_SESSION["id"] = $respuesta["id"];
26                         $_SESSION["nombre"] = $respuesta["nombre"];
27                         $_SESSION["usuario"] = $respuesta["usuario"];
28                         $_SESSION["foto"] = $respuesta["foto"];
29                         $_SESSION["perfil"] = $respuesta["perfil"];

```

Ilustración 5 Codificación ingreso de usuarios

```

82
83     static public function ctrCrearUsuario(){
84
85         if(isset($_POST["nuevoUsuario"])){
86
87             if(preg_match('/^[_a-zA-Z0-9ñÑéíóúÁÉÍÓÚ]+$/i', $_POST["nuevoNombre"]) &&
88                 preg_match('/^[_a-zA-Z0-9]+$/i', $_POST["nuevoUsuario"]) &&
89                 preg_match('/^[_a-zA-Z0-9]+$/i', $_POST["nuevoPassword"])){
90
91                 $ruta = "";
92
93                 if(isset($_FILES["nuevaFoto"]["tmp_name"])){
94
95                     list($ancho, $alto) = getimagesize($_FILES["nuevaFoto"]["tmp_name"]);
96
97                     $nuevoAncho = 500;
98                     $nuevoAlto = 500;
99                     $directorio = "vistas/img/usuarios/".$_POST["nuevoUsuario"];
100                     mkdir($directorio, 0755);
101                     if($_FILES["nuevaFoto"]["type"] == "image/jpeg"){
102
103                         $aleatorio = mt_rand(100,999);
104
105                         $ruta = "vistas/img/usuarios/".$_POST["nuevoUsuario"]."/".$aleatorio.".jpg";
106
107                         $origen = imagecreatefromjpeg($_FILES["nuevaFoto"]["tmp_name"]);
108
109                         $destino = imagecreatetruecolor($nuevoAncho, $nuevoAlto);
110
111                         imagecopyresized($destino, $origen, 0, 0, 0, 0, $nuevoAncho, $nuevoAlto, $ancho, $alto);
112
113                         imagejpeg($destino, $ruta);

```

Ilustración 6 Codificación creación de usuarios

```
modelos > categorias.modelo.php
1  <?php
2
3  require_once "conexion.php";
4
5  class ModeloCategorias{
6
7
8
9      static public function mdlIngresarCategoria($tabla, $datos){
10
11         $stmt = Conexion::conectar()->prepare("INSERT INTO $tabla(categoría) VALUES (:categoria)");
12
13         $stmt->bindParam(":categoria", $datos, PDO::PARAM_STR);
14
15         if($stmt->execute()){
16
17             return "ok";
18
19         }else{
20
21             return "error";
22
23         }
24
25         $stmt->close();
26         $stmt = null;
27
28     }
29
30
31     static public function mdlMostrarCategorias($tabla, $item, $valor){
32
33         if($item != null){
34
35             $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
36
37             $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
38
39             $stmt->execute();
40
41             $data = $stmt->fetch();
42
43             return $data;
44
45         }else{
46
47             $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
48
49             $stmt->execute();
50
51             $data = $stmt->fetchAll();
52
53             return $data;
54
55         }
56
57     }
58
59
60     static public function mdlMostrarCategoriasId($tabla, $id){
61
62         $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE id = :id");
63
64         $stmt->bindParam(":id", $id, PDO::PARAM_INT);
65
66         $stmt->execute();
67
68         $data = $stmt->fetch();
69
70         return $data;
71
72     }
73
74
75     static public function mdlBorrarCategoria($tabla, $id){
76
77         $stmt = Conexion::conectar()->prepare("DELETE FROM $tabla WHERE id = :id");
78
79         $stmt->bindParam(":id", $id, PDO::PARAM_INT);
80
81         if($stmt->execute()){
82
83             return "ok";
84
85         }else{
86
87             return "error";
88
89         }
90
91     }
92
93
94     static public function mdlActualizarCategoria($tabla, $id, $datos){
95
96         $stmt = Conexion::conectar()->prepare("UPDATE $tabla SET $datos WHERE id = :id");
97
98         $stmt->bindParam(":id", $id, PDO::PARAM_INT);
99
100        $stmt->execute();
101
102        if($stmt->execute()){
103
104            return "ok";
105
106        }else{
107
108            return "error";
109
110        }
111
112    }
113
114
115    static public function mdlMostrarCategoriasId($tabla, $id){
116
117        $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE id = :id");
118
119        $stmt->bindParam(":id", $id, PDO::PARAM_INT);
120
121        $stmt->execute();
122
123        $data = $stmt->fetch();
124
125        return $data;
126
127    }
128
129
130    static public function mdlMostrarCategorias($tabla, $item, $valor){
131
132        if($item != null){
133
134            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
135
136            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
137
138            $stmt->execute();
139
140            $data = $stmt->fetchAll();
141
142            return $data;
143
144        }else{
145
146            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
147
148            $stmt->execute();
149
150            $data = $stmt->fetchAll();
151
152            return $data;
153
154        }
155
156    }
157
158
159    static public function mdlMostrarCategorias($tabla, $item, $valor){
160
161        if($item != null){
162
163            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
164
165            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
166
167            $stmt->execute();
168
169            $data = $stmt->fetchAll();
170
171            return $data;
172
173        }else{
174
175            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
176
177            $stmt->execute();
178
179            $data = $stmt->fetchAll();
180
181            return $data;
182
183        }
184
185    }
186
187
188    static public function mdlMostrarCategorias($tabla, $item, $valor){
189
190        if($item != null){
191
192            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
193
194            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
195
196            $stmt->execute();
197
198            $data = $stmt->fetchAll();
199
200            return $data;
201
202        }else{
203
204            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
205
206            $stmt->execute();
207
208            $data = $stmt->fetchAll();
209
210            return $data;
211
212        }
213
214    }
215
216
217    static public function mdlMostrarCategorias($tabla, $item, $valor){
218
219        if($item != null){
220
221            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
222
223            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
224
225            $stmt->execute();
226
227            $data = $stmt->fetchAll();
228
229            return $data;
230
231        }else{
232
233            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
234
235            $stmt->execute();
236
237            $data = $stmt->fetchAll();
238
239            return $data;
240
241        }
242
243    }
244
245
246    static public function mdlMostrarCategorias($tabla, $item, $valor){
247
248        if($item != null){
249
250            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
251
252            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
253
254            $stmt->execute();
255
256            $data = $stmt->fetchAll();
257
258            return $data;
259
260        }else{
261
262            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
263
264            $stmt->execute();
265
266            $data = $stmt->fetchAll();
267
268            return $data;
269
270        }
271
272    }
273
274
275    static public function mdlMostrarCategorias($tabla, $item, $valor){
276
277        if($item != null){
278
279            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
280
281            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
282
283            $stmt->execute();
284
285            $data = $stmt->fetchAll();
286
287            return $data;
288
289        }else{
290
291            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
292
293            $stmt->execute();
294
295            $data = $stmt->fetchAll();
296
297            return $data;
298
299        }
300
301    }
302
303
304    static public function mdlMostrarCategorias($tabla, $item, $valor){
305
306        if($item != null){
307
308            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
309
310            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
311
312            $stmt->execute();
313
314            $data = $stmt->fetchAll();
315
316            return $data;
317
318        }else{
319
320            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
321
322            $stmt->execute();
323
324            $data = $stmt->fetchAll();
325
326            return $data;
327
328        }
329
330    }
331
332
333    static public function mdlMostrarCategorias($tabla, $item, $valor){
334
335        if($item != null){
336
337            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
338
339            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
340
341            $stmt->execute();
342
343            $data = $stmt->fetchAll();
344
345            return $data;
346
347        }else{
348
349            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
350
351            $stmt->execute();
352
353            $data = $stmt->fetchAll();
354
355            return $data;
356
357        }
358
359    }
360
361
362    static public function mdlMostrarCategorias($tabla, $item, $valor){
363
364        if($item != null){
365
366            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
367
368            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
369
370            $stmt->execute();
371
372            $data = $stmt->fetchAll();
373
374            return $data;
375
376        }else{
377
378            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
379
380            $stmt->execute();
381
382            $data = $stmt->fetchAll();
383
384            return $data;
385
386        }
387
388    }
389
390
391    static public function mdlMostrarCategorias($tabla, $item, $valor){
392
393        if($item != null){
394
395            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
396
397            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
398
399            $stmt->execute();
400
401            $data = $stmt->fetchAll();
402
403            return $data;
404
405        }else{
406
407            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
408
409            $stmt->execute();
410
411            $data = $stmt->fetchAll();
412
413            return $data;
414
415        }
416
417    }
418
419
420    static public function mdlMostrarCategorias($tabla, $item, $valor){
421
422        if($item != null){
423
424            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
425
426            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
427
428            $stmt->execute();
429
430            $data = $stmt->fetchAll();
431
432            return $data;
433
434        }else{
435
436            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
437
438            $stmt->execute();
439
440            $data = $stmt->fetchAll();
441
442            return $data;
443
444        }
445
446    }
447
448
449    static public function mdlMostrarCategorias($tabla, $item, $valor){
450
451        if($item != null){
452
453            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
454
455            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
456
457            $stmt->execute();
458
459            $data = $stmt->fetchAll();
460
461            return $data;
462
463        }else{
464
465            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
466
467            $stmt->execute();
468
469            $data = $stmt->fetchAll();
470
471            return $data;
472
473        }
474
475    }
476
477
478    static public function mdlMostrarCategorias($tabla, $item, $valor){
479
480        if($item != null){
481
482            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
483
484            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
485
486            $stmt->execute();
487
488            $data = $stmt->fetchAll();
489
490            return $data;
491
492        }else{
493
494            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
495
496            $stmt->execute();
497
498            $data = $stmt->fetchAll();
499
500            return $data;
501
502        }
503
504    }
505
506
507    static public function mdlMostrarCategorias($tabla, $item, $valor){
508
509        if($item != null){
510
511            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
512
513            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
514
515            $stmt->execute();
516
517            $data = $stmt->fetchAll();
518
519            return $data;
520
521        }else{
522
523            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
524
525            $stmt->execute();
526
527            $data = $stmt->fetchAll();
528
529            return $data;
530
531        }
532
533    }
534
535
536    static public function mdlMostrarCategorias($tabla, $item, $valor){
537
538        if($item != null){
539
540            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
541
542            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
543
544            $stmt->execute();
545
546            $data = $stmt->fetchAll();
547
548            return $data;
549
550        }else{
551
552            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
553
554            $stmt->execute();
555
556            $data = $stmt->fetchAll();
557
558            return $data;
559
560        }
561
562    }
563
564
565    static public function mdlMostrarCategorias($tabla, $item, $valor){
566
567        if($item != null){
568
569            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
570
571            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
572
573            $stmt->execute();
574
575            $data = $stmt->fetchAll();
576
577            return $data;
578
579        }else{
580
581            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
582
583            $stmt->execute();
584
585            $data = $stmt->fetchAll();
586
587            return $data;
588
589        }
590
591    }
592
593
594    static public function mdlMostrarCategorias($tabla, $item, $valor){
595
596        if($item != null){
597
598            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
599
600            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
601
602            $stmt->execute();
603
604            $data = $stmt->fetchAll();
605
606            return $data;
607
608        }else{
609
610            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
611
612            $stmt->execute();
613
614            $data = $stmt->fetchAll();
615
616            return $data;
617
618        }
619
620    }
621
622
623    static public function mdlMostrarCategorias($tabla, $item, $valor){
624
625        if($item != null){
626
627            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
628
629            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
630
631            $stmt->execute();
632
633            $data = $stmt->fetchAll();
634
635            return $data;
636
637        }else{
638
639            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
640
641            $stmt->execute();
642
643            $data = $stmt->fetchAll();
644
645            return $data;
646
647        }
648
649    }
650
651
652    static public function mdlMostrarCategorias($tabla, $item, $valor){
653
654        if($item != null){
655
656            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
657
658            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
659
660            $stmt->execute();
661
662            $data = $stmt->fetchAll();
663
664            return $data;
665
666        }else{
667
668            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
669
670            $stmt->execute();
671
672            $data = $stmt->fetchAll();
673
674            return $data;
675
676        }
677
678    }
679
680
681    static public function mdlMostrarCategorias($tabla, $item, $valor){
682
683        if($item != null){
684
685            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
686
687            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
688
689            $stmt->execute();
690
691            $data = $stmt->fetchAll();
692
693            return $data;
694
695        }else{
696
697            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
698
699            $stmt->execute();
700
701            $data = $stmt->fetchAll();
702
703            return $data;
704
705        }
706
707    }
708
709
710    static public function mdlMostrarCategorias($tabla, $item, $valor){
711
712        if($item != null){
713
714            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
715
716            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
717
718            $stmt->execute();
719
720            $data = $stmt->fetchAll();
721
722            return $data;
723
724        }else{
725
726            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
727
728            $stmt->execute();
729
730            $data = $stmt->fetchAll();
731
732            return $data;
733
734        }
735
736    }
737
738
739    static public function mdlMostrarCategorias($tabla, $item, $valor){
740
741        if($item != null){
742
743            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
744
745            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
746
747            $stmt->execute();
748
749            $data = $stmt->fetchAll();
750
751            return $data;
752
753        }else{
754
755            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
756
757            $stmt->execute();
758
759            $data = $stmt->fetchAll();
760
761            return $data;
762
763        }
764
765    }
766
767
768    static public function mdlMostrarCategorias($tabla, $item, $valor){
769
770        if($item != null){
771
772            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
773
774            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
775
776            $stmt->execute();
777
778            $data = $stmt->fetchAll();
779
780            return $data;
781
782        }else{
783
784            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
785
786            $stmt->execute();
787
788            $data = $stmt->fetchAll();
789
790            return $data;
791
792        }
793
794    }
795
796
797    static public function mdlMostrarCategorias($tabla, $item, $valor){
798
799        if($item != null){
800
801            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
802
803            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
804
805            $stmt->execute();
806
807            $data = $stmt->fetchAll();
808
809            return $data;
810
811        }else{
812
813            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
814
815            $stmt->execute();
816
817            $data = $stmt->fetchAll();
818
819            return $data;
820
821        }
822
823    }
824
825
826    static public function mdlMostrarCategorias($tabla, $item, $valor){
827
828        if($item != null){
829
830            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
831
832            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
833
834            $stmt->execute();
835
836            $data = $stmt->fetchAll();
837
838            return $data;
839
840        }else{
841
842            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
843
844            $stmt->execute();
845
846            $data = $stmt->fetchAll();
847
848            return $data;
849
850        }
851
852    }
853
854
855    static public function mdlMostrarCategorias($tabla, $item, $valor){
856
857        if($item != null){
858
859            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
860
861            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
862
863            $stmt->execute();
864
865            $data = $stmt->fetchAll();
866
867            return $data;
868
869        }else{
870
871            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
872
873            $stmt->execute();
874
875            $data = $stmt->fetchAll();
876
877            return $data;
878
879        }
880
881    }
882
883
884    static public function mdlMostrarCategorias($tabla, $item, $valor){
885
886        if($item != null){
887
888            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
889
890            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
891
892            $stmt->execute();
893
894            $data = $stmt->fetchAll();
895
896            return $data;
897
898        }else{
899
900            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
901
902            $stmt->execute();
903
904            $data = $stmt->fetchAll();
905
906            return $data;
907
908        }
909
910    }
911
912
913    static public function mdlMostrarCategorias($tabla, $item, $valor){
914
915        if($item != null){
916
917            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
918
919            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
920
921            $stmt->execute();
922
923            $data = $stmt->fetchAll();
924
925            return $data;
926
927        }else{
928
929            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
930
931            $stmt->execute();
932
933            $data = $stmt->fetchAll();
934
935            return $data;
936
937        }
938
939    }
940
941
942    static public function mdlMostrarCategorias($tabla, $item, $valor){
943
944        if($item != null){
945
946            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
947
948            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
949
950            $stmt->execute();
951
952            $data = $stmt->fetchAll();
953
954            return $data;
955
956        }else{
957
958            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
959
960            $stmt->execute();
961
962            $data = $stmt->fetchAll();
963
964            return $data;
965
966        }
967
968    }
969
970
971    static public function mdlMostrarCategorias($tabla, $item, $valor){
972
973        if($item != null){
974
975            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
976
977            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
978
979            $stmt->execute();
980
981            $data = $stmt->fetchAll();
982
983            return $data;
984
985        }else{
986
987            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
988
989            $stmt->execute();
990
991            $data = $stmt->fetchAll();
992
993            return $data;
994
995        }
996
997    }
998
999
1000   static public function mdlMostrarCategorias($tabla, $item, $valor){
1001
1002        if($item != null){
1003
1004            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1005
1006            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1007
1008            $stmt->execute();
1009
1010            $data = $stmt->fetchAll();
1011
1012            return $data;
1013
1014        }else{
1015
1016            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1017
1018            $stmt->execute();
1019
1020            $data = $stmt->fetchAll();
1021
1022            return $data;
1023
1024        }
1025
1026    }
1027
1028
1029    static public function mdlMostrarCategorias($tabla, $item, $valor){
1030
1031        if($item != null){
1032
1033            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1034
1035            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1036
1037            $stmt->execute();
1038
1039            $data = $stmt->fetchAll();
1040
1041            return $data;
1042
1043        }else{
1044
1045            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1046
1047            $stmt->execute();
1048
1049            $data = $stmt->fetchAll();
1050
1051            return $data;
1052
1053        }
1054
1055    }
1056
1057
1058    static public function mdlMostrarCategorias($tabla, $item, $valor){
1059
1060        if($item != null){
1061
1062            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1063
1064            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1065
1066            $stmt->execute();
1067
1068            $data = $stmt->fetchAll();
1069
1070            return $data;
1071
1072        }else{
1073
1074            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1075
1076            $stmt->execute();
1077
1078            $data = $stmt->fetchAll();
1079
1080            return $data;
1081
1082        }
1083
1084    }
1085
1086
1087    static public function mdlMostrarCategorias($tabla, $item, $valor){
1088
1089        if($item != null){
1090
1091            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1092
1093            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1094
1095            $stmt->execute();
1096
1097            $data = $stmt->fetchAll();
1098
1099            return $data;
1100
1101        }else{
1102
1103            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1104
1105            $stmt->execute();
1106
1107            $data = $stmt->fetchAll();
1108
1109            return $data;
1110
1111        }
1112
1113    }
1114
1115
1116    static public function mdlMostrarCategorias($tabla, $item, $valor){
1117
1118        if($item != null){
1119
1120            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1121
1122            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1123
1124            $stmt->execute();
1125
1126            $data = $stmt->fetchAll();
1127
1128            return $data;
1129
1130        }else{
1131
1132            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1133
1134            $stmt->execute();
1135
1136            $data = $stmt->fetchAll();
1137
1138            return $data;
1139
1140        }
1141
1142    }
1143
1144
1145    static public function mdlMostrarCategorias($tabla, $item, $valor){
1146
1147        if($item != null){
1148
1149            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1150
1151            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1152
1153            $stmt->execute();
1154
1155            $data = $stmt->fetchAll();
1156
1157            return $data;
1158
1159        }else{
1160
1161            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1162
1163            $stmt->execute();
1164
1165            $data = $stmt->fetchAll();
1166
1167            return $data;
1168
1169        }
1170
1171    }
1172
1173
1174    static public function mdlMostrarCategorias($tabla, $item, $valor){
1175
1176        if($item != null){
1177
1178            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1179
1180            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1181
1182            $stmt->execute();
1183
1184            $data = $stmt->fetchAll();
1185
1186            return $data;
1187
1188        }else{
1189
1190            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1191
1192            $stmt->execute();
1193
1194            $data = $stmt->fetchAll();
1195
1196            return $data;
1197
1198        }
1199
1200    }
1201
1202
1203    static public function mdlMostrarCategorias($tabla, $item, $valor){
1204
1205        if($item != null){
1206
1207            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1208
1209            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1210
1211            $stmt->execute();
1212
1213            $data = $stmt->fetchAll();
1214
1215            return $data;
1216
1217        }else{
1218
1219            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1220
1221            $stmt->execute();
1222
1223            $data = $stmt->fetchAll();
1224
1225            return $data;
1226
1227        }
1228
1229    }
1230
1231
1232    static public function mdlMostrarCategorias($tabla, $item, $valor){
1233
1234        if($item != null){
1235
1236            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1237
1238            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1239
1240            $stmt->execute();
1241
1242            $data = $stmt->fetchAll();
1243
1244            return $data;
1245
1246        }else{
1247
1248            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1249
1250            $stmt->execute();
1251
1252            $data = $stmt->fetchAll();
1253
1254            return $data;
1255
1256        }
1257
1258    }
1259
1260
1261    static public function mdlMostrarCategorias($tabla, $item, $valor){
1262
1263        if($item != null){
1264
1265            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1266
1267            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1268
1269            $stmt->execute();
1270
1271            $data = $stmt->fetchAll();
1272
1273            return $data;
1274
1275        }else{
1276
1277            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1278
1279            $stmt->execute();
1280
1281            $data = $stmt->fetchAll();
1282
1283            return $data;
1284
1285        }
1286
1287    }
1288
1289
1290    static public function mdlMostrarCategorias($tabla, $item, $valor){
1291
1292        if($item != null){
1293
1294            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1295
1296            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1297
1298            $stmt->execute();
1299
1300            $data = $stmt->fetchAll();
1301
1302            return $data;
1303
1304        }else{
1305
1306            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1307
1308            $stmt->execute();
1309
1310            $data = $stmt->fetchAll();
1311
1312            return $data;
1313
1314        }
1315
1316    }
1317
1318
1319    static public function mdlMostrarCategorias($tabla, $item, $valor){
1320
1321        if($item != null){
1322
1323            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1324
1325            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1326
1327            $stmt->execute();
1328
1329            $data = $stmt->fetchAll();
1330
1331            return $data;
1332
1333        }else{
1334
1335            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1336
1337            $stmt->execute();
1338
1339            $data = $stmt->fetchAll();
1340
1341            return $data;
1342
1343        }
1344
1345    }
1346
1347
1348    static public function mdlMostrarCategorias($tabla, $item, $valor){
1349
1350        if($item != null){
1351
1352            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1353
1354            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1355
1356            $stmt->execute();
1357
1358            $data = $stmt->fetchAll();
1359
1360            return $data;
1361
1362        }else{
1363
1364            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1365
1366            $stmt->execute();
1367
1368            $data = $stmt->fetchAll();
1369
1370            return $data;
1371
1372        }
1373
1374    }
1375
1376
1377    static public function mdlMostrarCategorias($tabla, $item, $valor){
1378
1379        if($item != null){
1380
1381            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$valor");
1382
1383            $stmt->bindParam(":$item", $valor, PDO::PARAM_STR);
1384
1385            $stmt->execute();
1386
1387            $data = $stmt->fetchAll();
1388
1389            return $data;
1390
1391        }else{
1392
1393            $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla");
1394
1395            $stmt->execute();
1396
1397            $data = $stmt->fetchAll();
1398
1399            return $data;
1400
1401        }
1402
1403    }
1404
1405
1406    static public function mdlMostrarCategorias($tabla, $item, $
```

Ilustración 5 Codificación ingresar categoría

```
static public function mdlIngresarUsuario($tabla, $datos){

$stmt = Conexion::conectar()->prepare("INSERT INTO $tabla(nombre, usuario, password, perfil, foto)

$stmt->bindParam(":nombre", $datos["nombre"], PDO::PARAM_STR);
$stmt->bindParam(":usuario", $datos["usuario"], PDO::PARAM_STR);
$stmt->bindParam(":password", $datos["password"], PDO::PARAM_STR);
$stmt->bindParam(":perfil", $datos["perfil"], PDO::PARAM_STR);
$stmt->bindParam(":foto", $datos["foto"], PDO::PARAM_STR);

if($stmt->execute()){

    return "ok";

} else{

    return "error";

}

$stmt->close();

$stmt = null;

}
```

Ilustración 7 Codificación registro de usuarios

7.3.5 Bocetos

A continuación, en las ilustraciones 9-15 se presentan los bocetos que se diseñaron con el cliente para llevar a cabo la creación del sistema



Ilustración 6 Boceto acceso al sistema

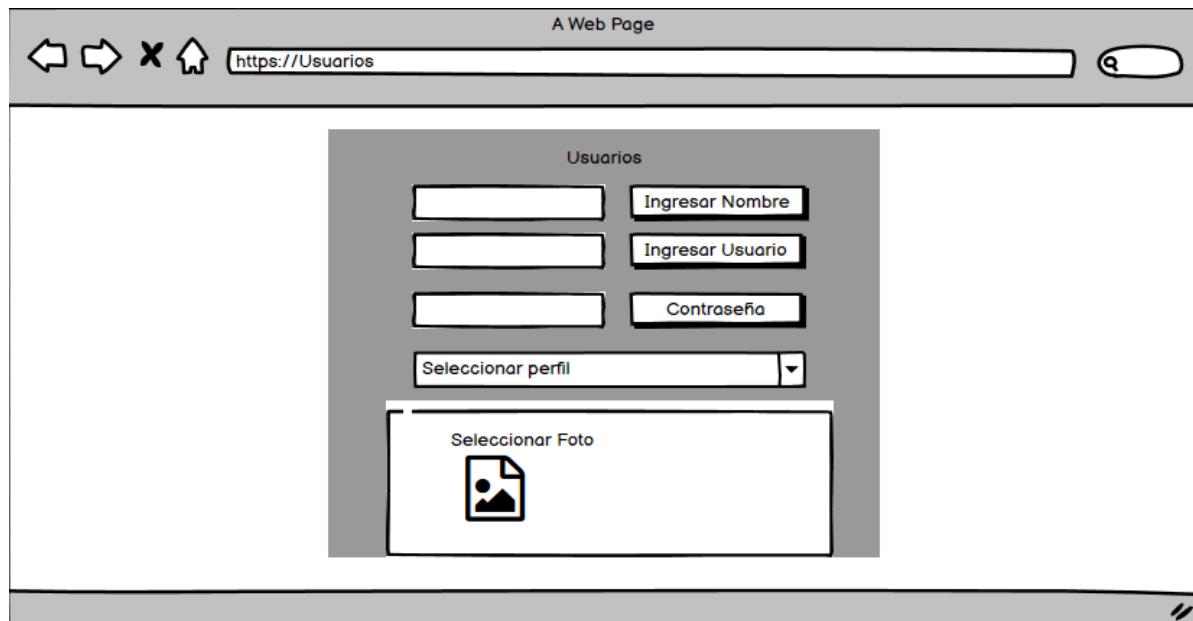


Ilustración 7 Boceto agregar usuario

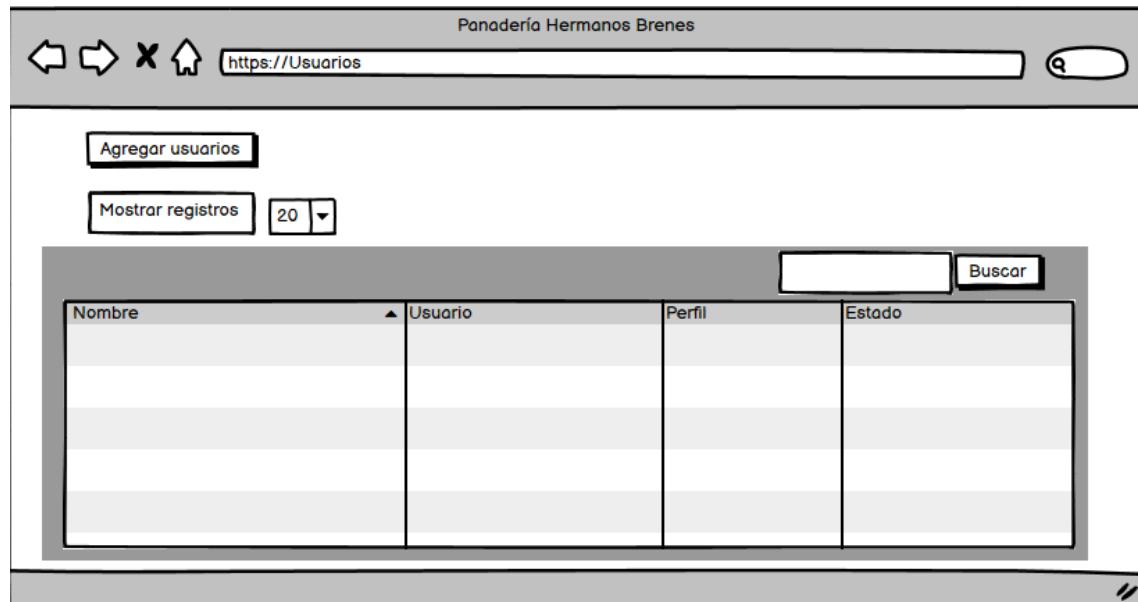


Ilustración 8 Boceto visualizar usuarios

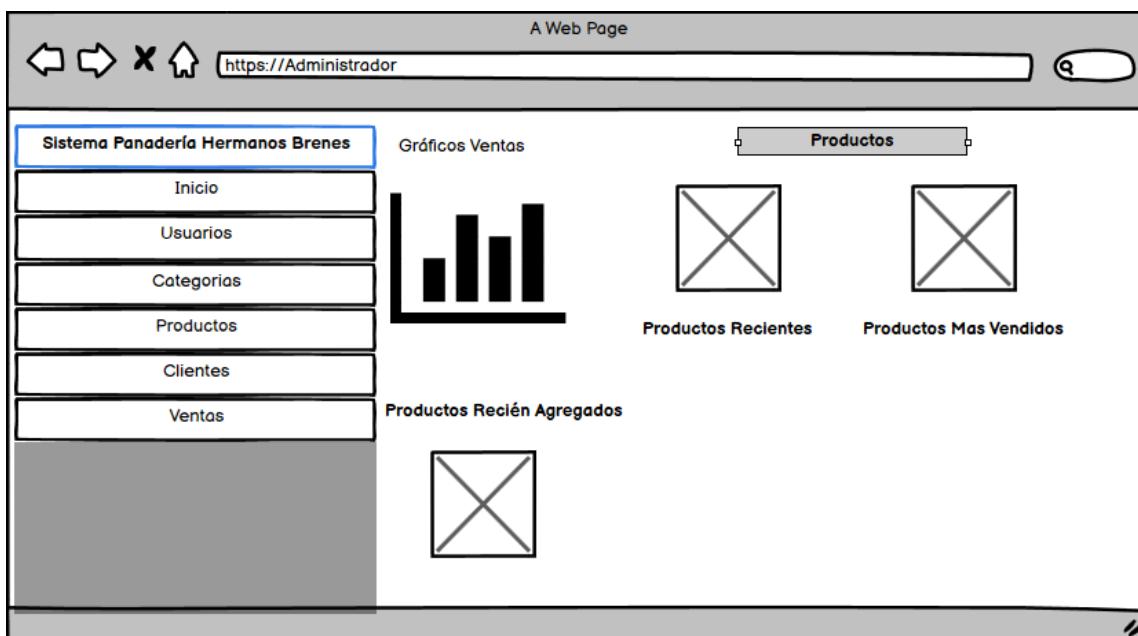


Ilustración 9 Boceto página inicio

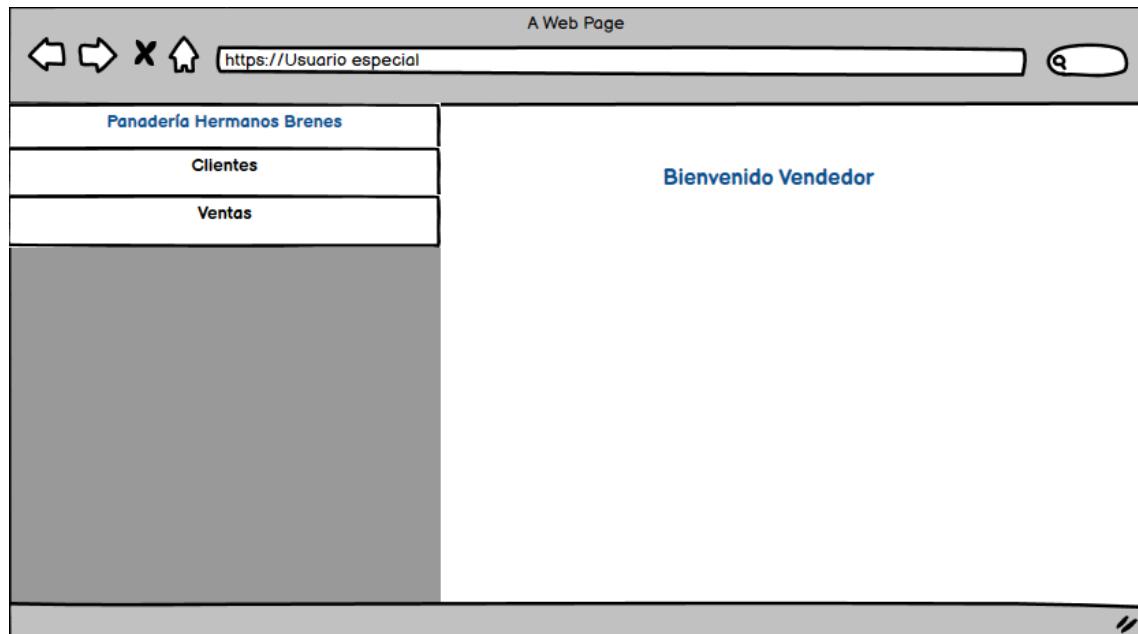


Ilustración 10 Boceto usuario especial

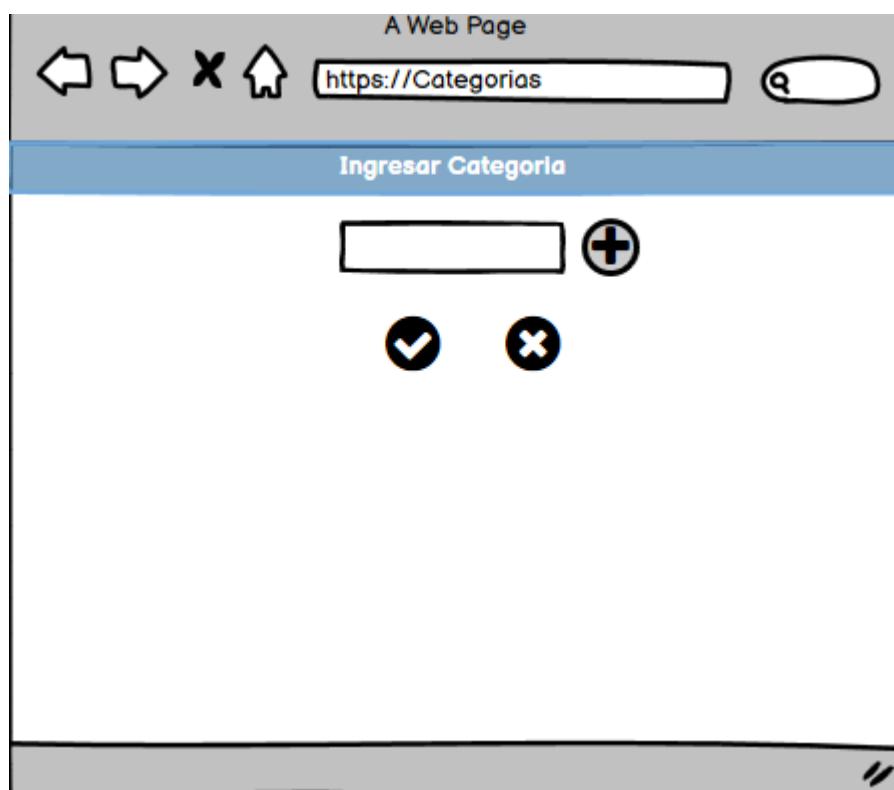


Ilustración 11 Boceto añadir categoría

http://localhost/panaderiaPro/categorias

Administrador Categorías

Agregar categoría

Mostrar 10 Registros Buscar:

Número	Categoría	Acciones
		<input type="radio"/>
		<input checked="" type="checkbox"/>
		<input type="checkbox"/>
		<input checked="" type="checkbox"/>
		Anterior 1 Siguiente

Ilustración 12 Boceto administrar categoría

7.3.6 Capturas de pantalla del Sistema Informático

A continuación, en las ilustraciones 16-21, se muestran las interfaces de los gestores de acceso al sistema, usuarios y categoría; correspondientes a la entrega de la primera iteración.

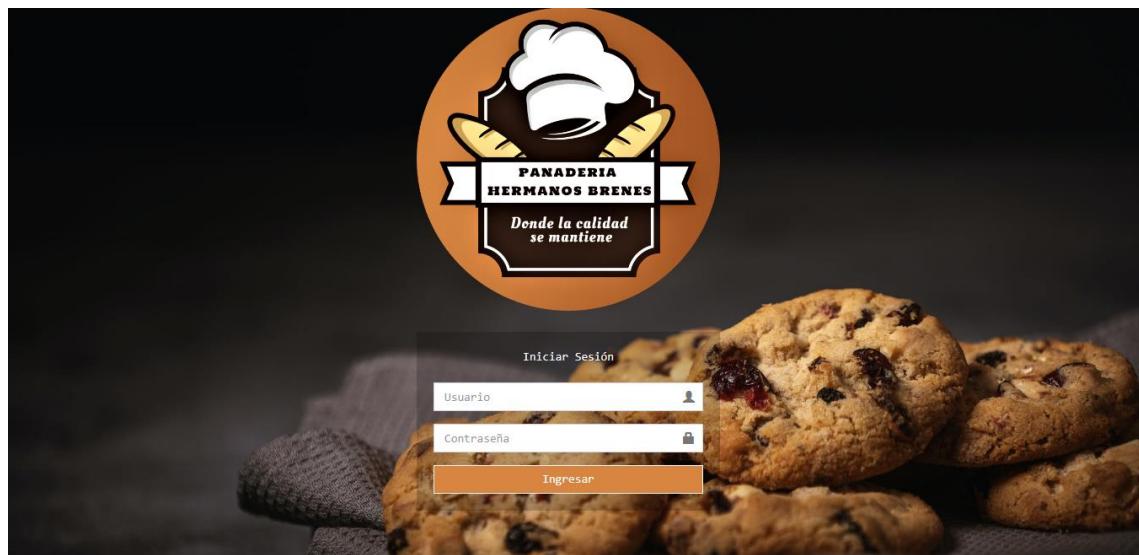


Ilustración 13 Captura de pantalla Vista login

#	Nombre	Usuario	Foto	Perfil	Estado	Último login	Acciones
1	Administrador	admin		Administrador	Activado	2022-09-10 01:18:40	
2	Vendedor	123		Vendedor	Activado	2022-09-09 01:58:06	
3	sofie	sofie		Especial	Activado	2022-09-06 21:58:59	
4	Christopher Paniagua	Chris		Especial	Activado	2022-04-30 01:22:17	

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Ilustración 17 Captura de pantalla administrar usuarios

\$44,513.20

Ventas

[Más información](#)

5

Categorías

[Más información](#)

4

Clientes

[Más información](#)

5

Productos

[Más información](#)

Gráfico de Ventas

Productos más vendidos

Productos Agregados Recientemente

Ilustración 15 Captura de pantalla Vista Panel de control

Panel de Control

Bienvenid@ sofie

Ilustración 16 Captura de pantalla Vista panel de control usuario E

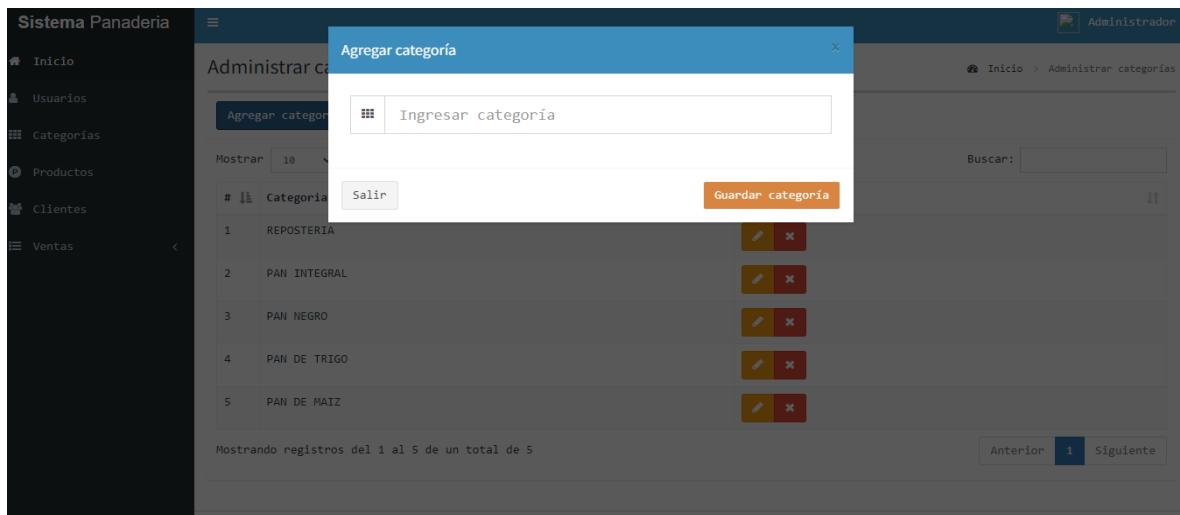


Ilustración 17 Captura de pantalla Vista agregar categoría

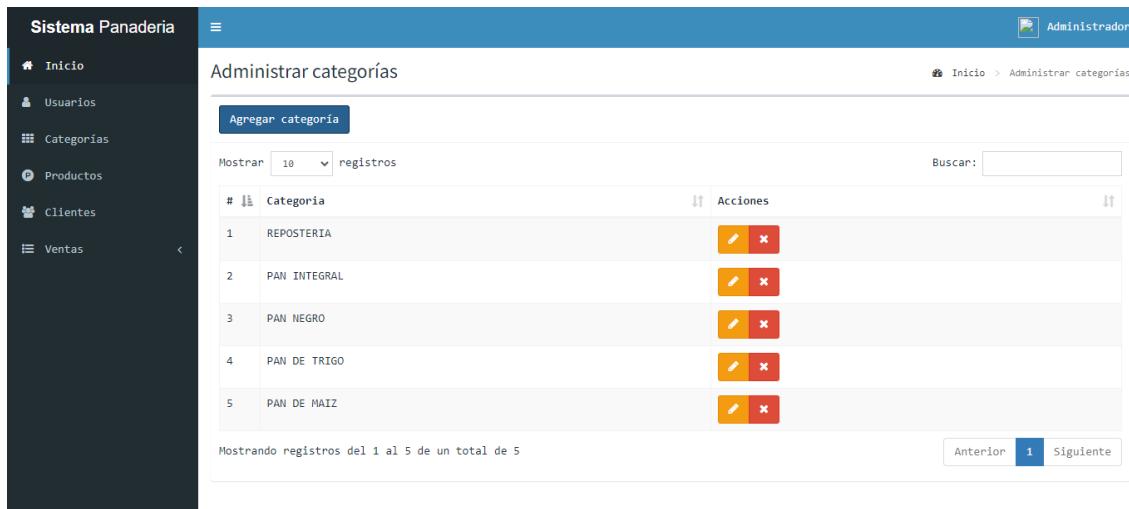


Ilustración 18 Captura de pantalla Vista visualizar categoría

7.3.7 Pruebas de aceptación

Una prueba de aceptación es la evaluación organizada y construida, llevada a cabo para proporcionar información objetiva sobre la calidad del producto o servicio que se está probando, de acuerdo con los requisitos del proyecto.

La intención de la prueba es encontrar errores, defectos u otros problemas de no conformidad en el producto o servicio [8] de lo anterior se puede

decir que una prueba de aceptación es la validación de que el sistema cumple con el funcionamiento esperado permitiendo la aceptación del usuario, estas son de vital importancia para determinar el éxito de cada iteración además garantiza una retroalimentación para el desarrollo de las próximas historias de usuario en las posteriores iteraciones. La plantilla a utilizarse para la elaboración de las pruebas de aceptación y sus componentes es la siguiente:

Tabla 31 Plantilla prueba de aceptación

Caso de pruebas de aceptación	
Código: Número Único que permite identificar la prueba de aceptación	Numero de Historia de usuario: Número único que identifica a la historia de usuario
Historia de Usuario: Nombre que indica de manera general la descripción de la historia de usuario.	
Condiciones de ejecución: E Condiciones previas que deben cumplirse para realizar la prueba de aceptación.	
Entrada/Pasos de ejecución: Pasos que siguen los usuarios para probar la funcionalidad de la historia de usuario	
Resultado esperado: Respuesta del sistema que el cliente espera, después de haber ejecutado una funcionalidad	
Evaluación de la prueba: La Nivel de satisfacción del cliente	

A continuación, en las tablas 32-39 se muestran las pruebas de aceptación que fueron empleadas para validar el éxito de la primera iteración. Los indicadores que se definieron como criterios de evaluación fueron los siguientes:

- Procesos internos propios de gestión de cada módulo que conforma el sistema informático
- Trazabilidad en los formularios
- Operatividad de usuarios

Tabla 32 Pruebas de aceptación Iteración 1

Numero de prueba	Numero de historia de usuario	Nombre de historia de usuario	Nombre de la prueba
1	1	Login	Acceso al sistema
2	2	Almacenamiento de usuarios	Registro de usuario
3	3	Definición de roles de usuarios	Asignación de permisos de usuario
4	4	Validación de tipos de usuarios	Acceso al sistema según tipo de usuario
5	5	Diseño de interfaz administrador	Interacción operativa del usuario de tipo administrador
6	6	Diseño de interfaz usuario especial	Interacción operativa del usuario de tipo especial
7	7	Diseño para registro categoría	Interacción operativa de los usuarios
8	8	Diseño para gestión de los registros categoría	Interacción operativa de los usuarios
9	9	Validación para la gestión de la pestaña categoría	Registro de la pestaña categoría
10	10	Validación para la visualización de la pestaña categoría	Operaciones de guardado de la pestaña categoría
11	11	Validación de la visualización de los registros	Información relevante

Tabla 33 Prueba de aceptación acceso al sistema

Caso de prueba: Acceso al sistema	
Código: 01	Numero de Historia de usuario: 1
Historia de Usuario: Acceso al sistema	
Condiciones de ejecución: El usuario debe de constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none">• Acceder a la pantalla principal del sistema conocido como login.• Completar el formulario de inicio de sesión con las credenciales Nombre de usuario, contraseña• Darle clic sobre el botón ingresar o presionar la tecla enter	
Resultado esperado: De ser correctas las credenciales accederá al sistema desde su perfil y a las funciones que visualice en el sistema dependerán del tipo de usuario designado al perfil.	
Evaluación de la prueba: La prueba se completó satisfactoriamente.	

Tabla 34 Prueba de aceptación almacenamiento de usuario

Caso de prueba: Almacenamiento de Usuario	
Código: 02	Numero de Historia de usuario: 2
Historia de Usuario: Almacenamiento de usuario	
Condiciones de ejecución: El usuario debe de ser de tipo administrador y deberá constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none">• Inicia sesión en el sistema• Accede al módulo o apartado usuarios desde el panel principal• Completa el formulario de agregar un usuario (Nombre de la persona, Nombre de usuario a utilizar, contraseña, tipo de usuario, fotografía)• Una vez registrado el usuario se da clic sobre el estado para activarlo	
Resultado esperado: El nuevo usuario agregado accederá al sistema desde su perfil y a las funciones que visualice en el sistema dependerán del tipo de usuario designado al perfil.	
Evaluación de la prueba: La prueba se completó satisfactoriamente.	

Tabla 35 Prueba de aceptación definición de roles

Caso de prueba: Definición de roles de usuario	
Código: 03	Numero de Historia de usuario: 3
Historia de Usuario: Roles de usuario	
Condiciones de ejecución: El usuario debe de constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema y deberá ingresar con el perfil de tipo administrador.	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none">• Inicia sesión en el sistema	

- | |
|--|
| <ul style="list-style-type: none"> • El usuario de tipo administrador accede al gestor usuarios y crea un perfil de tipo administrador. • El usuario activa este perfil. • El usuario de tipo administrador accede al gestor de usuarios y crea un perfil de tipo usuario especial. • El usuario activa este perfil. • El usuario de tipo administrador accede al gestor de usuarios y crea un perfil de tipo vendedor. • El usuario activa este perfil. • Si el usuario tiene un estado activo puede ingresar sus credenciales al login para acceder al sistema informático. |
|--|

Resultado esperado: El usuario de tipo administrador crea un nuevo perfil para un operador del sistema.

Evaluación de la prueba: La prueba se completó satisfactoriamente.

Tabla 36 Prueba de aceptación validación tipos de usuario

Caso de prueba: Validación de tipos de usuario	
Código: 04	Numero de Historia de usuario: 4
Historia de Usuario: Validación de tipos de usuario	
Condiciones de ejecución: El usuario debe de constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Inicia sesión en el sistema • El usuario ingresa su credencial de tipo admin y tendrá acceso a todos los formularios. • El usuario ingresa su credencial de tipo usuario especial y tendrá acceso únicamente a los formularios designados para ese perfil. • El usuario ingresará su credencial de tipo vendedor y tendrá acceso a los formularios predefinidos para ese perfil. 	
Resultado esperado: Cada usuario en dependencia del perfil que le corresponda podrá tener acceso parcialmente o a todas las funcionalidades del sistema..	
Evaluación de la prueba: La prueba se completó satisfactoriamente.	

Tabla 37 Prueba de aceptación Diseño de interfaz admin

Caso de prueba: Diseño de interfaz para el administrador	
Código: 05	Numero de Historia de usuario: 5
Historia de Usuario: Diseño de interfaz para el administrador	
Condiciones de ejecución: El usuario debe de constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema de tipo administrador.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Inicia sesión en el sistema 	

- | |
|--|
| <ul style="list-style-type: none"> • El usuario ingresa su credencial de tipo admin y tendrá acceso a todos los formularios. • El usuario visualiza la ventana principal home. • Los módulos del sistema serán funcionales. • El usuario puede acceder a cualquier modulo y puede hacer uso de sus distintas funcionalidades. • El usuario llene los formularios que desee. |
|--|

Resultado esperado: El usuario interactúa con éxito en la interfaz.

Evaluación de la prueba: La prueba se completó satisfactoriamente.

Tabla 38 Prueba de aceptación diseño de interfaz usuario especial

Caso de prueba: Diseño de interfaz para el usuario especial	
Código: 06	Numero de Historia de usuario: 6
Historia de Usuario: Diseño de interfaz para el usuario especial	
Condiciones de ejecución: El usuario debe de constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema de tipo usuario especial.	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none"> • Inicia sesión en el sistema • El usuario ingresa su credencial de tipo usuario especial y tendrá acceso a los formularios de ventas y categorías. • El usuario visualiza la ventana principal home. • Los módulos de ventas y categorías serán funcionales. • El usuario llena el formulario y guarda la información. • El usuario realiza una búsqueda de una venta. 	
Resultado esperado: El usuario interactúa con éxito en la interfaz.	
Evaluación de la prueba: La prueba se completó satisfactoriamente.	

Tabla 39 Prueba de aceptación diseño de interfaz categoría

Caso de prueba: Diseño de interfaz Categoría	
Código: 07	Numero de Historia de usuario: 7,8,9,10
Historia de Usuario: Categoría	
Condiciones de ejecución: El usuario debe de constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema .	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none"> • Inicia sesión en el sistema • El usuario visualiza la ventana principal home. • El usuario interactúa correctamente con la interfaz de usuario. • El usuario llena el formulario y guarda la información. • El usuario realiza una búsqueda de alguna categoría. • El usuario elimina o actualiza una categoría. 	

- En sistema notifica al usuario si no se está ingresando correctamente la información.
- El sistema notifica al usuario si falta información.
- Si falta información el sistema no guarda la información.
- El sistema permite personalizar las categorías.
- El usuario visualiza los registros de categoría.

Resultado esperado: El usuario registra con éxito una categoría.

7.3.8 Resultados de la primera iteración

Como resultado de la entrega correspondiente a la primera iteración del sistema informático Panadería Hermanos Brenes, el cliente quedó satisfecho con las funcionalidades de los módulos que se desarrollaron hasta este momento, así mismo con las interfaces desarrolladas en conformidad con los bocetos previamente diseñados y aceptados por el cliente.

7.4 Segunda iteración

A la iteración número 2 del plan de entrega de proyecto le corresponden las siguientes historias de usuario:

Tabla 40 Plan de entrega de proyecto Iteración 2

Numero	Nombre
4	Inventario de productos existentes
5	Registros de clientes
6	Gestión de ventas

7.4.1 Tareas de ingeniería iteración 2

A continuación, se detallan todas las tareas de ingeniería correspondiente a la entrega de la segunda iteración:

Tabla 41 Tareas de Ingeniería iteración 2

Numero de tarea	Numero de historias	Nombre de la tarea
1	4	Diseño, codificación y validación del módulo inventario de productos existentes
2	4	Validación del dataset de la visualización del stock de productos existentes
3	5	Diseño, codificación y validación del módulo clientes
4	5	Validación del dataset de la visualización del stock de clientes
5	6	Diseño, codificación y validación del módulo creación de ventas
6	6	Validación del dataset de la visualización del stock de creación de ventas
7	6	Automatización de procesos de gestión de ventas
8	6	Validación de los procesos de gestión de ventas

A continuación, en las tablas 42-49 se muestran las historias de usuario que fueron empleadas para llevar a cabo el desarrollo del sistema informático.

Tabla 42 Tarea de Ingeniería 1 Iteración 2 Módulos productos existentes

Tarea de Ingeniería	
Numero de tarea: 1	Numero de historia: 4
Nombre de la tarea: Diseño, codificación y validación del módulo inventario de productos existentes	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 25/09/2021	Fecha fin: 30/09/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se realiza un diseño de interfaz para el módulo de los productos existentes, a manera de utilizarlo como inventario, posteriormente se codifica los procesos que estarán presente en la gestión del stock existentes, tales como, agregar productos, eliminarlos, editar un producto, buscar un producto, finalmente se valida la codificación.	

Tabla 43 Tarea de Ingeniería 2 Iteración 2 Visualización dataset

Tarea de Ingeniería	
Numero de tarea: 2	Numero de historia: 4
Nombre de la tarea: Validación del dataset de la visualización del stock de productos existentes	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 11/10/2021	Fecha fin: 15/10/2021
Programador: Sofía Alejandra Miranda Valle	

Descripción: Se valida la visualización de los registros de productos existentes así como la automatización de algunos subprocesos de gestión.

Tabla 44 Tarea de Ingeniería 3 Iteración 2 Modulo clientes

Tarea de Ingeniería	
Numero de tarea: 3	Numero de historia: 5
Nombre de la tarea: Diseño, codificación y validación del módulo clientes	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 17/10/2021	Fecha fin: 21/10/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se realiza un diseño de interfaz correspondiente al registro de clientes y una interfaz para los procesos de gestión, posteriormente se codifica los procesos y subprocesos que estarán presente en la gestión del módulo cliente y finalmente se valida la codificación.	

Tabla 45 Tarea de Ingeniería 4 Iteración 2 Validación del dataset clientes

Tarea de Ingeniería	
Numero de tarea: 4	Numero de historia: 5
Nombre de la tarea: Validación del dataset de la visualización del módulo clientes	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 22/10/2021	Fecha fin: 30/10/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se valida la visualización de los registros de clientes, de igual forma la automatización de algunos subprocesos de gestión.	

Tabla 46 Tarea de Ingeniería 5 Iteración 2 Modulo creación de ventas

Tarea de Ingeniería	
Numero de tarea: 5	Numero de historia: 6
Nombre de la tarea: Diseño, codificación y validación del módulo creación de ventas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 02/11/2021	Fecha fin: 12/10/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se realiza un diseño de interfaz correspondiente al registro de ventas y una interfaz para los procesos de gestión, posteriormente se codifica los subprocesos que estarán presente en la gestión de la creación de una venta y finalmente se valida la codificación.	

Tabla 47 Tarea de Ingeniería 6 Iteración 2 Validación del dataset ventas

Tarea de Ingeniería	
Numero de tarea: 6	Numero de historia: 6
Nombre de la tarea: Validación del dataset de la visualización del módulo ventas.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/10/2021	Fecha fin: 17/10/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se valida la visualización de los registros de ventas existentes, de igual forma la automatización de algunos subprocesos de gestión.	

Tabla 48 Tarea de Ingeniería 7 Iteración 2 Automatización gestión ventas

Tarea de Ingeniería	
Numero de tarea: 7	Numero de historia: 6
Nombre de la tarea: Automatización de procesos de gestión de ventas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 18/10/2021	Fecha fin: 21/10/2021
Programador: Christopher Daniel Paniagua Martinez	
Descripción: Se crean instrucciones para que se reduzca la complejidad del llenado de los formularios, de manera que al realizar un nuevo registro de ventas se autocomplete o se pueda seleccionar o disponer de información de otros formularios.	

Tabla 49 Tarea de Ingeniería 8 Iteración 2 Validación gestión ventas

Tarea de Ingeniería	
Numero de tarea: 8	Numero de historia: 6
Nombre de la tarea: Validación de los procesos de gestión de ventas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 22/10/2021	Fecha fin: 23/10/2021
Programador: Christopher Daniel Paniagua Martinez	
Descripción: Se valida que las instrucciones anteriormente creadas en la historia de usuario pasada, resulten funcionales	

7.4.2 Tarjetas CRC

A continuación, en las tablas 50-56 se muestran las tarjetas CRC que fueron empleadas para llevar a cabo el desarrollo del sistema informático.

Tabla 50 Tarjeta CRC Tarea de Ingeniería 1 Modulo productos existentes

Clase: Diseño, codificación y validación del módulo inventario de productos existentes	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, usuario especial, vendedor
Diseñar la interfaz de administración de productos existentes	
Diseñar la interfaz de agregar productos	
Trazar el gestor categoría con el nuevo gestor módulo de inventario	
Personalizar los productos existentes y realizar búsquedas de productos mediante el código	
Verificar el stock de productos existentes	
Permitir guardar un producto	
Permitir eliminar un producto	
Permitir actualizar un producto	

Devolver mensajes en caso en caso que se logró o no se logró ingresar el producto	
Notificar si falta información para añadir un producto	
Calcular porcentaje de reducción de precios en el producto	

Tabla 51 Tarjeta CRC Tarea de Ingeniería 2 Validación dataset productos

Clase: Diseño, codificación y validación del módulo inventario de productos existentes	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, usuario especial, vendedor
Visualizar información de productos existentes en cantidades de 10,25,50 y 100 registros	
Permitir eliminar/actualizar productos desde el dataset	

Tabla 52 Tarjeta CRC Tarea de Ingeniería 3 Modulo clientes

Clase: Diseño, codificación y validación del módulo inventario de productos existentes	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, usuario especial, vendedor
Diseñar la interfaz del módulo clientes existentes	

Diseñar la interfaz de registrar un cliente	
Permitir registrar un cliente	
Permitir dar de baja a un cliente	
Permitir editar información del cliente	
Permitir actualizar información del cliente	
Devolver mensajes en caso en caso que se logró o no se logró realizar el registro del nuevo cliente	
Notificar si falta información para añadir un cliente	
Notificar en caso que no se haya realizado con éxito la operación de actualizar la información del cliente	

Tabla 53 Tarjeta CRC Tarea de Ingeniería 4 Validación dataset clientes

Clase: Validación del dataset del módulo clientes	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, usuario especial, vendedor
Visualizar información de los clientes en cantidades de 10,25,50 y 100 registros	
Permitir eliminar/actualizar clientes desde el dataset	

Permitir realizar búsquedas de clientes mediante su número de identificación y nombre	
---	--

Tabla 54 Tarjeta CRC Tarea de Ingeniería 5 Modulo ventas

Clase: Diseño, codificación y validación del módulo registro de ventas	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, vendedor
Diseñar la interfaz del módulo registro de ventas	
Diseñar la interfaz de crear una venta	
Permitir ingresar una venta	
Permitir eliminar una venta	
Permitir actualizar información de un cliente existente	
Permitir utilizar información del gestor del cliente	
Devolver mensajes en caso en caso que se logró o no se logró concretar la venta	
Notificar si falta información para crear la venta	
Permitir agregar cliente nuevo	
Permitir visualizar el registro de los productos existentes	
Permitir manipular la información de los productos existentes	

Permitir buscar dentro del registro de los productos existente	
Permitir seleccionar método de pago a manera de registro	

Tabla 55 Tarjeta CRC Tarea de Ingeniería 6 Validación dataset ventas

Clase: Validación del dataset del módulo ventas	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, usuario especial, vendedor
Visualizar información de los clientes en cantidades de 10,25,50 y 100 registros de ventas	
Permitir eliminar/actualizar clientes desde el dataset	
Permitir eliminar/buscar ventas desde el dataset	
Permitir realizar búsquedas de clientes mediante su número de identificación y nombre	

Tabla 56 Tarjeta CRC Tarea de Ingeniería 7,8 Gestión de ventas

Clase: Automatización y Validación gestión de ventas	
Responsabilidad	Christopher Daniel Paniagua Martínez
Colaboradores	Administrador, usuario especial, vendedor

Trazar gestor de usuarios con el gestor de creación de ventas	
Trazar gestor clientes con el gestor de ventas	
Trazar ventas con el gestor creación de ventas	

7.4.3 Codificación

En esta sección se presentan los resultados de la etapa de escritura de la solución propuesta del sistema informático, en otras palabras, el código fuente de las instrucciones definidas en el lenguaje de programación.

```
=====
static public function mdlIngresarProducto($tabla, $datos){

    $stmt = Conexion::conectar()->prepare("INSERT INTO $tabla(id_categoria, codigo, descripcion, imagen, stock, precio_compra, precio_venta)

    $stmt->bindParam(":id_categoria", $datos["id_categoria"], PDO::PARAM_INT);
    $stmt->bindParam(":codigo", $datos["codigo"], PDO::PARAM_STR);
    $stmt->bindParam(":descripcion", $datos["descripcion"], PDO::PARAM_STR);
    $stmt->bindParam(":imagen", $datos["imagen"], PDO::PARAM_STR);
    $stmt->bindParam(":stock", $datos["stock"], PDO::PARAM_STR);
    $stmt->bindParam(":precio_compra", $datos["precio_compra"], PDO::PARAM_STR);
    $stmt->bindParam(":precio_venta", $datos["precio_venta"], PDO::PARAM_STR);

    if($stmt->execute()){

        return "ok";
    }else{
        return "error";
    }
}
```

Ilustración 22 Codificación productos existentes

```
static public function mdlEliminarProducto($tabla, $datos){

    $stmt = Conexion::conectar()->prepare("DELETE FROM $tabla WHERE id = :id");

    $stmt -> bindParam(":id", $datos, PDO::PARAM_INT);

    if($stmt -> execute()){

        return "ok";
    }else{
        return "error";
    }
    $stmt -> close();
    $stmt = null;
}
```

Ilustración 23 Codificación eliminar productos

Ilustración 19 Codificación crear cliente Parte I

```

35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
    confirmButtonText: "Cerrar"
  }).then(function(result){
    if (result.value) {
      window.location = "clientes";
    }
  })
}
else{
  echo'<script>
  swal({
    type: "error",
    title: "El cliente no puede ir vacío o llevar caracteres especiales!",
    showConfirmButton: true,
    confirmButtonText: "Cerrar"
  }).then(function(result){
    if (result.value) {
      window.location = "clientes";
    }
  })
}
</script>';
}
}

```

```

/
8  if(isset($_POST["nuevoCliente"])){
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
if(isset($_POST["nuevoCliente"])){
  if(preg_match('/^a-zA-Z0-9ñáéíóúÁÉÍÓÚ ]+$/', $_POST["nuevoCliente"]) &&
    preg_match('/^0-9]+$', $_POST["nuevoDocumentoId"]) &&
    preg_match('/^0-9][a-zA-Z0-9_]+([.][a-zA-Z0-9_]+)*[@][a-zA-Z0-9_]+([.][a-zA-Z0-9_]+)*[.][a-zA-Z0-9_]+$', $_POST["nuevoEmail"]) &&
    preg_match('/^0-9]+$', $_POST["nuevoTelefono"]) &&
    preg_match('/^#.\a-zA-Z0-9 ]+$/', $_POST["nuevaDireccion"])){
    $tabla = "clientes";
    $datos = array("nombre"=>$_POST["nuevoCliente"],
                  "documento"=>$_POST["nuevoDocumentoId"],
                  "email"=>$_POST["nuevoEmail"],
                  "telefono"=>$_POST["nuevoTelefono"],
                  "direccion"=>$_POST["nuevaDireccion"],
                  "fecha_nacimiento"=>$_POST["nuevaFechaNacimiento"]);
    $respuesta = ModeloClientes::mdlIngresarCliente($tabla, $datos);
    if($respuesta == "ok"){
      echo'<script>
      swal({
        type: "success",
        title: "El cliente ha sido guardado correctamente",
        showConfirmButton: true,
        confirmButtonText: "Cerrar"
      }).then(function(result{
        if (result.value) {

```

Ilustración 20 Codificación crear cliente Parte II

7.4.4 Bocetos

A continuación, en las ilustraciones 26-30 se presentan los bocetos que se diseñaron con el cliente para llevar a cabo la creación del sistema.

http://localhost/panaderia/clientes

Agregar cliente

	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
<input type="button" value="Salir"/>	<input type="button" value="Guardar"/>

Ilustración 21 Boceto registro de cliente

Ilustración 22 Boceto administrar clientes

http://localhost/productos

Inventario de productos existentes

Realizar venta

Mostrar 10 Registros Buscar:

Código	Acciones	Imagen	Precio venta	Precio compra	Productos existentes	Descripción	Stock
	<input checked="" type="radio"/>						
	<input checked="" type="checkbox"/>						
	<input type="checkbox"/>						
	<input checked="" type="checkbox"/>						
	<input type="checkbox"/>						

Anterior 1 Siguiente

Ilustración 23 Boceto de inventario de productos existentes

Agregar cliente

Crear Venta

Administrador Seleccionar cliente Agregar cliente Impuesto Total

Código Método pago Mostrar 10 Registros Buscar:

Código	Acciones	Imagen	Precio venta	Precio compra	Productos existentes	Descripción	Stock
	<input checked="" type="radio"/>						
	<input checked="" type="checkbox"/>						
	<input type="checkbox"/>						
	<input checked="" type="checkbox"/>						
	<input type="checkbox"/>						

Anterior 1 Siguiente

Ilustración 24 Boceto registro cliente

http://localhost/panaderiapro/clientes

Administrador Clientes

Agregar cliente

Mostrar 10 Registros Buscar:

#	Nombre	Acciones	Email	Teléfono	Dirección	ID
		<input checked="" type="radio"/>				
		<input checked="" type="checkbox"/>				
		<input type="checkbox"/>				
		<input checked="" type="checkbox"/>				
		<input type="checkbox"/>				

Anterior 1 Siguiente

Ilustración 25 Boceto gestión de ventas

7.4.5 Capturas de pantalla

A continuación, en las ilustraciones 31-35, se muestran las interfaces de los gestores de acceso al sistema, usuarios y categoría; correspondientes a la entrega de la primera iteración.

The image shows a user interface titled 'Agregar cliente' (Add Client) in a blue header bar. Below the header are six input fields, each with a small icon and a placeholder text. The fields are: 'Ingresar nombre' (Enter name) with a person icon, 'Ingresar documento' (Enter document) with a magnifying glass icon, 'Ingresar email' (Enter email) with an envelope icon, 'Ingresar teléfono' (Enter phone) with a phone icon, 'Ingresar dirección' (Enter address) with a location pin icon, and 'Ingresar fecha nacimiento' (Enter date of birth) with a calendar icon. At the bottom left is a 'Salir' (Exit) button, and at the bottom right is a large orange 'Guardar cliente' (Save client) button.

Ilustración 26 Captura de pantalla registro de ventas

Administrar clientes

[Inicio](#) > Administrar clientes

Administrador de Clientes												
Listado de Clientes												
Nuevo Cliente												
Listado de Clientes												
#	Nombre	Documento ID	Email	Teléfono	Dirección	Fecha nacimiento	Total compras	Última compra	Ingreso al sistema	Acciones		
1	Marcos	9	maernesto@gmail.com	(999) 999-9999	calle 13	1973-04-13	15	2022-05-05 09:20:25	2022-05-08 05:49:06	 		
2	Ernesto	1	evega@gmail.com	(222) 222-2222	Masatepe	1990-08-25	0	0000-00-00 00:00:00	2022-05-05 08:57:04	 		
3	Marie Sanchez	1254	mmariee34@gmail.com	(885) 588-5545	San Marcos	1975-02-27	100	2022-05-05 10:47:17	2022-05-05 09:47:17	 		
4	Christopher Paniagua	5685	chrispaniagua@gmail.com	(584) 949-4254	Domitila Lugo	1983-02-14	1258	2022-05-05 10:46:20	2022-05-05 09:46:20	 		

Mostrando registros del 1 al 4 de un total de 4

Anterior **1** Siguiente

Ilustración 27 Captura de pantalla dataset clientes

 **Seleccionar categoría**

 **Ingresar código**

 **Ingresar descripción**

 **Stock**

 **Precio de venta**  **Precio de compra**

40  **Utilizar procentaje**

SUBIR IMAGEN

Seleccionar archivo Sin archivos seleccionados
Peso máximo de la imagen 2MB



Ilustración 28 Captura de pantalla añadir productos existentes

Agregar producto

Mostrar 10 registros Buscar:

#	Imagen	Código	Descripción	Categoría	Stock	Precio de compra	Precio de venta	Agregado	Acciones
1		25487	Pudines	Repostería	58855	15	15	2022-05-08 05:49:06	
2		004	Pan de ajonjoli	Pan de trigo	0	20	28.4	2022-05-05 09:43:23	
3		003	Pan aniquelado con mantequilla	Pan Negro	25	15	21	2022-05-08 05:49:06	
4		002	Pan dietetico	Pan Integral	5	28	36.4	2022-05-08 05:49:06	
5		001	Polvorones de canela	Repostería	75	50	55	2022-05-08 05:49:06	

Mostrando registros del 1 al 5 de un total de 5

Anterior **1** Siguiente

Ilustración 29 Captura de pantalla productos existentes

Crear venta

Administrador

10005

Seleccionar cliente

Impuesto	Total
0	%
\$	00000

Seleccione método de pago

Guardar venta

Inicio > Crear venta

Mostrar 10 registros Buscar:

#	Imagen	Código	Descripción	Stock	Acciones
1		25487	Pudines	58855	
2		004	Pan de ajonjoli	0	
3		003	Pan aniquelado con mantequilla	25	
4		002	Pan dietetico	5	
5		001	Polvorones de canela	75	

Mostrando registros del 1 al 5 de un total de 5

Anterior **1** Siguiente

Ilustración 30 Captura de pantalla crear ventas

7.4.6 Pruebas de aceptación

A continuación, en las tablas 57-60 se muestran las pruebas de aceptación que fueron empleadas para validar el éxito de la segunda iteración. Los indicadores que se definieron como criterios de evaluación fueron los siguiente.

Tabla 57 Pruebas de aceptación segunda iteración

Numero de prueba	Numero de historia	Nombre de la prueba
1	4	Inventario de productos existentes
2	5	Registro de clientes
3	6	Gestión de ventas

Tabla 58 Prueba de aceptación inventarios de productos existentes

Caso de prueba: Inventarios de productos existentes	
Código: 04	Numero de Historia de usuario: 4
Historia de Usuario: Inventarios de productos existentes	
Condiciones de ejecución: El usuario debe de ser de tipo administrador o usuario especial , el usuario deberá estar activo y debe constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none">• Inicia sesión en el sistema• Accede al módulo o apartado productos desde el panel principal• Completa el formulario, selecciona una categoría, ingresa el código el producto, la descripción o nombre, la cantidad en existencia, el precio de venta, el precio de compra, se estipula el porcentaje que se le quiere ganar al precio producto, si así se desea, en caso de que	

se desee mantener el precio justo del producto se desmarca la casilla porcentaje, seguidamente selecciona una imagen del producto y finalmente se guarda el registro.
Resultado esperado: La información del producto podrá visualizarse en el registro de los productos, así como en el módulo de creación de ventas para posteriormente utilizarse al efectuar una venta.
Evaluación de la prueba: La prueba se completó satisfactoriamente.

Tabla 59 Prueba de aceptación registro clientes

Caso de prueba: Registro de clientes	
Código: 05	Número de Historia de usuario: 5
Historia de Usuario: Registro de clientes	
Condiciones de ejecución: El usuario debe de ser de tipo administrador o usuario vendedor , el usuario deberá estar activo y debe constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema	
<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> • Inicia sesión en el sistema • Accede al módulo o apartado clientes desde el panel principal • Completa el formulario de agregar un cliente detallando el nombre del cliente, en este caso un cliente es un establecimiento, el código del establecimiento, el email, el número de teléfono, la dirección y la fecha de nacimiento de quien realiza la venta. • Guarda el registro de cliente 	
Resultado esperado: el cliente registrado podrá visualizarse en el registro de clientes para posteriormente utilizarse en el módulo de creación de ventas	
Evaluación de la prueba: La prueba se completó satisfactoriamente.	

Tabla 60 Prueba de aceptación gestión de ventas

Caso de prueba: Gestión de ventas	
Código: 06	Número de Historia de usuario: 6
Historia de Usuario: Gestión de ventas	
Condiciones de ejecución: El usuario debe de ser de tipo administrador o usuario vendedor , el usuario deberá estar activo y debe constar con las credenciales (Nombre de usuario, contraseña) para ingresar al sistema	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none">• Inicia sesión en el sistema• Accede al módulo o apartado ventas, crear una venta, desde el panel principal• Completa el formulario ingresa un cliente este puede estar previamente registrado aunque también se puede realizar un nuevo registro desde esa pestaña, agrega los productos a comprar e ingresa la cantidad que de productos que desea adquirir, en caso que se presente un impuesto se define el porcentaje, posteriormente se define el método de pago, en caso de ser en efectivo se ingresa el monto que el cliente pago, para calcular el vuelto, en caso de utilizar métodos de pago de crédito o débito se ingresa el código de la transacción• Se efectúa la venta	
Resultado esperado: La información del registro de venta ingresado podrá estar disponible y a su vez visualizarse en el módulo de administración de ventas, el cual únicamente tiene acceso el usuario de tipo administrador del sistema informático.	
Evaluación de la prueba: La prueba se completó satisfactoriamente.	

7.5 Tercera iteración

A la iteración número 3 del plan de entrega de proyecto le corresponden las siguientes historias de usuario:

Tabla 61 Plan entrega de proyecto Iteración 3

Numero	Historias de usuario	Nombre
1	7	Administración de ventas
2	8	Generación de reportes

7.5.1. Tareas de Ingeniería tercera iteración

A continuación, se detallan todas las tareas de ingeniería correspondiente a la entrega de la tercera iteración:

Tabla 62 Tareas de Ingeniería Tercera Iteración

Numero de tarea	Numero de historias	Nombre de la tarea
1	7	Diseño de interfaz para la administración de ventas
2	7	Codificación de los procesos de gestión para la administración de ventas
3	7	Validación de la codificación de los procesos de gestión de la

		administración de ventas
4	7	Diseño de factura, recibos
5	7	Impresión de las facturas o recibos
6	7	Obtención de visualización por rango de fechas
7	8	Diseño de interfaz para el módulo de reportes
8	8	Diseñar la estructura de la tabla de reportes en Excel
9	8	Validar la exportación de la información de los reportes en archivo Excel
10	8	Codificar subprocessos de la administración de ventas
11	8	Validar los subprocessos de la administración de ventas

Tabla 63 Tarea de Ingeniería 1 Iteración 3 Diseño de interfaz de ventas

Tarea de Ingeniería	
Numero de tarea: 1	Numero de historia: 7
Nombre de la tarea: Diseño de interfaz para la administración de ventas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/01/2022	Fecha fin: 07/01/2022
Programador: Sofía Alejandra Miranda Valle	

Descripción: Se realiza el diseño de interfaz para el módulo de administración de la ventas del sistema informático .

Tabla 64 Tarea de Ingeniería 1 Iteración 3 Codificación procesos de venta

Tarea de Ingeniería	
Numero de tarea: 2	Numero de historia: 7
Nombre de la tarea: Codificación de los procesos de gestión para la administración de ventas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 07/01/2022	Fecha fin:10/01/2022
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se codifican todos los procesos de gestión del módulo de la administración de las ventas	

Tabla 65 Tarea de Ingeniería 3 Iteración 3 Validación procesos ventas

Tarea de Ingeniería	
Numero de tarea: 3	Numero de historia: 7
Nombre de la tarea: Validación de la codificación de los procesos de gestión de la administración de ventas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 07/01/2022	Fecha fin:07/01/2022
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se valida de todos los procesos de la gestión de administración de ventas existentes.	

Tabla 66 Tarea de Ingeniería 4 Iteración 3 Diseño factura

Tarea de Ingeniería	
Numero de tarea: 4	Numero de historia: 7
Nombre de la tarea: Diseño de factura, recibos	

Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 08/01/2022	Fecha fin: 08/01/2022
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se realiza un diseño de factura para imprimirse después de efectuar una venta a su vez se realiza un diseño de recibo para imprimirse después de efectuar una venta.	

Tabla 67 Tarea de Ingeniería 5 Iteración 3 Impresión recibos

Tarea de Ingeniería	
Numero de tarea: 5	Numero de historia: 7
Nombre de la tarea: Impresión de las facturas o recibos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 08/01/2022	Fecha fin: 08/01/2022
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se valida la correcto funcionamiento de impresión de la factura y recibos .	

Tabla 68 Tarea de Ingeniería 6 Iteración 3 Stock clientes

Tarea de Ingeniería	
Numero de tarea: 6	Numero de historia: 7
Nombre de la tarea: Validación del dataset de la visualización del stock de clientes	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 28/10/2021	Fecha fin: 23/11/2021
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se valida la visualización de los registros del módulo de los clientes así como la automatización de algunos subprocesos de gestión.	

Tabla 69 Tarea de Ingeniería 7 Iteración 3 Histograma

Tarea de Ingeniería	
Numero de tarea: 7	Numero de historia: 7
Nombre de la tarea: Obtención de visualización por rango de fechas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 08/01/2022	Fecha fin: 10/01/2022
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se realiza el proceso de obtención de las ventas realizadas por distintos rangos de fecha posteriormente se valida la funcionalidad de este proceso	

Tabla 70 Tarea de Ingeniería Iteración 3 Diseño interfaz reportes

Tarea de Ingeniería	
Numero de tarea: 8	Numero de historia: 8
Nombre de la tarea: Diseño de interfaz para el módulo de reportes	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 10/01/2022	Fecha fin: 11/01/2022
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se realiza un diseño de interfaz para el módulo de reportes.	

Tabla 71 Tarea de Ingeniería 9 Iteración 3 Reportes en Excel

Tarea de Ingeniería	
Numero de tarea: 9	Numero de historia: 8
Nombre de la tarea: Validar la exportación de la información de los reportes en archivo Excel	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 12/01/2022	Fecha fin: 13/01/2022
Programador: Sofía Alejandra Miranda Valle	

Descripción: Se realiza un estructura de reportes en Excel y se codifica el proceso para finalmente realizar la validaciones pertinentes

Tabla 72 Tarea de Ingeniería 10 Iteración 3 Codificación subprocessos ventas

Tarea de Ingeniería	
Numero de tarea: 10	Numero de historia: 8
Nombre de la tarea: Codificación de subprocessos de la administración de ventas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/01/2022	Fecha fin: 14/01/2022
Programador: Sofía Alejandra Miranda Valle	
Descripción: Una vez determinados los subprocessos de administración de ventas se realiza la codificación y automatización de estos	

Tabla 73 Tarea de Ingeniería 11 Iteración 3 Validar subprocessos ventas

Tarea de Ingeniería	
Numero de tarea: 11	Numero de historia: 8
Nombre de la tarea: Validar los subprocessos de la administración de ventas	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 14/01/2022	Fecha fin: 15/01/2022
Programador: Sofía Alejandra Miranda Valle	
Descripción: Se validan los subprocessos previamente codificados en la administración de ventas.	

7.5.2 Tarjetas CRC

A continuación, en esta sección se muestran las tarjetas CRC que fueron utilizadas al desarrollar este sistema informático.

Tabla 74 Tarjeta CRC administración de ventas

Clase: Diseño, codificación y validación de la administración de ventas	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, vendedor
Visualizar todas las ventas realizadas de todos los vendedores	
Visualizar las ventas en un registro de 10, 25, 50, 100 registros	
Trazar el gestor agregar ventas con el gestor administrar ventas	
Filtrar información a manera de histograma por rangos de fechas: ayer, último mes, últimos 30 días, rango personalizado de fecha	
Visualizar los resultados obtenidos en el dataset de administración de ventas	

Tabla 75 Tarjeta CRC reporte de ventas

Clase: Diseño, codificación y validación de la administración de ventas	
Responsabilidad	Sofie Valle
Colaboradores	Administrador, vendedor
Obtener un gráfico de punto de las ventas	
Obtener un rango con filtro personalizado por fechas	
Visualizar el stock existente	
Visualizar los resultados de mejores vendedores	
Visualizar los resultados obtenidos de productos más vendidos	
Descargar Reporte de ventas en archivos Excel y PDF	

7.5.3 Codificación

En esta sección se presenta los resultados del código fuente de las instrucciones definidas en el lenguaje de programación correspondiente a la entrega de la tercera iteración.

```

modelos > ventas.modelo.php
 1  <?php
 2
 3  require_once "conexion.php";
 4
 5  class ModeloVentas{
 6
 7
 8
 9      static public function mdlMostrarVentas($tabla, $item, $valor){
10
11      if($item != null){
12
13          $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla WHERE $item = :$item ORDER BY id ASC");
14
15          $stmt -> bindParam(":.$item, $valor, PDO::PARAM_STR");
16
17          $stmt -> execute();
18
19          return $stmt -> fetch();
20
21      }else{
22
23          $stmt = Conexion::conectar()->prepare("SELECT * FROM $tabla ORDER BY id ASC");
24
25          $stmt -> execute();
26
27          return $stmt -> fetchAll();
28
29      }
30
31      $stmt -> close();
32
33      $stmt = null;

```

Ilustración 31 Codificación administración de ventas

```

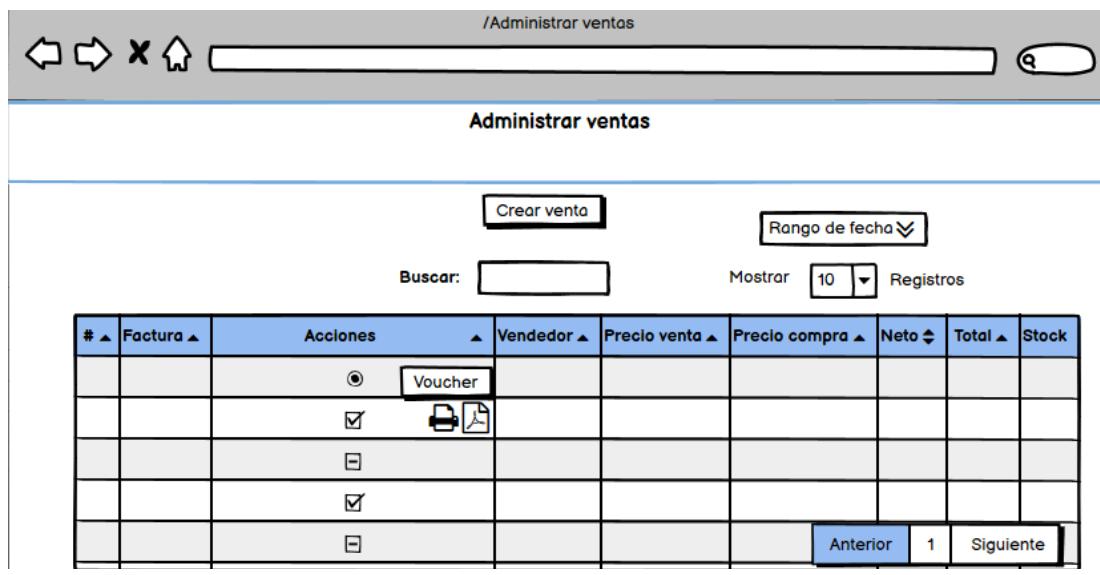
controladores > ventas.controlador.php
 575  public function ctrDescargarReporte(){
 576
 577      if(isset($_GET["reporte"])){
 578
 579          $tabla = "ventas";
 580
 581          if(isset($_GET["fechaInicial"]) && isset($_GET["fechaFinal"])){
 582
 583              $ventas = ModeloVentas::mdlRangoFechasVentas($tabla, $_GET["fechaInicial"], $_GET["fechaFinal"]);
 584
 585          }else{
 586
 587              $item = null;
 588              $valor = null;
 589
 590              $ventas = ModeloVentas::mdlMostrarVentas($tabla, $item, $valor);
 591
 592          }
 593
 594
 595          $Name = $_GET["reporte"].'.xls';
 596
 597          header('Expires: 0');
 598          header('Cache-control: private');
 599          header("Content-type: application/vnd.ms-excel"); // Archivo de Excel
 600          header("Cache-Control: cache, must-revalidate");
 601          header('Content-Description: File Transfer');
 602          header('Last-Modified: '.date('D, d M Y H:i:s'));
 603          header("Pragma: public");
 604          header('Content-Disposition: filename="'.$Name.'"');
 605          header("Content-Transfer-Encoding: binary");
 606

```

Ilustración 32 Generación reporte de ventas

7.5.4 Bocetos

A continuación, en las siguientes ilustraciones se presentan los bocetos que se diseñaron con el cliente para llevar a cabo la culminación del sistema informático.



Este boceto muestra la interfaz de administrar ventas. La barra superior tiene iconos para navegar entre páginas y un campo de búsqueda. El título es "Administrar ventas". Hay botones para "Crear venta" y "Rango de fecha". Un campo "Buscar:" y un selector "Mostrar 10 Registros". Una tabla de datos con columnas: #, Factura, Acciones, Vendedor, Precio venta, Precio compra, Neto, Total, Stock. Una fila seleccionada tiene un radio botón "Voucher" y un icono de impresión. Botones para "Anterior", "1" y "Siguiente".

Ilustración 34 Boceto administrar ventas



Este boceto muestra la interfaz de reportes de ventas. La barra superior tiene iconos para navegar entre páginas y un campo de búsqueda. El título es "Reportes de ventas". Hay un selector "Rango de fecha" y un botón "Descargar en Excel". Seis iconos representan diferentes tipos de informes: "Gráficos de ventas" (grafico de líneas), "Productos mas vendidos" (cuadro con una X), "Vendedores" (grafico de barras), y "Compradores" (grafico de barras).

Ilustración 33 Boceto reporte de ventas

7.5.5 Capturas de pantalla

A continuación, en las siguientes dos ilustraciones se muestran las interfaces de los gestores administración de ventas y generación de reportes de ventas; correspondientes a la entrega de la tercera iteración.

Administrador ventas



#	Código factura	Cliente	Vendedor	Forma de pago	Neto	Total	Fecha	Acciones
1	10001	Marcos	Administrador	Efectivo	\$ 426.00	\$ 430.26	2022-05-05 09:20:25	
2	10002	Christopher Paniagua	Administrador	Efectivo	\$ 41,462.20	\$ 41,462.20	2022-05-05 09:43:24	
3	10003	Christopher Paniagua	Administrador	Efectivo	\$ 525.00	\$ 525.00	2022-05-05 09:46:20	
4	10004	Marie Sanchez	Administrador	Efectivo	\$ 2,100.00	\$ 2,100.00	2022-05-05 09:47:17	

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Ilustración 35 Captura de pantalla administración ventas

Reportes de ventas

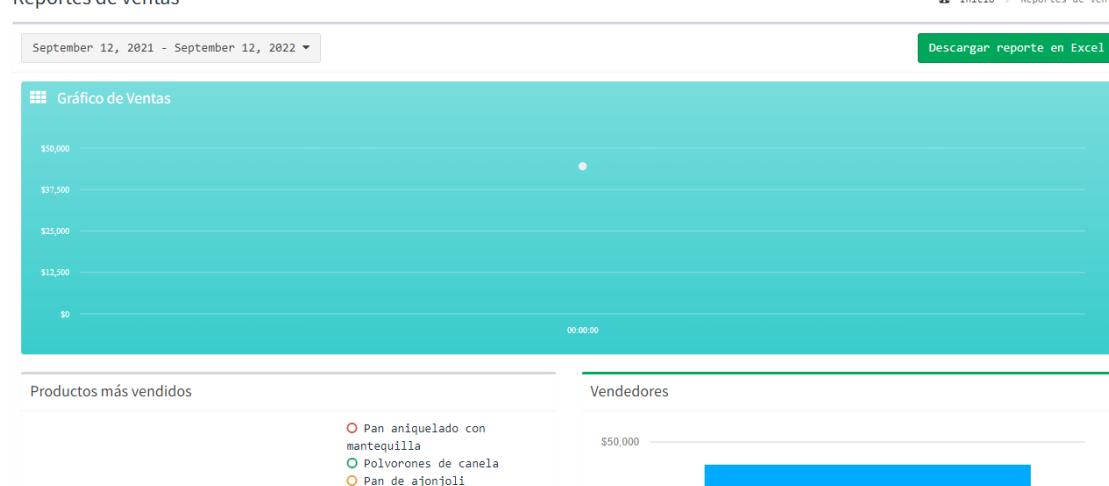


Gráfico de Ventas

September 12, 2021 - September 12, 2022

Descargar reporte en Excel

Productos más vendidos	Vendedores
● Pan aníquelado con mantequilla	\$50,000
● Polvorones de canela	\$37,500
● Pan de ajo y jengibre	\$0
● Pan dietético	\$0,00

Ilustración 36 Captura de pantalla generar reportes parte



I Ilustración 37 Captura de pantalla generar reportes parte II

7.5.6 Pruebas de aceptación

A continuación, en las tablas 76-78 se muestran las pruebas de aceptación que fueron empleadas para validar el éxito de la tercera iteración. Los indicadores que se definieron como criterios de evaluación fueron los siguiente.

Tabla 76 Pruebas de aceptación de la tercera iteración

Numero de prueba	Numero de historia	Nombre de la prueba
1	7	Administración de ventas
2	8	Generación de reportes

Tabla 77 Descripción de la prueba administración de ventas

Caso de prueba: Administración de ventas	
Código: 07	Numero de Historia de usuario: 7
Historia de Usuario: Administración de ventas	
Condiciones de ejecución: El usuario debe de ser de tipo administrador y deberá estar activo	
Entrada/Pasos de ejecución:	
<ul style="list-style-type: none">• Inicia sesión en el sistema• Accede al módulo o apartado administración de ventas• Agrega una venta desde este apartado• Completa el formulario• Registra la venta• Escoge como visualizar el registro de las ventas en un rango de fechas	

<ul style="list-style-type: none"> • En caso de que lo requiera puede imprimir la factura de la venta o visualizar un recibo en formato pdf.
Resultado esperado: La información de la venta registrada podrá estar disponible para que el operador manipule los datos de forma pertinente y a su vez pueda visualizarse en administración de ventas, por el rango que el usuario designe, también se podrá obtener facturas y recibos por cada registro de ventas.
Evaluación de la prueba: La prueba se completó satisfactoriamente.

Tabla 78 Descripción de la prueba de reportes

Caso de prueba: Reportes	
Código: 08	Número de Historia de usuario: 8
Historia de Usuario: Reportes	
Condiciones de ejecución: El usuario debe de ser de tipo administrador y deberá estar activo	
<ul style="list-style-type: none"> • Entrada/Pasos de ejecución: • Inicia sesión en el sistema • Accede al módulo o apartado reportes • Descarga el reporte en Excel • Puede visualizar el monto total de las ventas en un rango de fecha 	
Resultado esperado: Se obtiene el reporte de la administración de todas las ventas en un archivo de tipo Excel.	
Evaluación de la prueba: La prueba se completó satisfactoriamente.	

8. Métricas

Las métricas son un sistema de medición que sirven para cuantificar y evaluar aspectos de un negocio [29] de lo anterior podemos decir que hablar de métrica es referirse a una medida que describe un comportamiento o un resultado, son de gran importancia debido a que a

través de las métricas es que se puede evaluar el desempeño de cualquier acción en este caso de cualquier sistema informático , demostrando si el resultado es lo que se espera.

Se utilizó este sistema de métricas para evaluar el resultado del sistema web de inventario de productos existentes y control de ventas de la microempresa “Panaderia Hermanos Brenes” con el propósito de verificar si el sistema informático tiene la capacidad de responder positivamente las expectativas de la empresa, presentando resultados satisfactorios.

Se analizaron sus factores de éxito a partir de las historias de usuario, definiendo que para ser un resultado exitoso el sistema informático debe de:

- Ahorrar Papelería
- Ahorrar tiempo entre procesos.
- Usabilidad del Sistema Web
- Presentar la información de forma resumida y comprensible

8.1 Definición de métricas

Una vez identificados los indicadores que determinarán el éxito del uso del sistema web se definen las métricas de evaluación la aplicación, las cuales fueron las siguientes:

Desempeño

- Tiempo de respuesta
- Número de imprimibles

Usabilidad

- Interfaz fácil de navegar, diseño intuitivo y útil para el usuario
- Simplificar y sintetizar la información útil para el establecimiento

Las métricas de desempeño se obtuvieron mediante el empleo de una prueba comparativa con las herramientas actuales (reportes manuales - registros en papel) y el sistema informático web desarrollado, en el cual se registró el tiempo de respuesta y el número de impresiones que se necesitan para obtener información.

Las métricas de usabilidad se obtuvieron haciendo uso de una encuesta a los principales usuarios del sistema quienes son el administrador y los vendedores que laboran en el establecimiento, sobre el uso del sistema en sus tareas cotidianas.

8.2 Definición de los instrumentos

Se estructuro una prueba cuyo propósito es obtener información sobre actividades específicas de Panaderia Hermanos Brenes, dichas actividades son los procesos de reportes y pagos, por tal razón se definen los siguientes casos:

Caso de prueba 1: Obtener reporte de ventas del día

Caso de prueba 2: Obtener reporte de administración de ventas

En el caso de la encuesta se utilizó una única respuesta para cada pregunta realizada al personal, referente a los siguientes valores:

Tabla 79 Definición de los indicadores de evaluación

Indicadores de evaluación	Escala de valores
Interfaz fácil de navegar	Simple, Complejo
Diseño de interfaz intuitivo	Simple, Complejo
Comprensión en los procesos de las actividades	Si, No
Automatización en los procesos de las actividades	Si, No
Visualización de la información de forma resumida	Simple, Complejo
Agilidad y facilidad de obtención de reportes	Si, No

8.3 Resultado de la prueba

La prueba fue realizada a los usuarios principales del sistema, quienes son el administrador y los vendedores, asimismo los casos de prueba fueron realizados tanto con el método anterior como con la implementación del software.

Tabla 80 Resultados de caso de prueba 1

Caso prueba 1	Herramienta Anterior(A)		Herramienta Actual (B)		Ahorro (A-B)/A*100%	
	Tiempo de respuestas(S)	Número de reportes	Tiempo de respuestas(S)	Número de reportes	Tiempo(%)	Número de reportes(%)
1	800	1	60	1	93%	1
2	560	1	40	1	93%	1
Media	610	1	20	1	97%	1

8.3.1 Resultados caso de prueba 1

Para la realización de este caso de prueba con el método actual, se recopiló una a una las ventas actuales del día, buscándolas en el cuaderno de registro en donde cada venta es anotada, mientras que con el uso del sistema únicamente el usuario debe de acceder al apartado reporte de ventas y solicitar el reporte de ventas con el rango de fecha del día, finalmente exportarlo en formato Excel, formato que fue definido a conveniencia de la empresa.

La siguiente tabla detalla los resultados correspondientes al caso de prueba 1, reportados por el personal de la empresa involucrado. La siguiente fórmula muestra el procedimiento para calcular el ahorro porcentual:

Ahorro:

$Resultado\ Metodo_anterior - ResultadoMA$

$X = \underline{\hspace{100pt}} * 100$

$Resultado_metodo_anterior$

Tabla 81 Resultado caso de prueba 2

Caso prueba 2	Herramienta Anterior (A)		Herramienta Actual (B)		Ahorro (A-B)/A*100%	
	Tiempo de respuestas (S)	Número de reportes	Tiempo de respuestas (S)	Número de reportes	Tiempo (%)	Número de reportes (%)
1	510	1	65	1	87%	0%
2	400	1	50	1	88%	0%
Media	350	1	35	1	90%	0%

8.3.2 Resultado de caso de prueba 2

Para la aplicación de esta prueba con el método actual que utiliza la empresa, se siguen los pasos del caso de prueba 1.

En comparación a los resultados obtenidos tras la aplicación de ambas pruebas al realizar tareas diarias en la microempresa “Panaderia Hermanos Brenes”, se obtuvo el resultado que el uso del sistema informático web brinda un ahorro promedio del 89% respecto al método manual con el que actualmente cuenta la empresa.

8.4 Resultados de la encuesta

Se procedió a realizar una encuesta a los usuarios que hacen uso del sistema informático web, los cuales son: el administrador y los vendedores. Esta consulta consistió en 6 preguntas con el objetivo de evaluar el desempeño de ambos métodos (método anterior, método actual) obteniendo que la implementación del sistema web facilita la búsqueda de información, estructurándola y mostrándola de forma resumida y pertinente, de igual forma presenta los reportes de manera sistematizada mediante una interfaz fácil de navegar con un diseño intuitivo, siendo esta última la característica más importante para los usuarios del software.

Tabla 82 Resultado de la encuesta

Interfaz fácil de navegar	Diseño de interfaz intuitivo		Fácil navegación		Comprensión proceso de actividades		Visualización de reportes	
Encuestado	Sistema manual	Sistema informático	Sistema manual	Sistema manual	Sistema informático	Sistema informático	Sistema manual	Sistema informático
1	Complejo	Simple	Complejo	Complejo	Simple	Simple	Complejo	Simple
2	Complejo	Simple	Complejo	Complejo	Simple	Simple	Complejo	Simple
Moda	Complejo	Simple	Complejo	Complejo	Simple	Simple	Complejo	Simple

En el siguiente grafico se muestran de forma visual el resultado de la encuesta, la barra naranja representa el método actual, la barra azul representa el método anterior, la columna de la izquierda representa los indicadores en los que fue estructurada la encuesta.

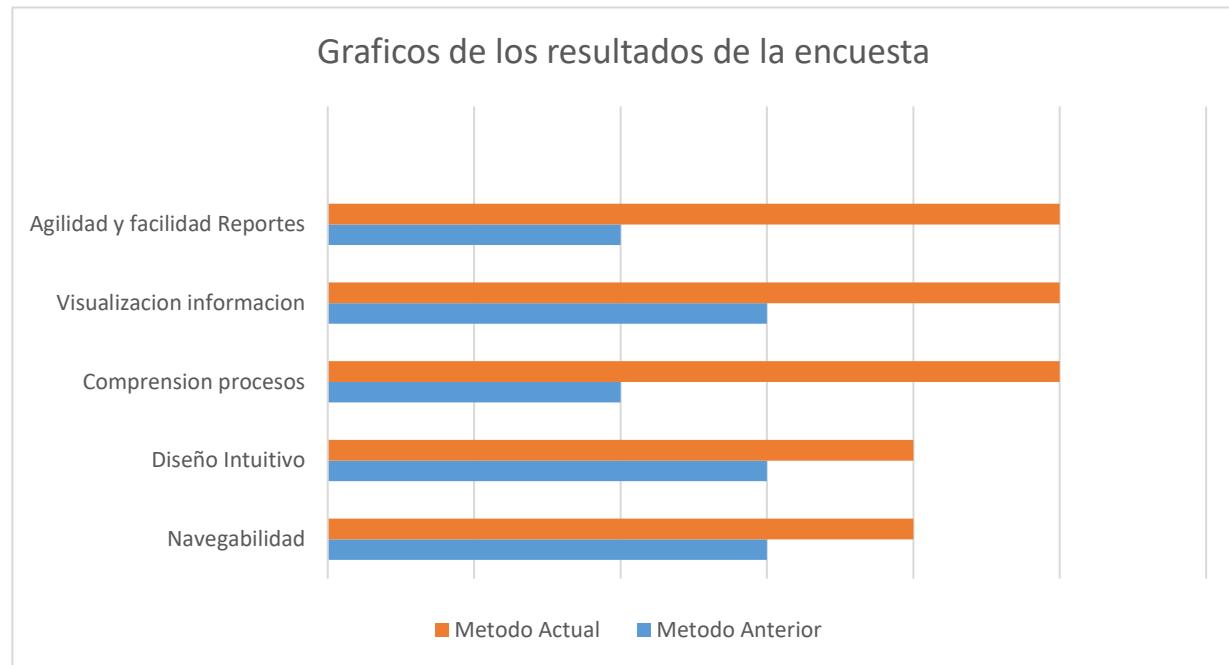


Ilustración 38 Gráficos de la encuesta

9. Conclusiones

En síntesis la implementación del sistema web para la gestión de inventario de productos y control de ventas para la microempresa panadería Hermanos Brenes logró optimizar los recursos, costos y esfuerzos del establecimiento, debido que los procesos internos involucrados en la gestión de inventario de productos existentes y control de ventas fueron en su mayoría automatizados, lo que vino a mejorar significativamente los procesos relacionados con la eficiencia de las ventas ya que se disminuye el tiempo de búsqueda de la información requerida, cabe señalar que el sistema informático aún no se encuentra en línea, está en modo local (intranet).

Definitivamente fue mediante la incorporación de la metodología Xp que se obtuvo un resultado satisfactorio, a través de pruebas constantes y ciclos de desarrollo cortos se simplificó el trabajo de desarrollo. Por otra parte, con el análisis de las historias de usuario se determinaron los requerimientos para desarrollar el sistema informático, en consecuencia, se definieron las tareas de ingeniería para cada entrega de iteración.

Una vez identificado los requerimientos se diseñó la base de datos lo cual permitió profundizar y diferenciar los procesos principales de los procesos secundarios que estarían presentes en el diseño de las interfaces. Por último, se codificó la solución dando origen a una potencial herramienta de trabajo, porque al final esa es la mejor forma para hacer referencia a un sistema informático.

Finalmente se realizaron pruebas de aceptación para verificar la funcionalidad del sistema, una vez aprobadas las pruebas con éxito se implementó el sistema informático en el local, adicionalmente puedo decir que tras el uso del sistema se han agilizado las operaciones diarias logrando los resultados esperados por parte del cliente.

10. Recomendaciones

Se recomienda a la empresa tomar en cuenta lo siguiente:

- Realizar mantenimiento periódico al sistema informático por posibles mejoras.
- Capacitar acerca del uso del sistema a los operadores de este.
- Proporcionar el manual de usuario de extensión .PDF a cada uno de los operadores del sistema informático.
- El área de sistema es la encargada de realizar respaldos en DB una vez por semana en un disco duro externo, manteniendo tres copias de archivos, la original y dos respaldos, esto con la finalidad de evitar perdida de información.
- Proteger todos los equipos que estén en red manteniendo el antivirus actualizado simultáneamente gestionando y controlando que las conexiones a la red sean seguras.

11. Referencias bibliográficas

- [1] S. D.H, Informática presente y futuro, Mexico: McGraw Hill, 1991.
- [2] E. & P. Caldentey, Administracion de inventario, Ecuador, 2016.
- [3] M. G. C.-M. V. J. A. C. B. J. & F. S. L. Piattini Velthuis, Análisis y diseño detallado de aplicaciones informáticas de gestión, Mexico DF: Alfaomega Grupo Editor-Rama., 2000.
- [4] Escuela superior politecnica del litoral, de *Programación, Lenguajes. Lenguajes de programación.*, 2009.
- [5] F. a. A. J. a. A. J. a. S. M. Sierra, «analisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web,» *Investigacion y desarrollo en TIC*, pp. 14-26, 2013.
- [6] L. Lopez, Programacion Orientada a Objetos, Mexico: Alfaomega Grupo Editor, SA, 2006.
- [7] D. P. Alonso, Evaluación de la accesibilidad en software generado por un entorno de desarrollo integrado, 2005.
- [8] Project Management Institute, PMBOOK GUIDE, Newtown Square, Pennsylvania, 2017.
- [9] E. G. Burgués, Aprende a Modelar Aplicaciones con UML: 2^a Edición, IT Campus Academy, 2016.
- [10] Á. Cobo, PHP y MySQL: Tecnología para el desarrollo de aplicaciones web, Brasil: Ediciones Díaz de Santos, 2005.
- [11] A. Lazalde, «Hipertextual.com,» 07 febrero 2011. [En línea]. Available: <https://hipertextual.com/2011/02/7-tecnologias-de-software-que-sostienen-a-facebook>.
- [12] J. D. Gauchat, El gran libro de HTML5, CSS3 y Javascript, Marcombo, 2012.

- [13] Y. T. & Luján-Mora, «iDESWEB,» 2013. [En línea]. Available: www.iDESWEB.com.
- [14] A. Durango, Diseño Web con CSS: 2^a Edición. IT, Campus Academy., 2015.
- [15] J. D. Gauchat, El gran libro de HTML5, CSS3 y Javascript, Marcombo, 2012.
- [16] E. Haughee, Instant Sublime Text Starter, Packt Publishing., 2013.
- [17] R. A. T. Schutta, Introducing Ajax, 2006.
- [18] © 2021, Oracle Corporation y / o sus afiliadas, «MySQL.com,» Enero 2021. [En línea]. Available: <https://www.mysql.com/products/enterprise/>.
- [19] K. H. & Silberschatz, Fundamentos de bases de datos, Madrid: Madrid, 1993.
- [20] IDESWEB UA/Armando Suárez Cueto., «Curso "Introducción al desarrollo web",» [En línea].
- [21] M. C. P. P. L. José H. Canós, «Metodologias Agiles en el desarrollo del software,» 2012. [En línea]. Available: canos2012metodologias.
- [22] P. & P. M. C. Letelier, Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)., Técnica Administrativa issn: 1666-1680, 2006.
- [23] P. a. C. Letelier, «Metodologias agiles para el desarrollo de software: eXtreme Programming (XP),» 2012. [En línea]. Available: Metodologiasletelier.com.
- [24] J. C. Garrido, «Arquitectura y diseño de sistemas web modernos. InforMAS,» *Revista de Ingeniería Informática del CIIIRM*, 1., pp. 10-24, 2004.
- [25] C. Alfaro, «Designlopers,» 22 marzo 2019. [En línea]. Available: <https://designlopers.com/post/Desarrollo-de-aplicaciones-profesionales-en-PHP-y-MVC/>.

- [26] H. A. Johnson, «Meed Libr Assoc,» 2017. [En línea]. Available: www.ncbi.nlm.nih.gov/pmc/articles/PMC5370621/.
- [27] M. A. Cetina, Gestion de procesos con BMP Tecnologia Investigacion y Academia, 2016.
- [28] Bizagi, «Bizagi Modeler,» [En línea]. Available: <https://www.bizagi.com/es/plataforma/pruebe-modeler>.
- [29] workmeter world, «Marketing Digital,» 2002. [En línea]. Available: <https://rockcontent.com/es/blog/metricas-de-mercadotecnia/>.
- [30] L. J. & J. L. Herrera, «Ingenieria De Requerimientos i,» 02 Dicimebre 2012. [En línea]. Available: <http://www.ingenieria.com/trabajos6/resof/resof.shtml>, 15.
- [31] E. P. Explained, Klint Beck.
- [32] J. L. Vilagrau, «La Metodología XP: la metodología de desarollo de software más existosa,» Julio 2016. [En línea]. Available: <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/>.

12. Anexos

Anexo A: Estudio de factibilidad

En definitiva, antes de comenzar cualquier proyecto informático es de vital importancia realizar un estudio de factibilidad con la intención de evaluar y determinar la viabilidad de dicho proyecto, esto se realiza mediante el análisis de cuatro puntos de importancia: técnico, económico, operativo y finalmente legal.

Factibilidad técnica

Considerando que el enfoque del desarrollo de este sistema web es orientado a objetos es ideal garantizar un conjunto de servicios previamente definidos con el propósito de precisar el entorno de trabajo necesario para la funcionalidad del proyecto.

En la factibilidad técnica se establecen las condiciones para llevar a cabo el proyecto informático, esta permite conocer los recursos ideales para la correcta implementación de la solución presentada, no obstante, se evalúa la disponibilidad de los recursos existentes dentro de la empresa en términos tanto de hardware, software como de recursos humanos.

El presente proyecto ha sido encontrado factible debido a que la Panadería Hermanos Brenes posee la infraestructura informática necesaria para la implementación del proyecto a desarrollar: una conexión estable de internet de ancha banda de 50 Mbps, 3 equipos portátiles y una impresora.

A su vez cuenta con el personal necesario para operar el sistema: un técnico de computación, quien posteriormente se encargará de darle mantenimiento al sistema, un administrador, quien operará y tendrá acceso a todas las funcionalidades del sistema y finalmente 3 vendedores quienes harán uso parcial del sistema, de manera que la empresa puede responder de forma favorable al desarrollo de esta solución.

Características del Hardware disponible para el desarrollo

Equipos	Elementos	Especificaciones	Capacidad
Laptop 1	1	Procesador Intel Memoria RAM Disco duro Monitor pantalla completa	Corei5 4GB 500 GB Generic pnp monitor 15"
Laptop 2	1	Procesador Intel Memoria RAM Disco duro Monitor pantalla completa	Corei3 2GB 500 GB Generic pnp monitor 15"
Laptop 3	1	Procesador Intel Memoria RAM Disco duro Monitor pantalla completa	Corei3 2GB 500 GB Generic pnp monitor 15 "
Impresora laser	1	Dell 1720 mn	Salida monochrome Interfaz USB
Router	1	Servicio de internet	50 Mbps
Cartucho de Tóner negro	1	Cartucho EPSON de tóner	Monocromático
Silla secretaria	3	Silla ergonómica	para oficina
Escritorio para laptop	3	Escritorio plegable para laptop	estilo ejecutivo

Tabla 83 Aspectos técnicos del hardware

Dentro de las características del hardware disponible para el correcto funcionamiento de la aplicación web se consideraron las anteriormente presentadas en la tabla:

Características técnicas de la capacidad de desarrollo del proyecto

Una vez garantizada la infraestructura informática se comprueba la capacidad técnica para llevar a cabo el desarrollo del proyecto, esto quiere decir que teniendo en cuenta los aspectos técnicos del hardware se muestra la capacidad técnica para desarrollar el proyecto.

Sistema Operativo

El sistema operativo corresponde a uno de los elementos principales debido a que es el encargado de coordinar y dirigir todos los demás programas en una computadora cumpliendo características primordiales al momento de desarrollar e implementar un sistema: estabilidad, facilidad de uso, seguridad, multitareas, no obstante, soportado el entorno de desarrollo e instalado el sistema informático el sistema operativo deberá brindar velocidad de conexión entre base de datos, sistema y usuarios.

Lenguaje de desarrollo

Los lenguajes de desarrollo corresponden una serie de instrucciones y procedimientos proporcionando diversos servicios insustituibles que facilitan al programador el desarrollo del software. Las características principales que deben cumplir son las siguientes:

- Facilidad de administración de sistemas
- Soporte de base de datos
- Herramientas de entorno .net
- Mejora continua

Sistema Gestor

Este software determinara la manera en que se procese la información, la velocidad del procesamiento, la seguridad y finalmente el respaldo de los datos, debe de ser escalable, estable, en mejora continua y conectable con diferentes lenguajes de programación del entorno de desarrollo.

Equipo de desarrollo

El equipo de desarrollo son personas que disponen del conocimiento y las capacidades requeridas para llevar la codificación, implementación y soporte técnico del sistema, cuentan con diferentes habilidades que aportan al cumplimiento de los objetivos previamente definidos antes de realizar el sistema informático.

Tabla 84 Especificaciones del software

Especificaciones software	
Ámbito Técnico	Detalle
Plataforma de procesamiento del sistema	Sistema operativo Windows 10
Gestor de Base de Datos	MySQLServer
Lenguajes de desarrollo	Php Ajax JSP ASP.NET
Equipo de Desarrollo	Dos técnicos informáticos Dos programadores dedicados al desarrollo del sistema.
Conocimientos	Lenguajes de programación.net Modelado de base de datos SQL

	Base de Datos relacionales Modelados de diagramas Metodología XP
--	--

Conclusión de la Factibilidad técnica

En contraste con lo anterior se concluye que se dispone tanto de recursos humanos calificado para llevar a cabo la codificación y mantenimiento del sistema como la capacidad técnica para desarrollarlo. Finalmente, la microempresa Panadería Hermanos Brenes posee los requisitos técnicos necesarios para concluir con éxito el proyecto, en otras palabras, es técnicamente idóneo tanto a nivel de Hardware como a nivel de Software, brindando los resultados deseados.

Factibilidad Económica

En la factibilidad económica se determinan los costos de los recursos de la factibilidad técnica también se evalúan los beneficios económicos que conllevara la realización del proyecto. En dicha factibilidad se establecen todos los costos relacionados al desarrollo del proyecto desde tres enfoques informáticos de importancia: hardware, software y recursos humanos, estableciendo la comparación de los costos-beneficios del sistema manual y proyectando la relación de los costos-beneficios del sistema informático.

Descripción del sistema manual

Para ilustrar la situación actual del sistema con el que la empresa está trabajando se utiliza un diagrama Ishikawa. Los diagramas de Ishikawa identifican y clasifican las causas de la problemática y el efecto de esta.



Ilustración 39 Descripción del sistema manual

Haciendo un análisis del diagrama anterior podemos decir que existen múltiples limitaciones en el sistema manual con el que cuenta la microempresa, cabe destacar la existencia de recursos económicos innecesarios, el mal uso y la incorrecta distribución de algunos de estos, asimismo la baja productividad en el personal involucrado en la utilización de este sistema. Para hacer efectivo lo anterior se establecen los costos del sistema actual detallando precios:

Tabla 85 Salario del personal involucrado en el sistema manual

Cargo	Cantidad Personas	Salario/Mes	Salario/Día	Salario/Horas	Horas trabajadas	Días trabajados	Horas Extras	Costo H.E	Total
Técnico Informático/eléctrico	1	C\$15,000	C\$625	C\$78.12	8	6	-	-	C\$15.000
Encargado de atención al cliente	3	C\$10,000	C\$416	C\$52.08	8	6	7	729.09	C\$10,729
Administrador	1	C\$12,000	C\$500	C\$62.5	8	6	8	1,000	C\$13,000

Tabla 86 Costos mensuales sistema manual

Elementos	Cantidad	Precio Unitario	Total
Papel Bond Resma de papel tamaño carta	2	C\$200.00	C\$400.00
Libro diario de contabilidad	1	C\$300.00	C\$300.00
Lapiceros	2	C\$15.00	C\$30.00
Tóner tinta monocromático	1	C\$750.00	C\$750.00

Dada la anterior tabla ahora se presenta una tabla total del costo mensual que implica el uso del sistema manual de la empresa.

Tabla 87 Costos totales sistema manual

Especificación del recurso	Monto
Recursos Humanos	C\$38,729
Materiales	C\$1,480
Servicio de Internet	C\$1,400

El costo mensual total que conlleva la utilización del sistema manual con el que cuenta la microempresa es de C\$41,609 córdobas, partiendo de estos resultados se presentan los costos que implica el sistema informático.

Tabla 88 Salario del personal involucrado en el sistema informático

Cargo	Cantidad Personal	Salario/Mensual	Salario/Día	Salario/Horas	Horas trabajadas	Días trabajados	Horas Extras	Costo H.E	Total
Técnico Informático/eléctrico	1	C\$15,000	C\$625	C\$78.12	8	6	-	-	C\$15,000
Encargado de atención al cliente	3	C\$10,000	C\$416	C\$52.08	8	6	7	-	C\$10,000
Administrador	1	C\$12,000	C\$500	C\$62.5	8	6	8	-	C\$12,000

Tabla 89 Descripción de los costos de recursos

Especificación del recurso	Monto
Recursos Humanos	C\$37,000
1 tóner tinta monocromática	C\$750
Servicio de Internet	C\$1,400

El costo mensual que implicaría el sistema informático es de 39,950 córdobas, como se observa la diferencia es notoria, por otra parte, es necesario destacar que en los costos del sistema informático variaría el plazo de cambio del tóner ya que se dará cada dos meses, esto último a razón de que el nuevo sistema proporciona un reporte de tipo Excel en tiempo real, reduciendo así las fotocopias de los archivos de registro: Un archivo suele tener tres copias, una copia para el dueño de la Panadería Hermanos Brenes, otra para el administrador y otra que queda a disposición de los operadores de atención al cliente.

A continuación, se calculan los costos mensuales comparando los dos sistemas: el sistema con el que actualmente cuenta la empresa, el sistema manual y el sistema informático a implementar, por otra parte, se aclara que se mantienen los costos a futuro en otras palabras están definidos como valores estáticos , dicho de otra forma no se toma en cuenta el aumento o disminución de los costos a lo largo del año, tomando en consideración lo anterior mencionado conviene destacar que para el análisis de este estudio de factibilidad se tomaron en cuenta dos variables: la variable de impacto y la variable de resultado.

La variable de impacto es aquella que genera un resultado a corto plazo, así como se muestra en las tablas del costo mensual, de forma similar es la variable de resultado con la diferencia que aquí se procesa el costo mensual obtenido por un plazo de tiempo determinado, puede ser semestral o anual, con el propósito de obtener una demostración más exacta de la valoración del análisis de los costos.

El presente análisis es un aproximado de la comparación de costos de ambos sistemas, por el tiempo de un año, obteniendo como resultado la siguiente proyección,

Tabla 90 Costos sistema manual

Numero de meses	1	2	3	4	5	6	7	8	9	10	11	12
Costo mensual en córdobas	41,609	41,609	41,609	41,609	41,609	41,609	41,609	41,609	41,609	41,609	41,609	41,609
Costo mensual en dólares	1163	1163	1163	1163	1163	1163	1163	1163	1163	1163	1163	1163
Costo total anual en dólares	13,956											
Costo total anual en córdobas	499,308											

Tabla 91 Costos sistema informático

Numero de meses	1	2	3	4	5	6	7	8	9	10	11	12
Costo mensual en córdobas	39,150	37,000	37,000	39,150	39,150	37,000	37,000	39,150	37,000	37,000	39,150	37,000
Costo mensual en dólares	1090	1030	1030	1090	1030	1030	1090	1030	1030	1090	1030	1030
Costo total anual en dólares	12,600											
Costo total anual en córdobas	452,360											

Tabla 92 Comparación de costos de ambos sistemas

Costo anual-sistema manual C\$	\$13,956
Costo anual-sistema manual C\$	C\$499,308
Costo anual-sistema informático \$	\$12,600
Costo anual-sistema informático C\$	C\$452,360
Diferencia de costos C\$	\$1,356
Diferencia de costos C\$	C\$46,938

Conclusión de la factibilidad económica

De tabla anterior se concluye que el sistema informático es económicamente factible debido a que ahorra aproximadamente un 10% de recursos económicos en términos de un año comparado con el sistema actual, notándose los resultados desde el primer mes de implementación.

Factibilidad operativa

La factibilidad operativa es fundamental para las empresas que fabrican productos o prestan servicios ya que todo proceso de producción debe ser realizable, por otra parte, la eficiencia en la gestión del tiempo o de los costos laborales también es fundamental, ya que cuando se presta un servicio implica que tan adecuadamente se cumple una función en relación a las variables costo y tiempo.

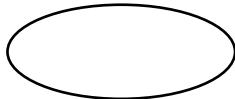
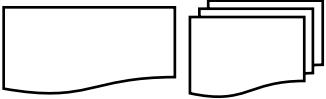
Antes de proceder a implementar cualquier solución informática, es de vital importancia establecer una relación entre el sistema informático y la organización que lo ejecutara. La factibilidad operativa es la más importante de todas las factibilidades ya en esta se realiza un análisis en detalle del estudio de las necesidades del negocio y si estas necesidades pueden ser satisfechas con la solución propuesta, además de medir a que grado el sistema propuesto resuelve el problema y que tantos beneficios tangibles conlleva su uso.

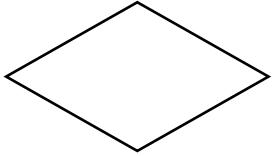
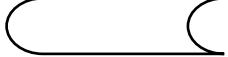
Cuando se presta un servicio, conviene desglosarlo en pasos o etapas y calcular el tiempo y el coste óptimos que implica la realización de este servicio. Para ilustrar este proceso de pasos y etapas se hace uso de un diagrama de flujo.

Los diagramas de flujo son una herramienta que representa el funcionamiento del cualquier trabajo como un proceso y determinar dónde está el problema: “cada proceso realiza una serie de pasos, cualquier paso que realice un proceso obtiene un resultado” [8], dicho lo anterior es que se dice que un cada necesidad se convierte en una tarea, una tarea posteriormente se convierte en un proceso, un proceso es un numero de funciones que como resultado puede generar información, productos, servicios o una combinación de todos estos ellos.

Los diagramas de flujo usan símbolos, a cada símbolo le corresponde un significado previamente definido como se mostrará en la siguiente tabla:

Tabla 93 Símbolos de los diagramas de flujo

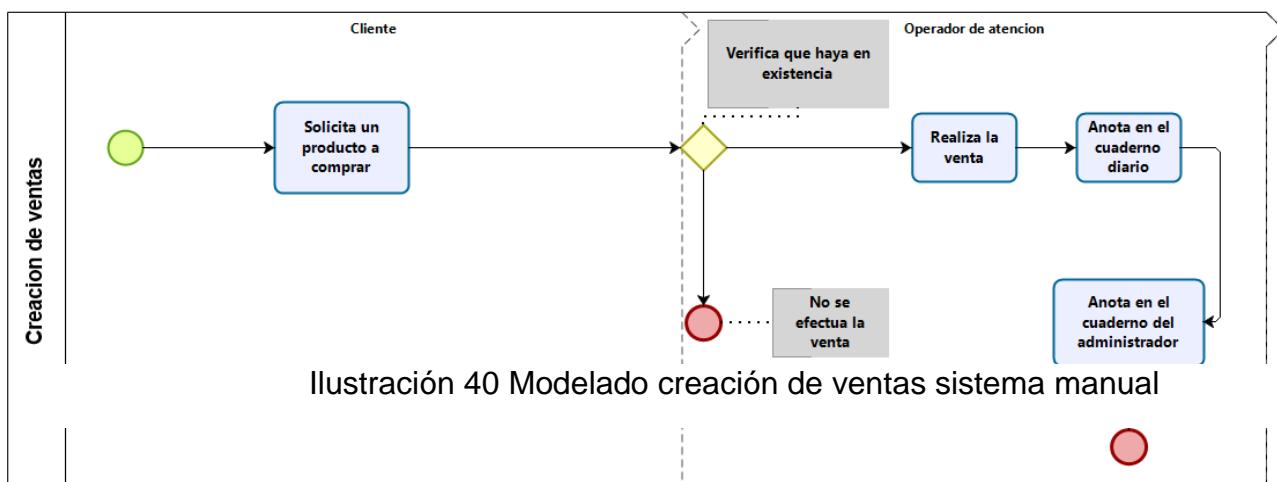
Símbolo	Nombre	Actividad
 Ovalo	Terminador	Se utiliza en el inicio (materiales, información o acción que inicia el proceso) y el final (resultados) del proceso.
 Rectángulo o cuadrado	Proceso	En este símbolo se establece la actividad desarrollada en el proceso
 Figura de entrada manual	Entrada Manual	Este símbolo indica que existe entrada de datos u otros objetos al proceso
 Figura de documento y multidocumentos	Figura de documento y multidocumentos	Mediante este símbolo se establecen la cantidad de documentos y copias de los mismos que se requieren, se generan o se consultan en el proceso

 <p>Diamante</p>	<p>Decisión</p>	<p>Muestra los momentos dentro del proceso en los que una pregunta se hace (si/no) o una decisión es requerida; y determinará el flujo a seguir dentro del proceso.</p>
 <p>Datos almacenados</p>	<p>Datos Almacenados</p>	<p>Este símbolo es utilizado para representar que en momentos determinados del proceso existe la acción de guardar o almacenar cierto objeto</p>

El sistema web de gestión de inventario y control de ventas de la microempresa “panadería hermanos Brenes” estará constituido por los siguientes componentes principales:

- Usuarios
- Gestor de Base de datos
- Aplicación web
- Sistema operativo
- Hardware
- Manuales de Usuario
- Manual técnico del sistema
- Reportes

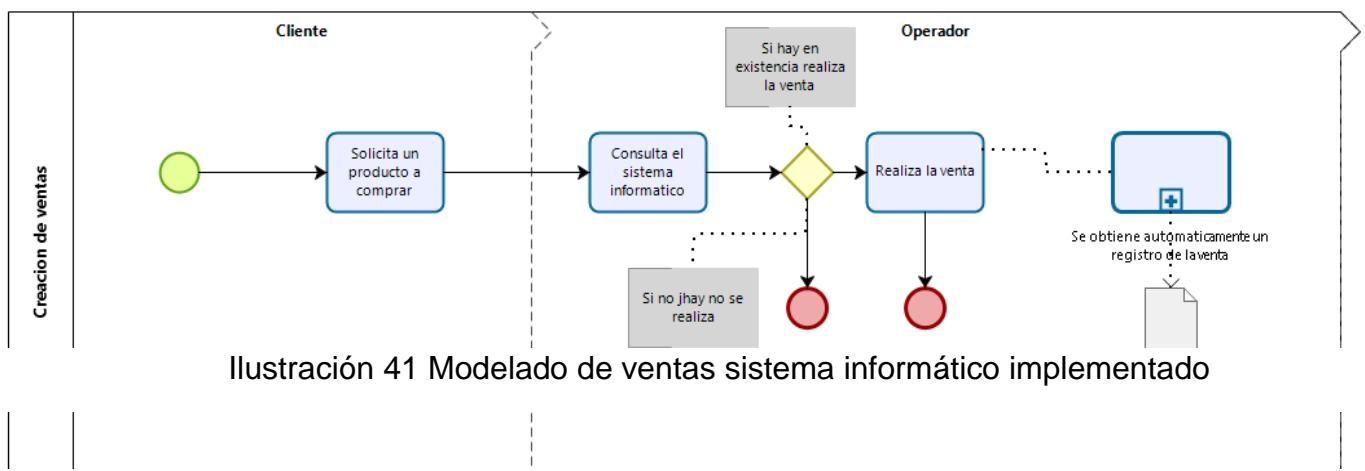
Antes de comenzar a ilustrar los diferentes procesos de trabajo que se realizan dentro de la microempresa primeramente se explica el mecanismo del trabajo. En síntesis, la principal actividad es la creación de una venta, posteriormente se gestiona cada una de las ventas realizadas en el día a manera de reporte (cabe destacar que otra función principal es verificar el suministro de los productos existentes) con el propósito de mantener un control de las entradas y salidas que realiza el establecimiento. Una entrada es una compra, una salida es una venta. A continuación, se ilustra el proceso de negocio de la creación de una venta con el mecanismo manual que utiliza la empresa, seguidamente del sistema informático.



establecimiento que solicita productos, el cliente se presenta dispuesto a realizar una compra, el vendedor atiende la consulta del cliente, revisa los suministros en existencia de forma manual, si hay en existencia lo que el cliente busca se efectúa la venta, una vez realizada la venta manualmente se registra la venta en dos libros diferentes, uno de diario que utilizan los vendedores y otro que utiliza el administrador o dueño del local.

El siguiente modelado de procesos representa la creación de una venta una vez implementado el sistema informático:

Un cliente solicita comprar uno o más productos, el operador verifica en el sistema informático si hay en existencia, en caso que haya en existencia



se efectúa la venta y automáticamente esa venta se añade al registro de ventas total dicho registro estará disponible para todos los usuarios del sistema web, así como también un reporte de todo el registro de ventas.

Anexo B: Ingeniería de requerimientos

Ingeniería de requerimientos

La ingeniería de requerimiento es un proceso sistemático y disciplinado para la especificación y administración de una necesidad de negocio [30], ciertamente un requerimiento define la funcionalidad que se espera que tenga un determinado sistema informático. Un requerimiento expresa lo que el cliente quiere, a su vez representa una tarea programable con procesos automatizables y finalmente representa un resultado esperado por el usuario final.

Requerimientos funcionales

Los requerimientos de un sistema informático se dividen en funcionales y no funcionales, los funcionales se refieren a un resultado en particular del sistema cuando el usuario realiza una tarea en ellos. A continuación, se definen los requerimientos funcionales que tendrá el sistema informático y la prioridad que preside a cada uno:

Tabla 94 Requerimientos funcionales del sistema

Número	Requerimiento	Prioridad
01	El sistema permite a los usuarios autorizados ingresar al sistema informático mediante un login	Alta
02	El sistema permite registrar diferentes tipos de usuario	Alta
03	El sistema permite administrar los diferentes tipos de usuarios	Alta
04	El sistema permite asignar permisos a los diferentes tipos de usuario	Alta
05	El sistema mantiene la información de los diferentes tipos de usuario	Alta
06	El sistema mantiene el registro de los clientes	Alta

07	El sistema permite registrar las diferentes categorías de pan existente en la Panadería	Alta
08	El sistema permite llevar un control de inventario de los productos existentes en la Panadería	Alta
09	El sistema permite registrar a los clientes guardando la información pertinente	Alta
10	El sistema permite visualizar la información básica de los clientes	Alta
11	El sistema permite registrar la información de las ventas efectuadas	Alta
12	El sistema permite llevar un inventario del stock de productos existentes	Alta
13	El sistema permite visualizar el stock de productos existentes	Alta
14	El sistema permitirá gestionar la información de las ventas (actualizar, eliminar, buscar, registrar)	Alta
15	El sistema tendrá que generar recibos tras una transacción de ventas	Alta
16	El sistema deberá administrar las ventas por rangos de fechas	Alta
17	El sistema deberá generar un reporte de ventas exportable en un archivo Excel	Alta
18	El sistema deberá de mostrar los productos más vendidos así como los vendedores y clientes más destacados	Alta

Requerimientos no funcionales

Por otro lado, el requisito no funcional da el comportamiento general del sistema o su componente y no en función, es decir, un requisito no funcional aborda 'qué debería ser un sistema' en lugar de 'qué debería hacer un sistema' (requisito funcional). A continuación, se definen los requerimientos no funcionales que tendrá el sistema informático, también la prioridad que preside a cada uno de ellos y las métricas utilizadas al momento de definirlos.

Tabla 95 Requerimientos no funcionales del sistema

Numero	Requerimiento	Prioridad	Métrica
01	El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña	Alta	Seguridad de Datos
02	El sistema enviará alertas de intentos fallidos cuando se ingrese la contraseña o el usuario de forma incorrecta	Alta	Seguridad de Datos
03	El sistema presenta una interfaz agradable al usuario	Alta	Usabilidad
04	El sistema estará disponible 24/7	Alta	Disponibilidad
05	El sistema permitirá el acceso remoto con solo iniciar con las credenciales autorizadas	Alta	Disponibilidad
06	El sistema deberá de responder a las	Alta	Rendimiento

	peticiones del usuario en un tiempo de respuesta prudente		
07	El sistema permite hacer uso de la información previamente registrada	Alta	Integridad de datos
08	Los procesos presentes en el sistema son ampliables	Alta	Escalabilidad
09	Las diferentes interfaces del sistema comparten información entre ellas	Alta	Interoperabilidad
10	El sistema deberá procesar transacciones de múltiples ventas	Alta	Eficiencia
11	El sistema deberá de contar con manuales de usuario documentados adecuadamente	Alta	Usabilidad
12	El sistema contara con manuales técnicos para los programadores que le darán mantenimiento	Alta	Escalabilidad
13	El sistema deberá de iniciarse con la información y los elementos visuales con rapidez	Alta	Rapidez

Anexo C: Manual de usuario

1. Introducción al manual de usuario

En el siguiente documento se explica paso a paso y de forma gráfica la forma en cómo usted puede utilizar el sitio web Panaderiapro, además de todas las opciones que se encuentran disponibles para su uso y trabajo.

Esperemos que su experiencia navegando a través de esta página sea de lo más grata y logre satisfacer completamente sus expectativas y necesidades de información.

Panadería Hermanos Brenes, es una empresa destinada a la producción y comercialización de pan, donde a través de los años han satisfecho con las necesidades de sus clientes ofreciendo siempre productos de la más alta calidad a los mejores precios, como también brindan servicio de calidad que los distingue y los coloca en la primera posición en el gusto del público.

2. Objetivos del manual de usuario

Objetivo General

Capacitar a los usuarios finales sobre las características y funcionamiento del sitio web.

Objetivos Específicos

- Proporcionar al usuario la información necesaria para utilizar el sistema.
- Detallar paso a paso el funcionamiento del sistema.

3. Vista General del sitio web

En la página principal aparecen los accesos a las diferentes ventanas de la página web:

1. Inicio: Desde aquí se podrá acceder a las diferentes ventanas del sitio web.
2. Usuarios: Muestra toda la información sobre los diferentes tipos de usuarios los cuales utilizaran el sitio web.
3. Categorías: Muestra la información de las diferentes categorías.
4. Productos: Muestra toda la información acerca de los productos disponibles en la panadería.
5. Clientes: Muestra toda la información sobre los clientes de dicho negocio.
6. Ventas: Aquí se lleva un control de productos vendidos.

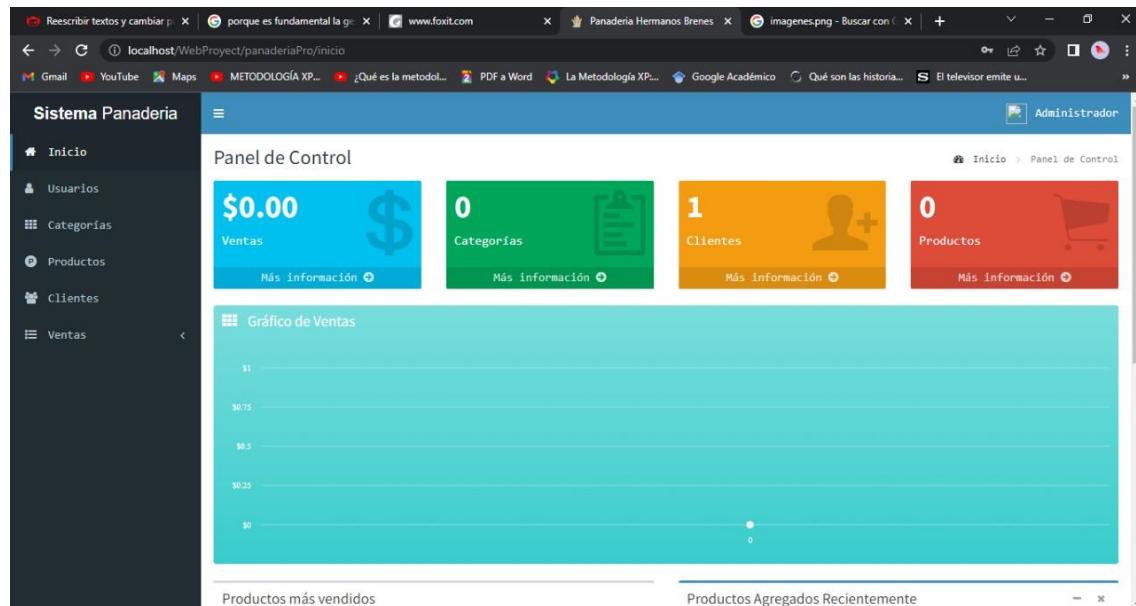


Ilustración 42 Vista General del sitio web

4. ¿Cómo acceder?

Para acceder al sitio web primeramente tenemos que abrir el navegador y en la barra de tareas escribiremos lo siguiente localhost/panaderiapro/ y presionamos enter para acceder.

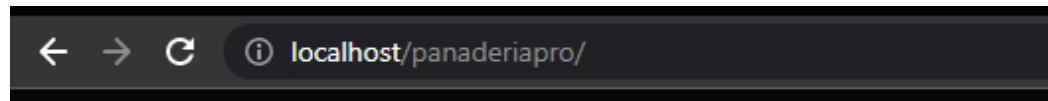


Ilustración 43 Acceso

Luego de acceder nos aparecerá la primera ventana que es la del Login en donde llenaremos los campos y daremos clic en ingresar.

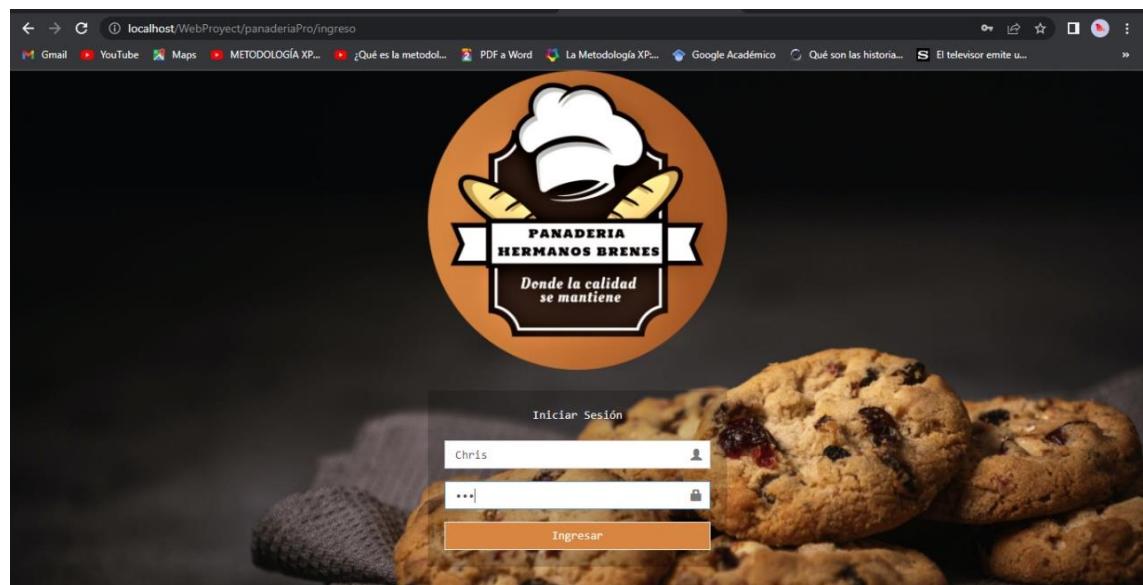


Ilustración 44 Inicio sesión

Al acceder con tu usuario, la siguiente ventana que aparecerá será la de bienvenida.

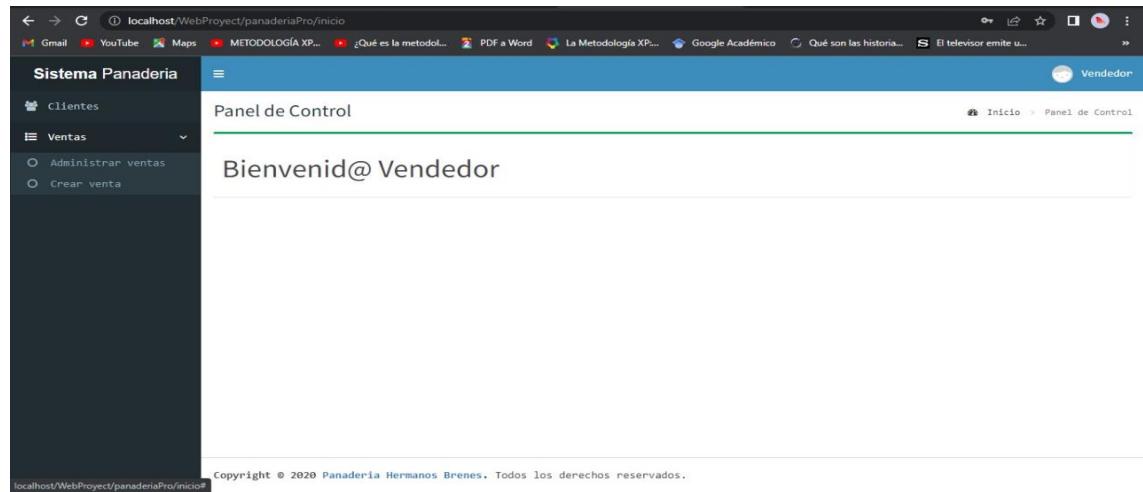


Ilustración 45 Vendedor especial

5. Inicio

Luego accedemos a la página de inicio, la cual muestra una vista general en donde aparece un panel de control de los diferentes módulos de la página web.

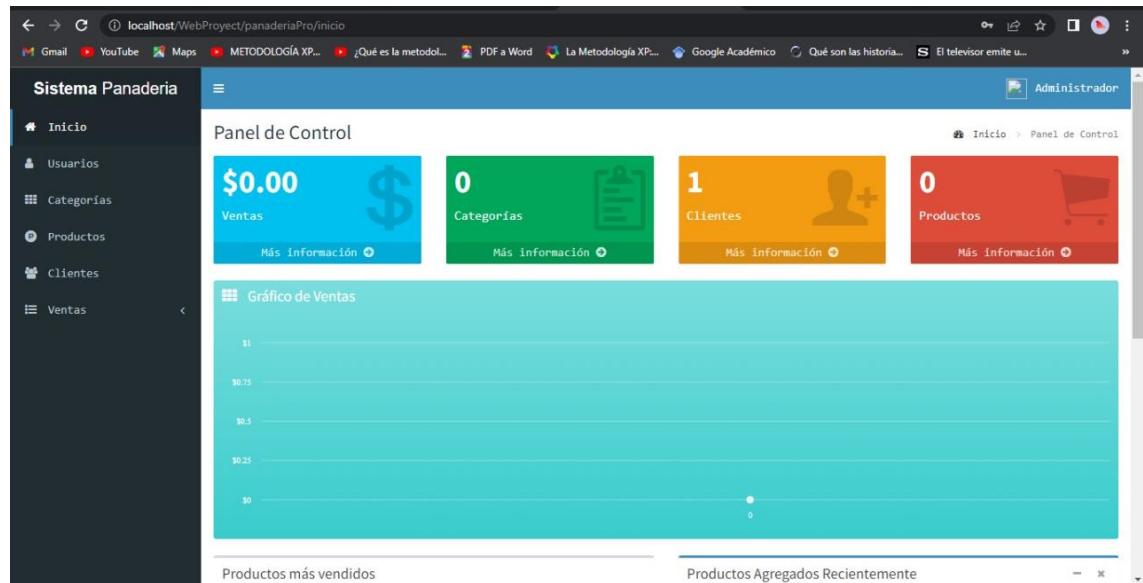


Ilustración 46 Inicio

6. Usuarios

La ventana de usuarios nos muestra toda la información de los usuarios disponibles en el sitio web, desde aquí el administrador podrá agregar, modificar y eliminar usuarios.

#	Nombre	Usuario	Foto	Perfil	Estado	Último login	Acciones
1	Administrador	admin		Administrador	Activado	2020-08-13 02:30:07	
2	administrador	123		Administrador	Activado	2020-08-13 02:44:58	
3	sofie	sofie		Administrador	Activado	2020-08-14 06:37:21	
4	Elie	Elie		Administrador	Activado	2022-04-30 01:08:58	

Ilustración 47 Usuarios

7. Perfiles de usuarios

A cada uno de los usuarios se les asignara un perfil de 3 existentes los cuales son:

1. Administrador.
2. Vendedor.
3. Especial.

a. Administrador

El administrador es el perfil principal del sistema, este usuario podrá visualizar y manejar todas las funciones que hay en el sistema web, el será el único que podrá agregar, modificar y eliminar usuarios.

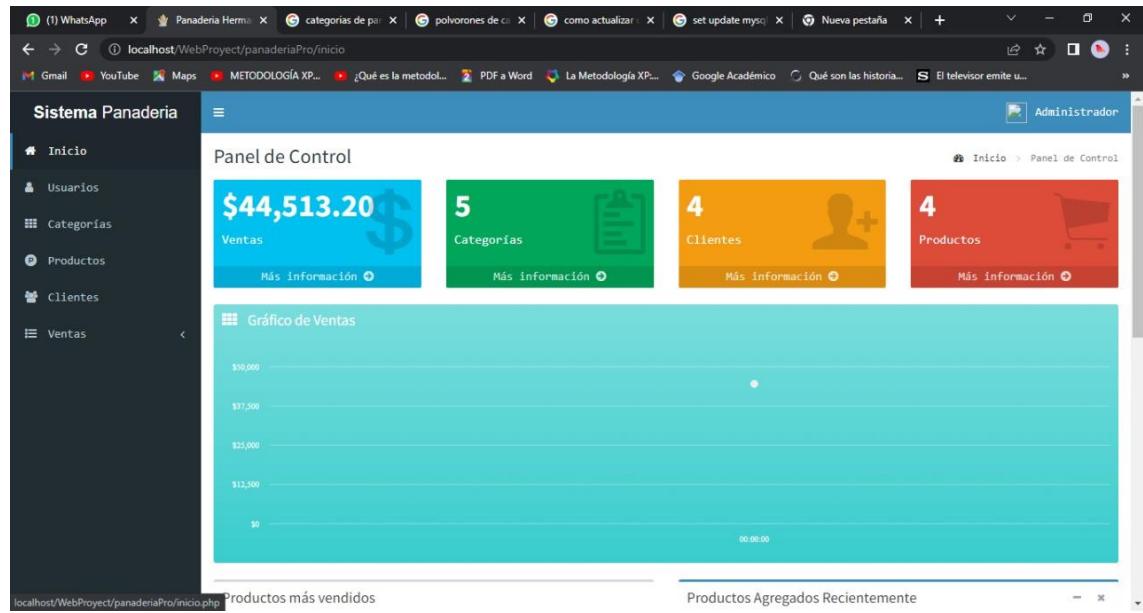


Ilustración 48 Usuario Administrador

b. Vendedor

Por otra parte, el perfil del vendedor es el que es encargado de ofrecer los productos, dicho perfil solamente tiene acceso a 2 funciones dentro del sistema que son los clientes y las ventas.

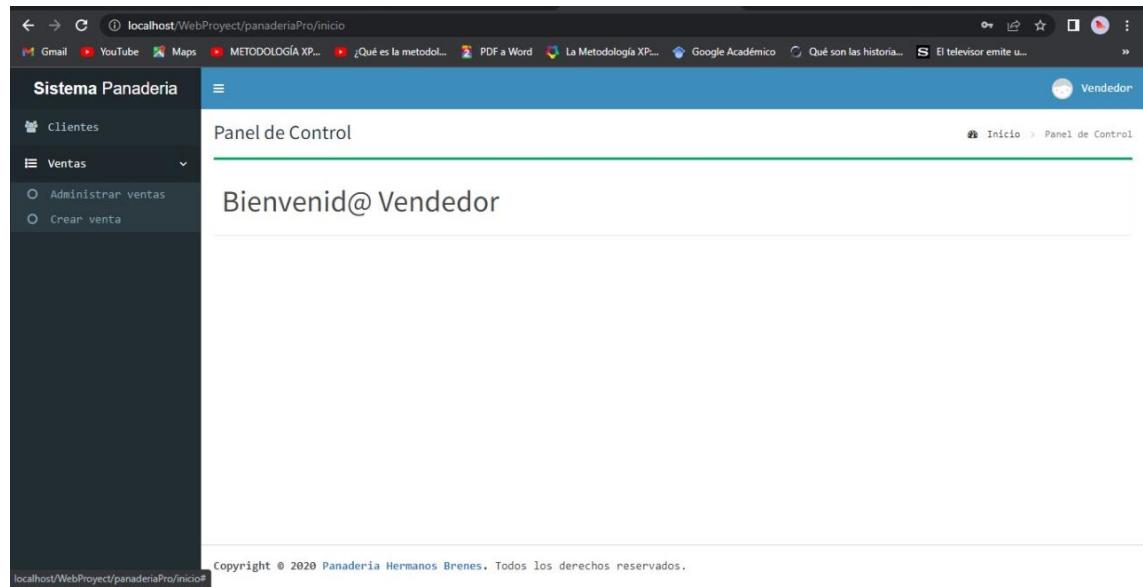


Ilustración 49 Usuario vendedor

c. Especial

El perfil de usuario especial es el encargado de ingresar productos y crear categorías al sistema, para que así el vendedor a la hora de hacer un negocio le aparezcan disponibles dichos productos.

#	Imagen	Código	Descripción	Categoría	Stock	Precio de compra	Precio de venta	Agregado	Acciones
1		004	Pan de ajonjoli	Pan de trigo	0	20	28.4	2022-05-05 09:43:23	
2		003	Pan aniquelado con mantequilla	Pan Negro	25	15	21	2022-05-05 09:47:17	
3		002	Pan dietetico	Pan Integral	5	28	36.4	2022-05-05 09:43:23	
4		001	Polvorones de canela	Repostería	75	50	55	2022-05-05 09:43:23	

Ilustración 50 Usuario especial

8. Agregar nuevo usuario

Para agregar un nuevo usuario, primeramente, hay que tener permisos de administrador ya que los demás roles no tienen permisos para manipular a los usuarios.

En la ventana de usuarios damos clic en agregar nuevo usuario y nos aparecerá otra ventana en donde tenemos que llenar los campos con la información del mismo, aquí se le asignará que tipo de usuario será, una vez llenado los datos guardamos cambios.

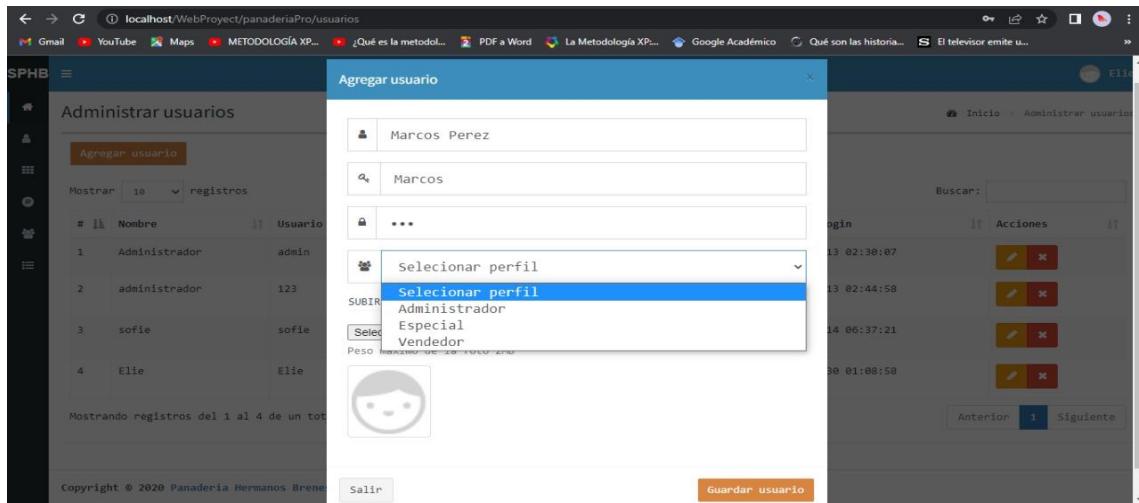


Ilustración 51 Agregar nuevo usuario

Al guardar dicha información nos aparecerá un mensaje en donde nos dice que el usuario ha sido guardado correctamente.

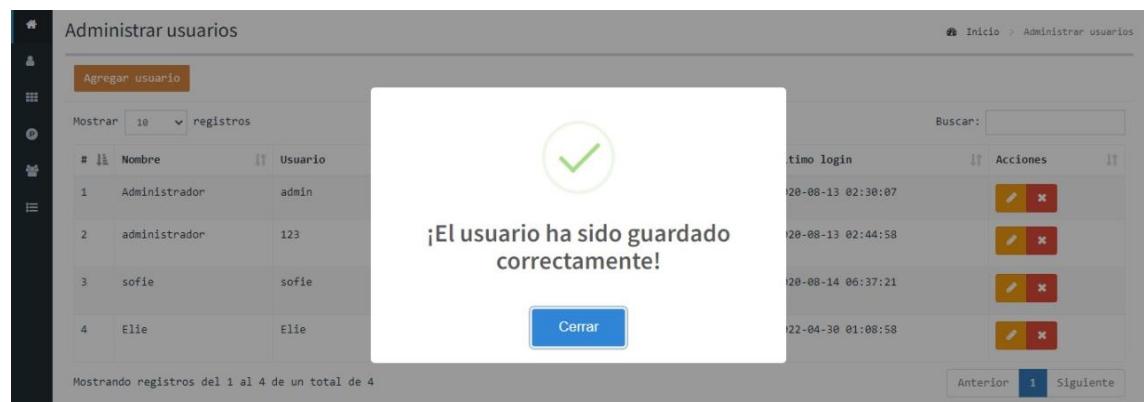


Ilustración 52 Mensaje de confirmación del gestor añadir usuario

Luego revisamos nuestra lista de usuarios y podremos notar que el usuario que acabamos de agregar está desactivado y daremos clic en activar.

#	Nombre	Usuario	Foto	Perfil	Estado	Último login	Acciones
1	Administrador	admin		Administrador	Activado	2020-08-13 02:30:07	
2	administrador	123		Administrador	Activado	2020-08-13 02:44:58	
3	sofie	sofie		Administrador	Activado	2020-08-14 06:37:21	
4	Elie	Elie		Administrador	Activado	2022-04-30 01:08:58	
5	Marcos Perez	Marcos		Administrador	Desactivado	0000-00-00 00:00:00	

Ilustración 53 Vista usuarios

9. Modificar Usuario

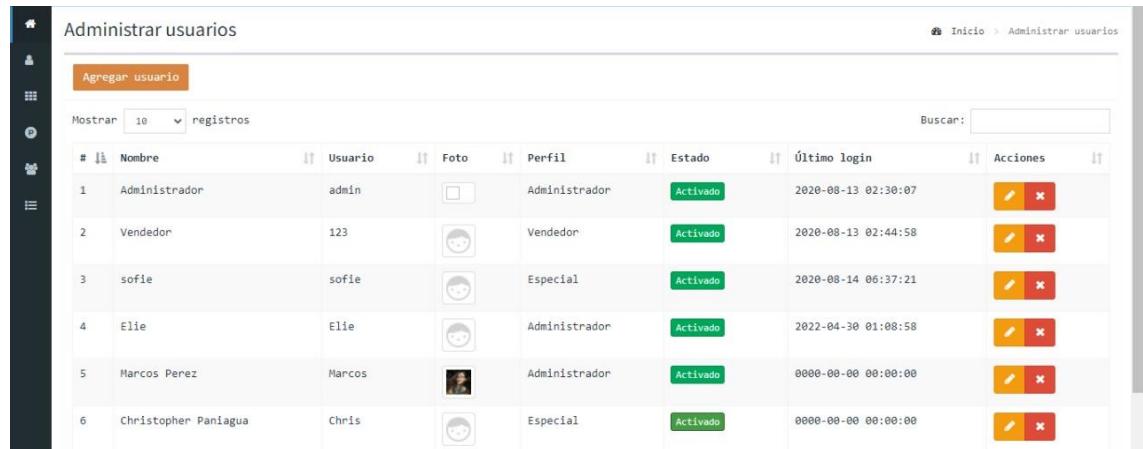
Si queremos modificar un usuario lo primero que tenemos que hacer es ir a nuestra lista y seleccionar al usuario que queremos modificar, para ello tendremos que dar un clic en el botón con el símbolo del lápiz.

Luego nos aparecerá una ventana con la información del usuario que queremos modificar y luego solo damos clic en guardar cambios.

Ilustración 54 Modificar un usuario

10. Eliminar usuario

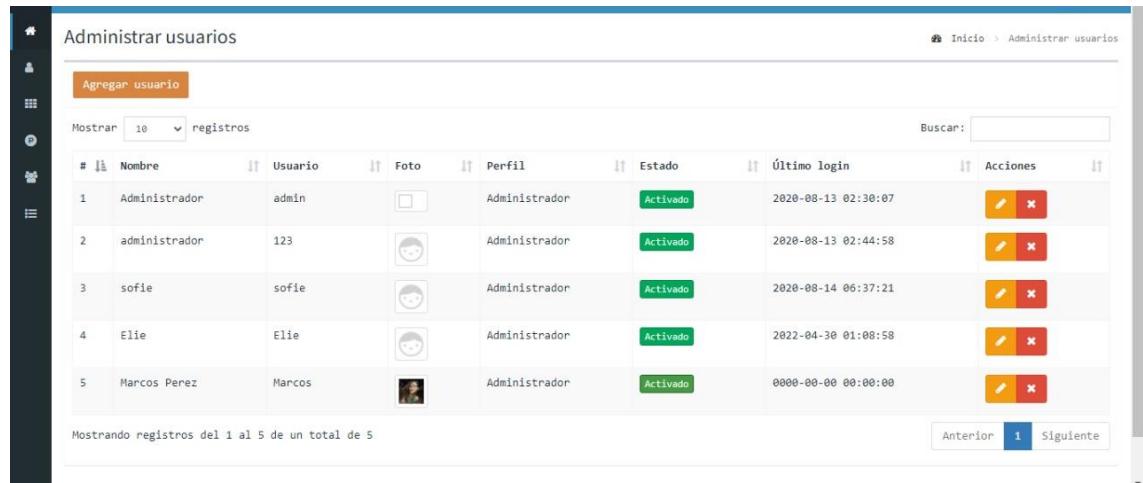
Para eliminar un usuario solamente tendremos que ir a nuestra lista de usuarios, seleccionamos el usuario que queremos eliminar y daremos clic en el botón que contiene una X.



#	Nombre	Usuario	Foto	Perfil	Estado	Último login	Acciones
1	Administrador	admin		Administrador	Activado	2020-08-13 02:30:07	
2	Vendedor	123		Vendedor	Activado	2020-08-13 02:44:58	
3	sofie	sofie		Especial	Activado	2020-08-14 06:37:21	
4	Elie	Elie		Administrador	Activado	2022-04-30 01:08:58	
5	Marcos Perez	Marcos		Administrador	Activado	0000-00-00 00:00:00	
6	Christopher Paniagua	Chris		Especial	Activado	0000-00-00 00:00:00	

Ilustración 55 Eliminar un usuario

Luego revisamos y veremos que el usuario ha sido eliminado correctamente.



#	Nombre	Usuario	Foto	Perfil	Estado	Último login	Acciones
1	Administrador	admin		Administrador	Activado	2020-08-13 02:30:07	
2	administrador	123		Administrador	Activado	2020-08-13 02:44:58	
3	sofie	sofie		Administrador	Activado	2020-08-14 06:37:21	
4	Elie	Elie		Administrador	Activado	2022-04-30 01:08:58	
5	Marcos Perez	Marcos		Administrador	Activado	0000-00-00 00:00:00	

Ilustración 56 Revisión de usuario eliminado

11. Agregar Categoría

Para poder agregar modificar y eliminar categorías, primeramente, hay que tener un usuario de tipo administrador o especial, ellos son los únicos que pueden manejar la información en la ventana categorías.

Para agregar una nueva categoría daremos un clic en agregar categoría y se nos abrirá una ventana en donde ingresaremos la información correspondiente a la nueva categoría.

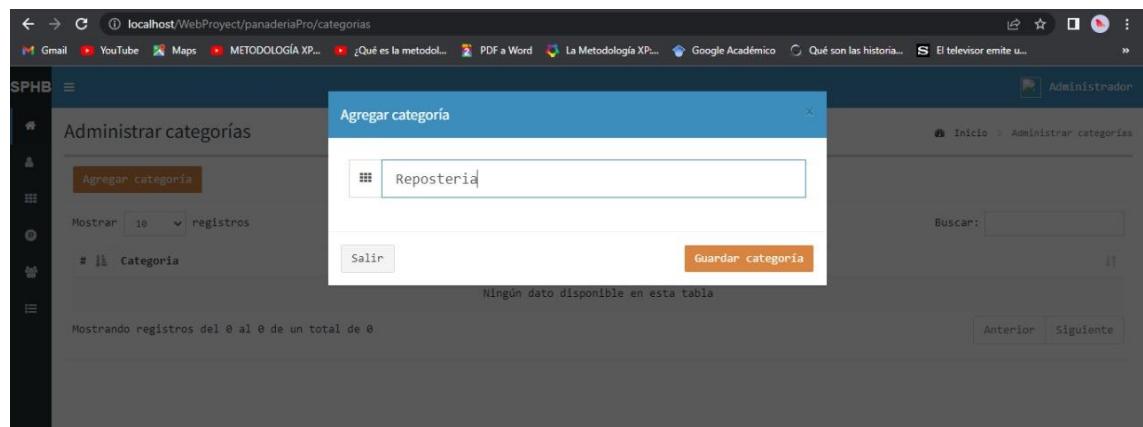


Ilustración 57 Agregar categoría

Luego de ingresar dicha información daremos clic en guardar y luego revisamos nuestra lista de categorías para corroborar que la categoría se haya guardado correctamente.

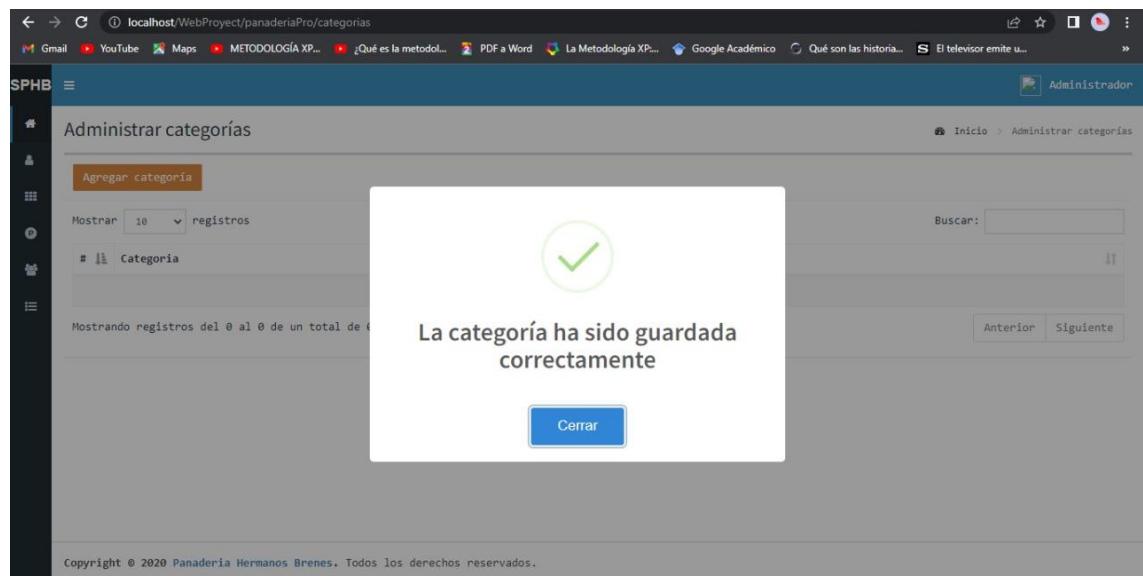


Ilustración 58 Mensaje de éxito de agregar categoría

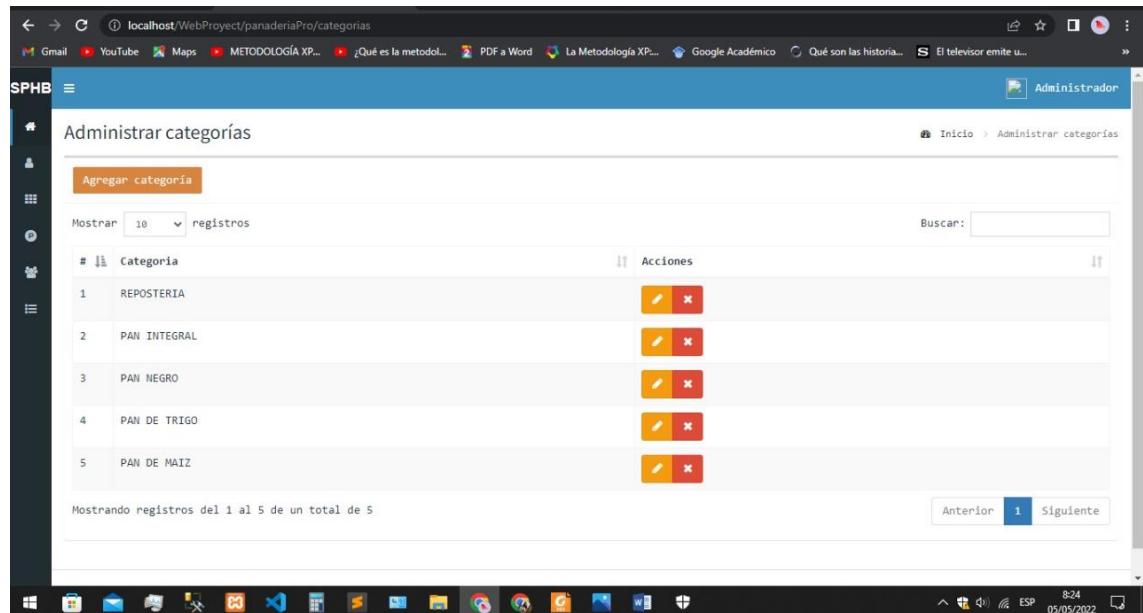


Ilustración 59 Administrar categoría

12. Modificar categoría

Si queremos modificar una categoría lo primero que tenemos que hacer es ir a nuestra lista y seleccionar la categoría que queremos modificar, si no encontramos la categoría entre la lista nos iremos a la caja de búsqueda e ingresamos el nombre y damos clic en buscar.

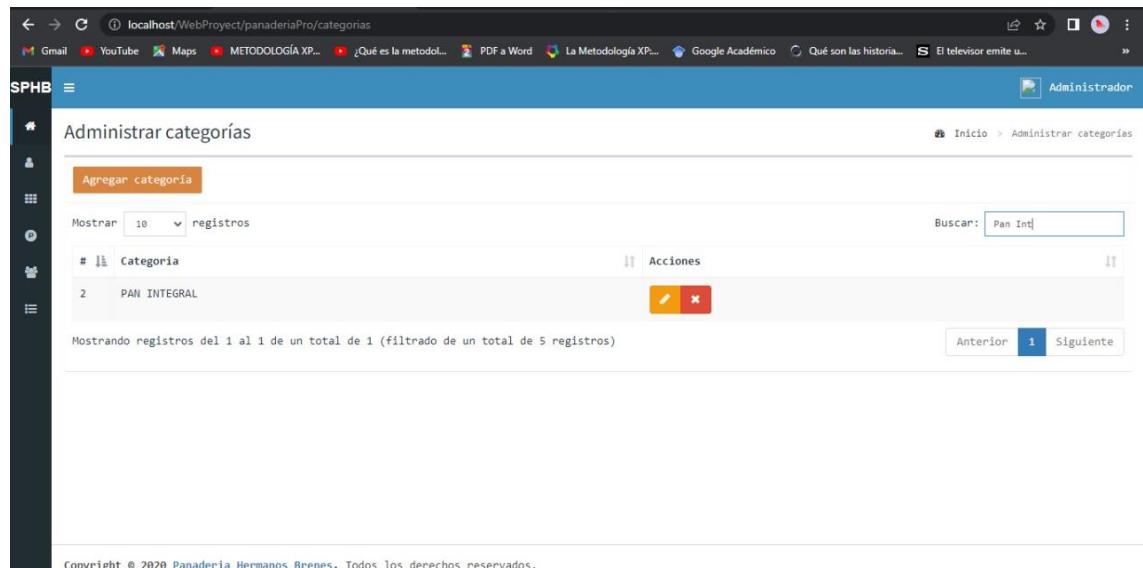


Ilustración 60 Modificar categoría

Una vez encontrada procedemos a modificar, para ello tendremos que dar un clic en el botón con el símbolo del lápiz.

Luego nos aparecerá una ventana con la información de la categoría modificamos la información y luego solo damos clic en guardar cambios.

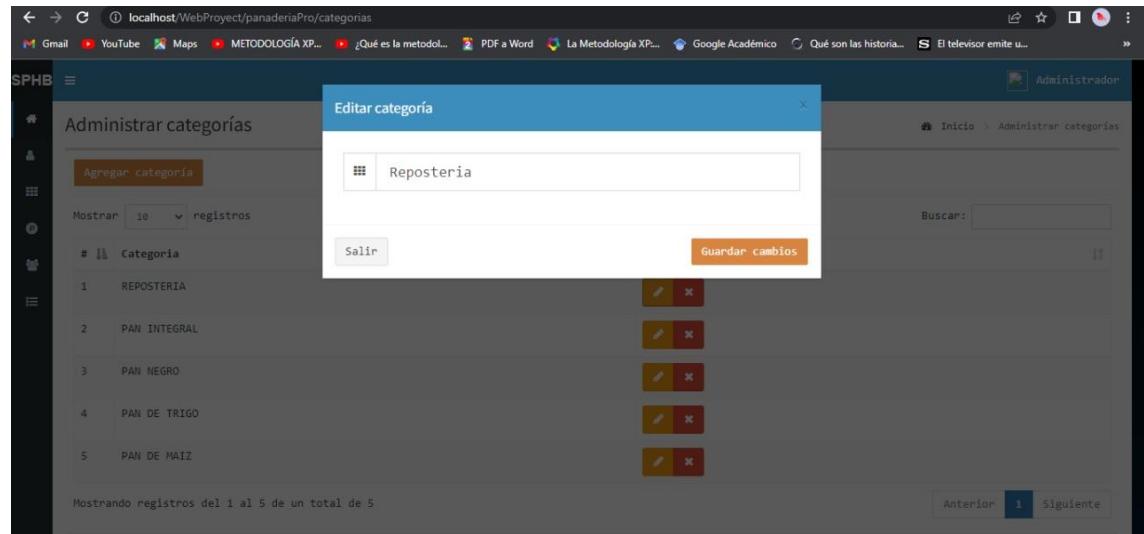


Ilustración 61 Guardar cambios de categoría

13. Eliminar categoría

Para eliminar una categoría solamente tendremos que ir a nuestra lista, seleccionamos la categoría que queremos eliminar y daremos clic en el botón que contiene una X.

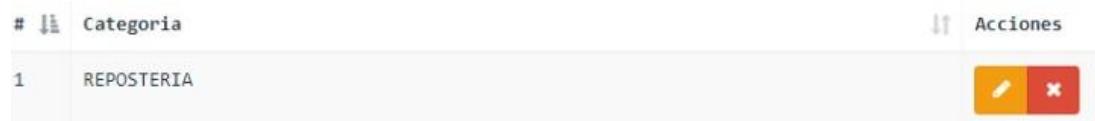


Ilustración 62 Eliminar categoría

14. Productos

En la ventana productos están visualizados los productos existentes, ósea todo lo que ya está en el inventario, claro está en que se pueden ingresar modificar y eliminar productos

Ilustración 63 Vista general de productos

15. Agregar productos

Para agregar un nuevo producto nos dirigiremos a la ventana llamada productos y daremos un clic en agregar producto, luego nos aparecerá una ventana en donde ingresaremos la información correspondiente.

Ilustración 64 Agregar un producto

Cuando estemos ingresamos la información primero hay que seleccionar la categoría a la cual pertenece dicho producto.

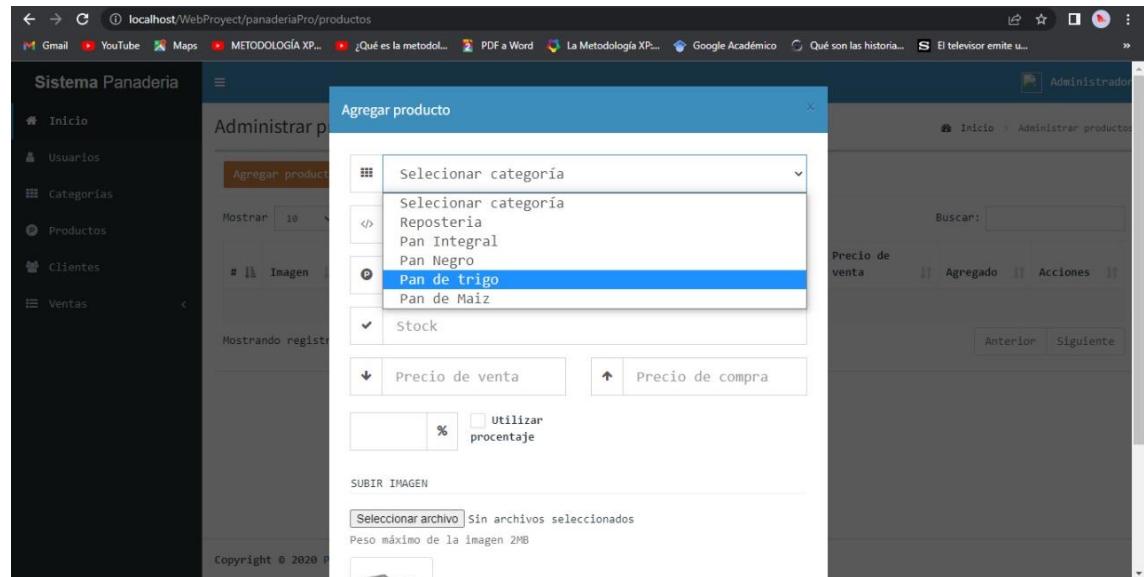


Ilustración 65 Seleccionar un producto

Luego el stock es la cantidad de productos que hay en existencia, precio de compra y venta, el porcentaje el cual se quiere obtener en ganancia y añadir una imagen del producto.

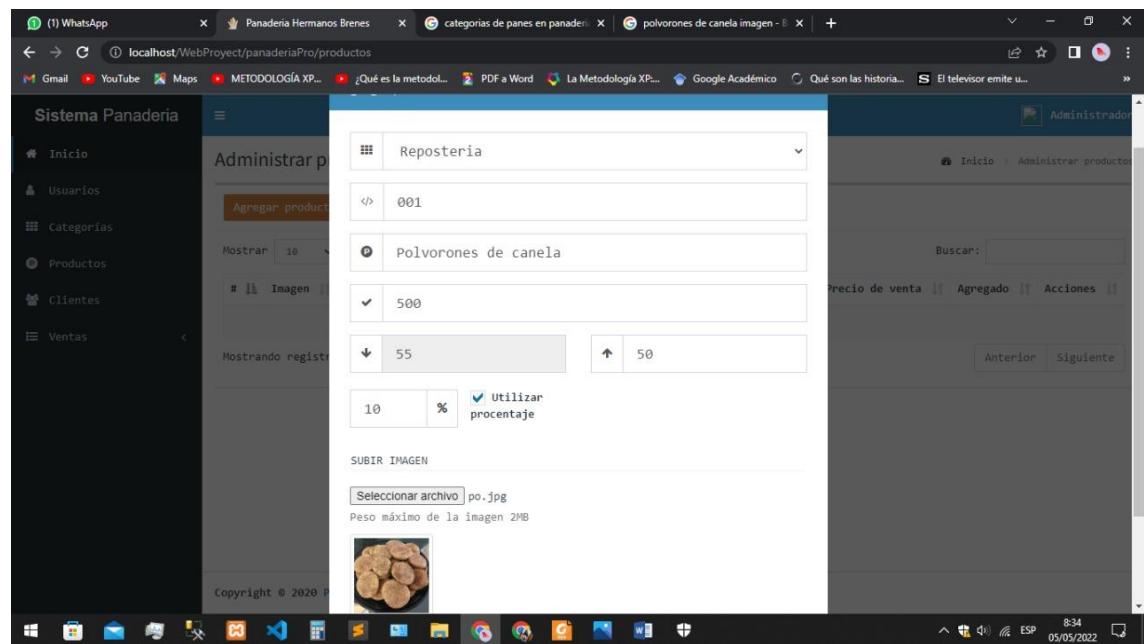


Ilustración 66 Registrar la venta

Por últimos damos clic en guardar y nuestro producto se visualizará en la lista.

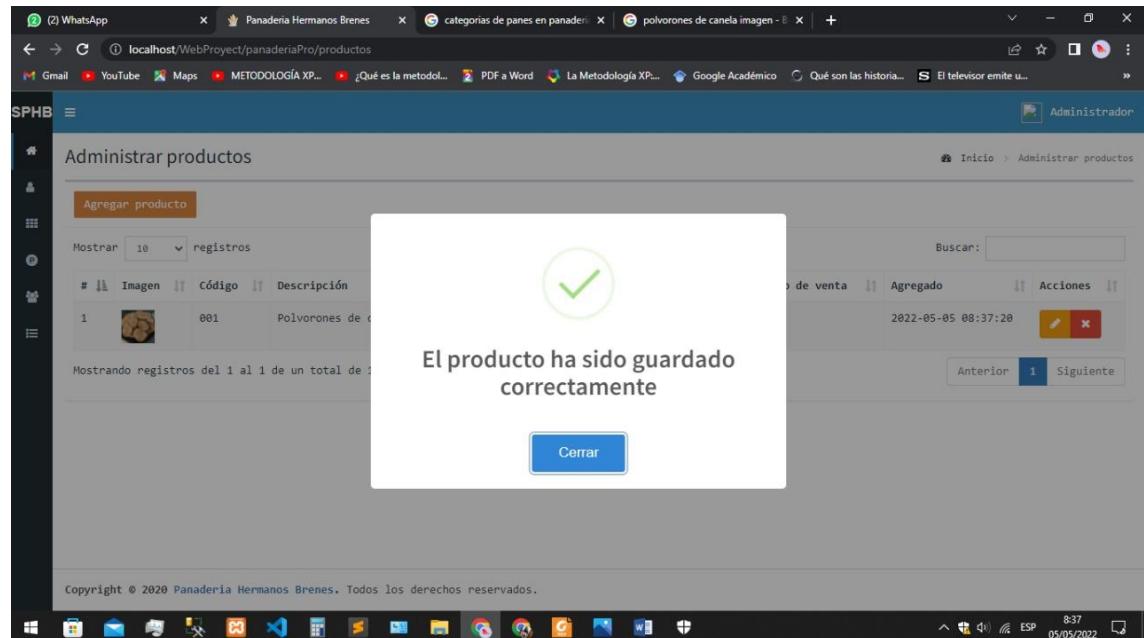


Ilustración 67 Mensaje de confirmación de la venta registrada

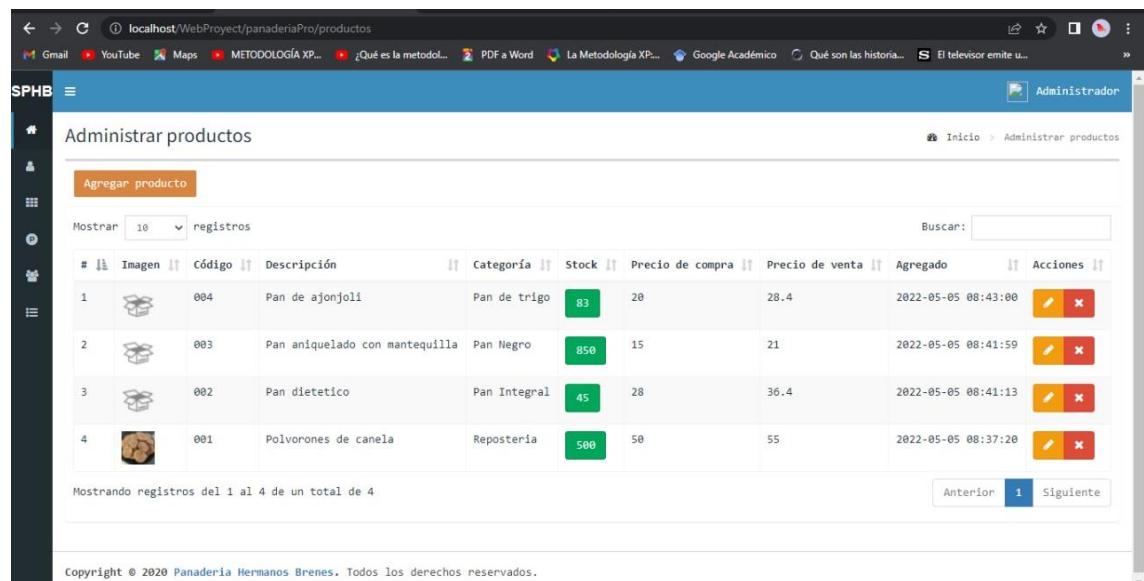


Ilustración 68 Vista general de administrar productos

16. Modificar producto

Para modificar un producto, primero buscamos el producto que quiero modificar lo seleccionamos y damos un clic en el botón de editar, luego se

nos abrirá una ventana en donde modificaremos la información y guardamos cambios.

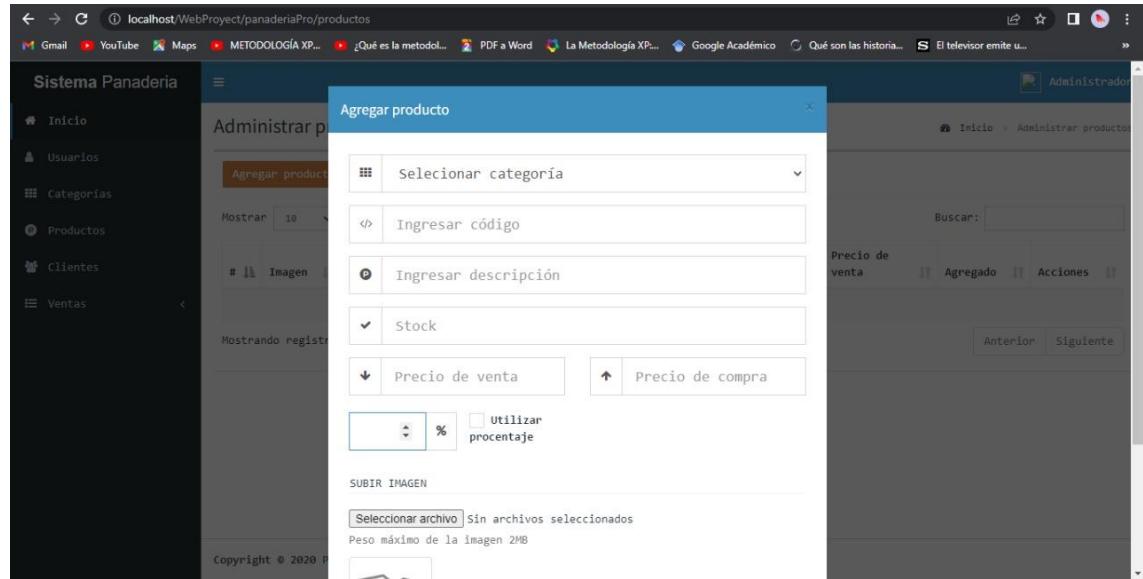


Ilustración 69 Modificar un producto

17. Eliminar producto

Para eliminar un producto, primero buscamos el producto que queremos eliminar lo seleccionamos y damos clic en el botón de eliminar.

	002	Pan dietetico	Pan Integral	45	28	36.4	2022-05-05 08:41:13	
	001	Polvorones de canela	Reposteria	500	50	55	2022-05-05 08:37:20	

Ilustración 70 Eliminar un producto

18. Advertencia

Cuando un producto ya es vendido automáticamente el contador de stock llegara a cero y se reflejara en nuestra lista.

#	Imagen	Código	Descripción	Categoría	Stock	Precio de compra	Precio de venta	Agregado	Acciones
1		004	Pan de ajonjoli	Pan de trigo	0	28	28.4	2022-05-05 09:43:23	
2		003	Pan aniquelado con mantequilla	Pan Negro	150	15	21	2022-05-05 09:43:23	
3		002	Pan dietético	Pan Integral	5	28	36.4	2022-05-05 09:43:23	
4		001	Polvorones de canela	Repostería	75	50	55	2022-05-05 09:43:23	

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Copyright © 2020 Panadería Hermanos Brenes. Todos los derechos reservados.

Ilustración 71 Advertencia

19. Clientes

En la ventana de los clientes cabe recalcar que los clientes son establecimientos en donde compran el producto al por mayor, en donde se le solicita la información necesaria para que sea un cliente fijo.

Los únicos que pueden manipular la lista de los clientes son el administrador y el vendedor, ellos son los encargados de ingresar, modificar y eliminar clientes de la lista.

#	Nombre	Documento ID	Email	Teléfono	Dirección	Fecha nacimiento	Total compras	Última compra	Ingreso al sistema	Acciones
1	Marcos	9	maernesto@gmail.com	(999) 999-9999	calle 13	1973-04-13	15	2022-05-05 10:20:24	2022-05-05 09:20:24	
2	Ernesto	1	evega@gmail.com	(222) 222-2222	Masatepe	1990-08-25	0	0000-00-00 00:00:00	2022-05-05 08:57:04	
3	Marie Sanchez	1254	mariel34@gmail.com	(885) 588-5545	San Marcos	1975-02-27	100	2022-05-05 10:47:17	2022-05-05 09:47:17	
4	Christopher Paniagua	5685	chrispaniagua@gmail.com	(584) 949-4254	Domitila Lugo	1983-02-14	1258	2022-05-05 10:46:20	2022-05-05 09:46:20	

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Ilustración 72 Vista general clientes

20. Agregar cliente

Para agregar un nuevo cliente, daremos un clic en el botón de agregar cliente, luego se nos abrirá una ventana en donde ingresaremos la información de nuestro nuevo cliente.

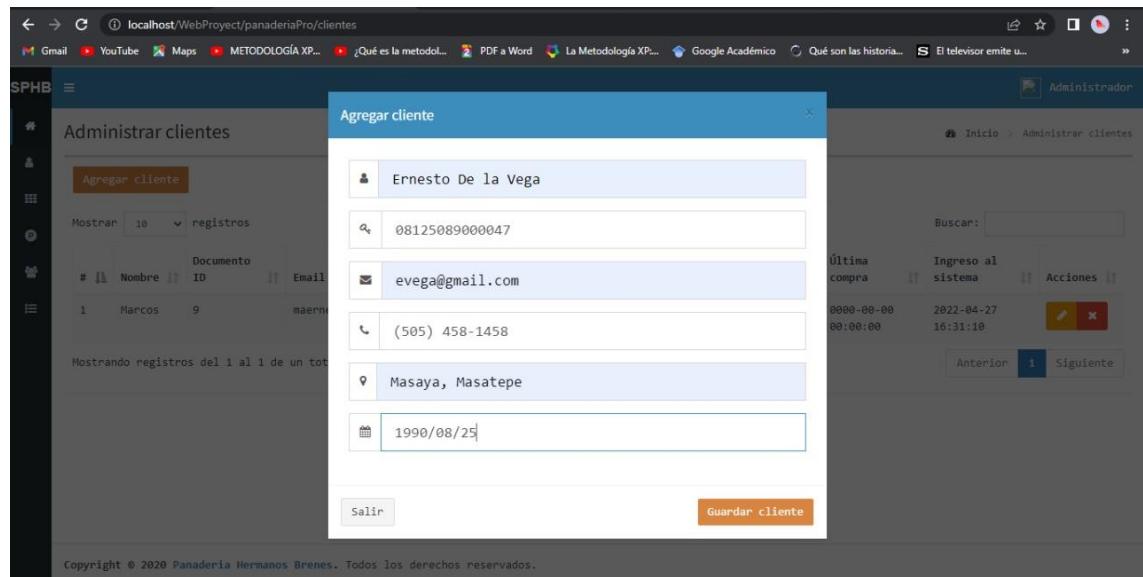


Ilustración 73 Agregar cliente

Al terminar de llenar los campos, guardamos y revisaremos nuestra lista de clientes para ver los agregados recientemente.

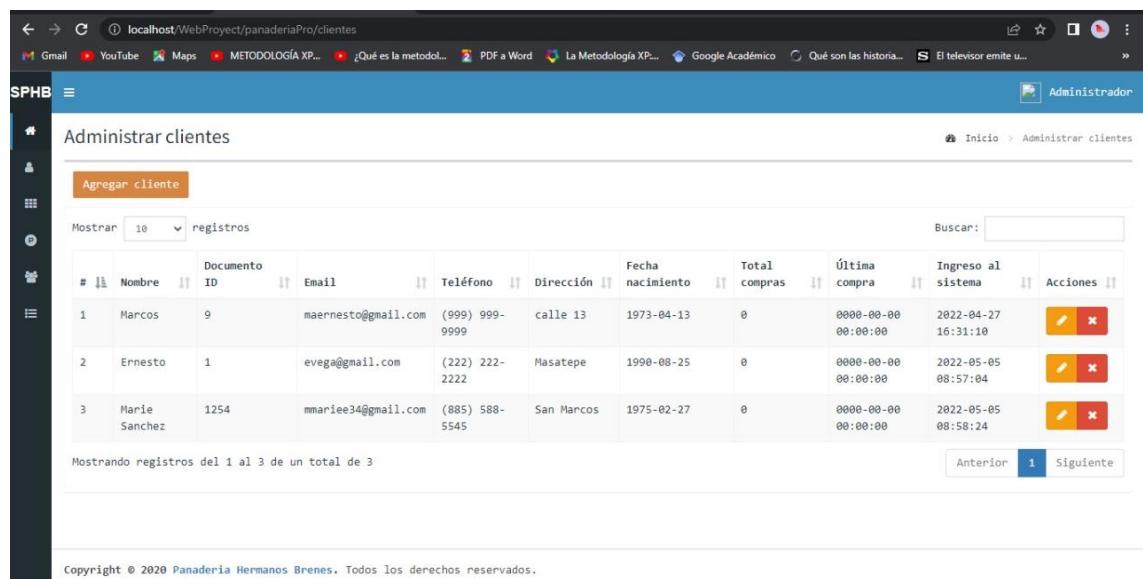


Ilustración 74 Clientes agregados

21. Modificar cliente

Para modificar un cliente, primeramente, buscaremos al cliente que queremos modificar una vez encontrado, procederemos a seleccionar y en el apartado de acciones seleccionamos modificar.

Luego se nos abrirá una ventana en donde modificaremos la información y guardamos cambios.

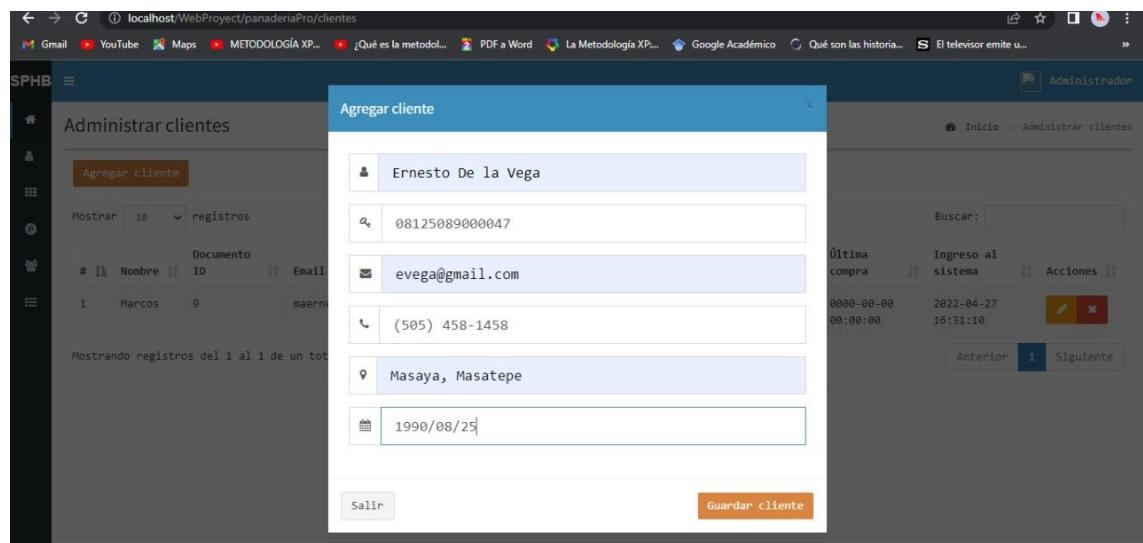


Ilustración 75 Modificar un cliente

22. Eliminar cliente

Para eliminar un cliente, primero tenemos que buscar en nuestra lista al cliente que queremos borrar, una vez encontrado seleccionamos y en el apartado de las acciones daremos un clic en el botón rojo para eliminar al cliente.

Nombre	ID	Documento	Email	Teléfono	Dirección	Fecha nacimiento	Total compras	Última compra	Ingreso al sistema	Acciones
Marcos	9		maernesto@gmail.com	(999) 999-9999	calle 13	1973-04-13	0	0000-00-00 00:00:00	2022-04-27 16:31:10	 

Ilustración 76 Eliminar un cliente

23. Ventas

En la ventana de ventas se puede visualizar toda la información sobre las ventas que se han guardado, en esta venta el administrador y el vendedor son los únicos que tienen acceso para manipular el contenido de la misma.

Aquí podemos agregar, modificar y eliminar una venta, también podemos obtener reportes, para saber cuáles han sido los promedios de ventas.

#	Código factura	Cliente	Vendedor	Forma de pago	Neto	Total	Fecha	Acciones
1	10001	Marcos	Administrador	Efectivo	\$ 426.00	\$ 430.26	2022-05-05 09:20:25	<button>Voucher</button> <button>PDF</button>
2	10002	Christopher Paniagua	Administrador	Efectivo	\$ 41,462.20	\$ 41,462.20	2022-05-05 09:43:24	<button>Voucher</button> <button>PDF</button>
3	10003	Christopher Paniagua	Administrador	Efectivo	\$ 525.00	\$ 525.00	2022-05-05 09:46:20	<button>Voucher</button> <button>PDF</button>
4	10004	Marie Sanchez	Administrador	Efectivo	\$ 2,100.00	\$ 2,100.00	2022-05-05 09:47:17	<button>Voucher</button> <button>PDF</button>

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Copyright © 2020 Panaderia Hermanos Brenes. Todos los derechos reservados.

Ilustración 77 Vista general ventas

24. Agregar venta

Para agregar una venta, primero daremos un clic en agregar venta, luego se nos abrirá una ventana en donde llenaremos los campos con la información correspondiente.

#	Imagen	Código	Descripción	Stock	Acciones
1		004	Pan de ajonjoli	83	<button>Agregar</button>
2		003	Pan aníquelado con mantequilla	850	<button>Agregar</button>
3		002	Pan dietético	45	<button>Agregar</button>
4		001	Polvorones de canela	500	<button>Agregar</button>

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Seleccionar método de pago

Guardar venta

Ilustración 78 Agregar una venta

Podemos notar que dentro la información aparece el nombre del vendedor por defecto esto debido a que este se llenara automáticamente con el nombre del usuario de la cuenta, en este caso usamos la de administrador. Luego procedemos a seleccionar un cliente, el cual está previamente registrado dentro del el sistema.

#	Imagen	Código	Descripción	Stock	Acciones
1		004	Pan de ajojoli	83	<button>Agregar</button>
2		003	Pan aniquelado con mantequilla	850	<button>Agregar</button>
3		002	Pan dietetico	45	<button>Agregar</button>
4		001	Polvorones de canela	500	<button>Agregar</button>

Ilustración 79 Seleccionar cliente

Luego agregamos los productos de acuerdo al pedido de nuestro cliente, aplicamos impuestos, seleccionamos la forma de pago y guardamos nuestra venta.

Se elige método de pago, porque en un futuro se planea implementar tarjetas de crédito.

localhost/WebProject/panaderiaPro/crear-venta

SPHB Administrador

Crear venta

Administrador

10001

Marcos Agregar cliente

\$ 426.00 15 Pan de ajonjoli

Impuesto Total

0 % \$ 426.00

Seleccionar método de pago

Mostrar 10 registros Buscar:

#	Imagen	Código	Descripción	Stock	Acciones
1		004	Pan de ajonjoli	83	Agregar
2		003	Pan aniquelado con mantequilla	850	Agregar
3		002	Pan dietetico	45	Agregar
4		001	Polvorones de canela	500	Agregar

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Ilustración 80 Método de pago parte I

localhost/WebProject/panaderiaPro/crear-venta

SPHB Administrador

Crear venta

Administrador

10001

Marcos Agregar cliente

\$ 426.00 15 Pan de ajonjoli

Impuesto Total

0 % \$ 426.00

Efectivo
Seleccionar método de pago
Efectivo
Tarjeta Crédito
Tarjeta Débito

Guardar venta

Mostrar 10 registros Buscar:

#	Imagen	Código	Descripción	Stock	Acciones
1		004	Pan de ajonjoli	83	Agregar
2		003	Pan aniquelado con mantequilla	850	Agregar
3		002	Pan dietetico	45	Agregar
4		001	Polvorones de canela	500	Agregar

Mostrando registros del 1 al 4 de un total de 4

Anterior 1 Siguiente

Ilustración 81 Método de pago parte II

#	Imagen	Código	Descripción	Stock	Acciones
1		004	Pan de ajonjoli	83	<button>Agregar</button>
2		003	Pan aniquelado con mantequilla	850	<button>Agregar</button>
3		002	Pan dietético	45	<button>Agregar</button>
4		001	Polvorones de canela	500	<button>Agregar</button>

Ilustración 82 Crear venta

Ilustración 83 Mensaje de confirmación de un registro de venta

Una vez que nuestra venta ha sido guardada exitosamente podemos imprimir la factura del cliente.

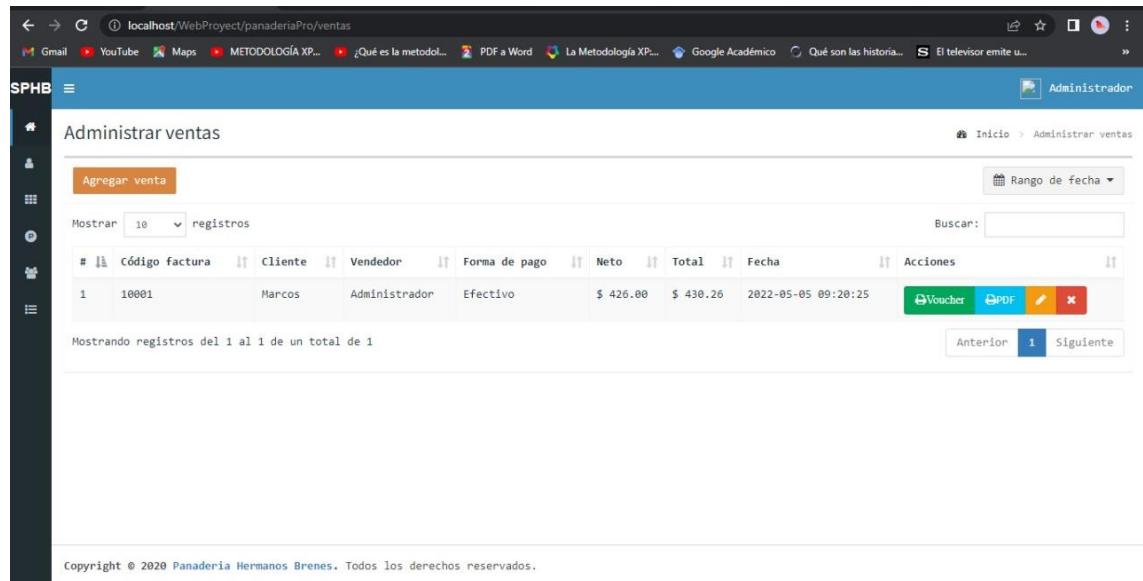


Ilustración 84 Impresión de factura

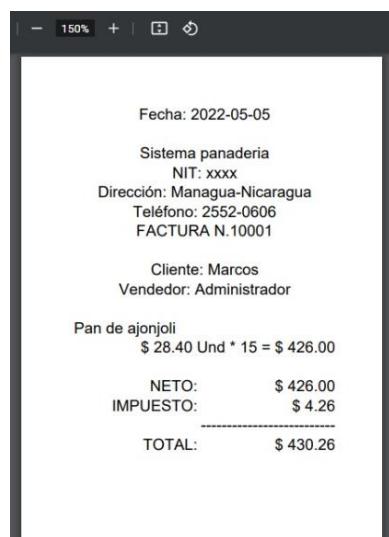


Ilustración 85 Formato de la factura del cliente

25. Modificar venta

Para modificar una venta, primero tenemos que buscar la venta la cual queremos modificar, luego seleccionamos la venta y en el apartado de acciones daremos clic en modificar y se nos abrirá una ventana la en donde cambiaremos la información y guardamos.

#	Imagen	Código	Descripción	Stock	Acciones
1		004	Pan de ajonjoli	83	<button>Agregar</button>
2		003	Pan aniquelado con mantequilla	850	<button>Agregar</button>
3		002	Pan dietetico	45	<button>Agregar</button>
4		001	Polvorones de canela	500	<button>Agregar</button>

Ilustración 86 Modificar venta

26. Eliminar Venta

Para eliminar una venta, primeramente, tenemos que buscar la venta la cual deseamos eliminar, una vez encontrada la seleccionamos y nos vamos al apartado de acciones en donde daremos clic en el botón rojo, para eliminar la información.

#	Código factura	Cliente	Vendedor	Forma de pago	Neto	Total	Fecha	Acciones
1	10001	Marcos	Administrador	Efectivo	\$ 426.00	\$ 430.26	2022-05-05 09:28:25	   

Ilustración 87 Eliminar ventas

27. Reportes de ventas

Nuestro sistema también crea los reportes de ventas ya sea de 1 a N días del mes, esto con el fin de saber el promedio de ganancias, para ello en la ventana de ventas seleccionamos la ventana de reportes de ventas y damos clic para acceder.

Seleccionamos un rango de tiempo del cual queremos ver dicho reporte, este rango de tiempo puede ser de 1 a N días y luego solo aplicamos y generamos el reporte.

Ya una vez generado el reporte podemos apreciar que nos aparece información como: promedio de ventas, producto mas vendido, ganancias y quien fue el mejor vendedor.

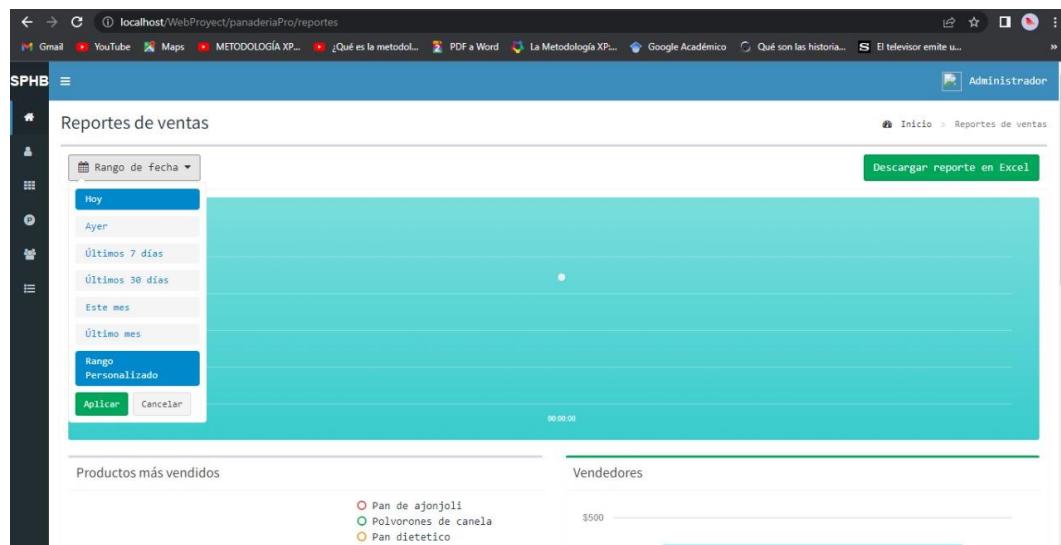


Ilustración 88 Reportes de ventas parte I

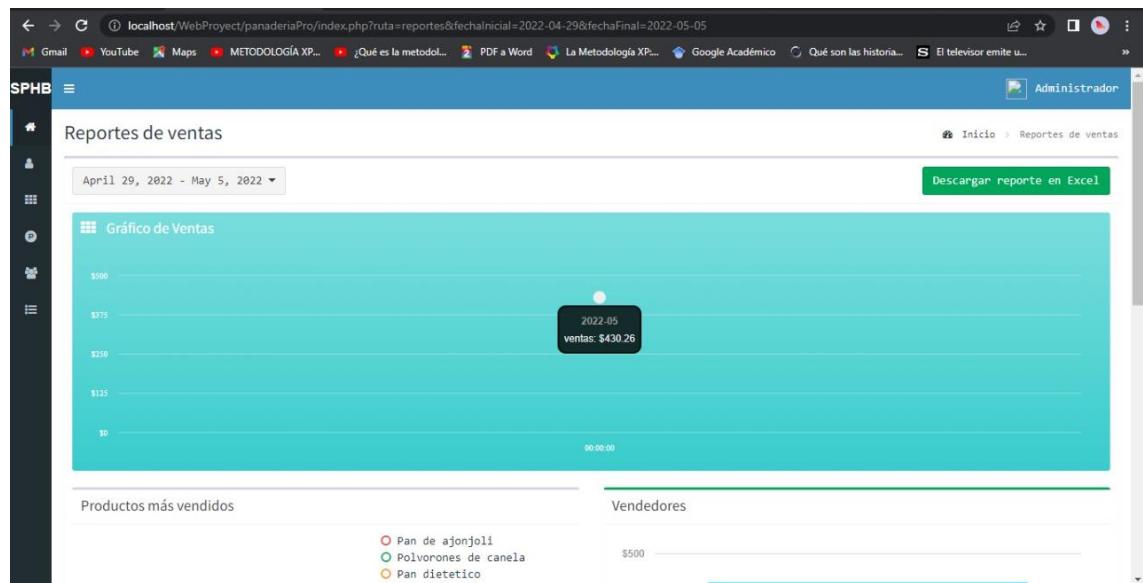


Ilustración 89 Reporte de ventas parte II

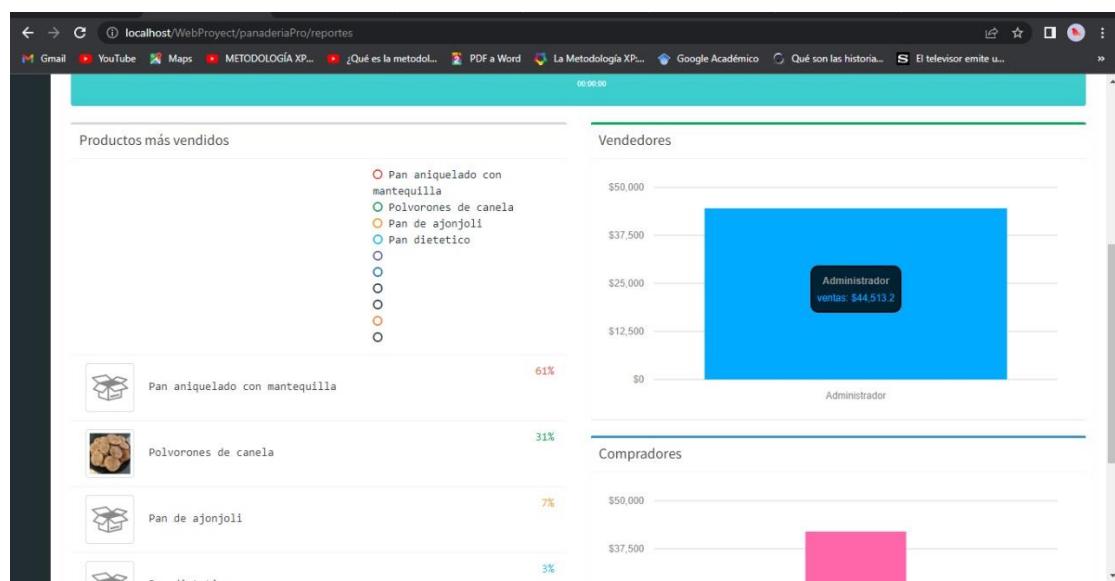


Ilustración 90 Reporte de ventas parte III

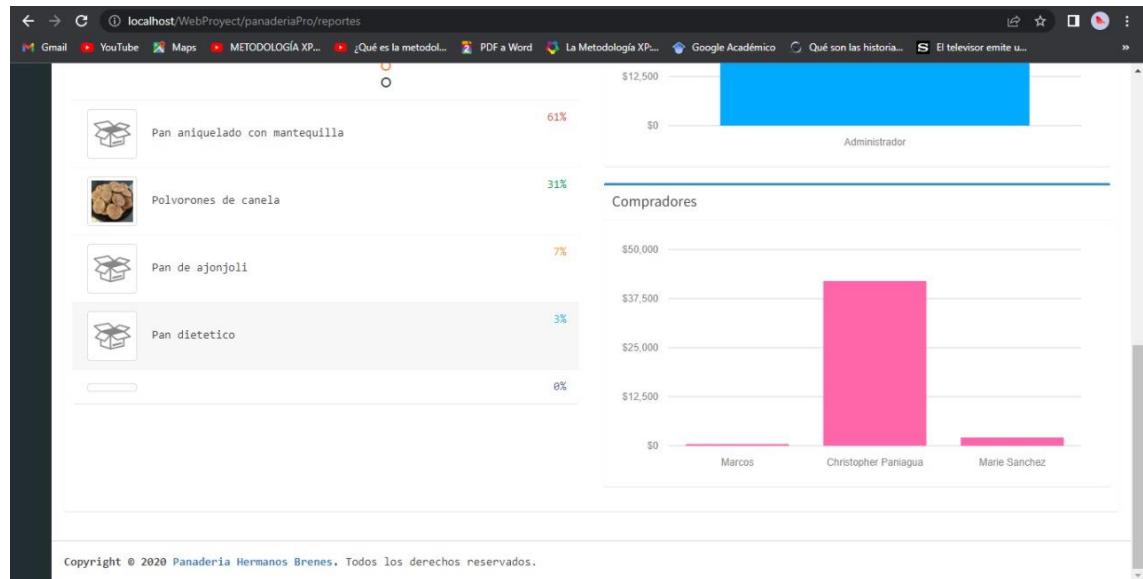


Ilustración 91 Reporte de ventas parte IV

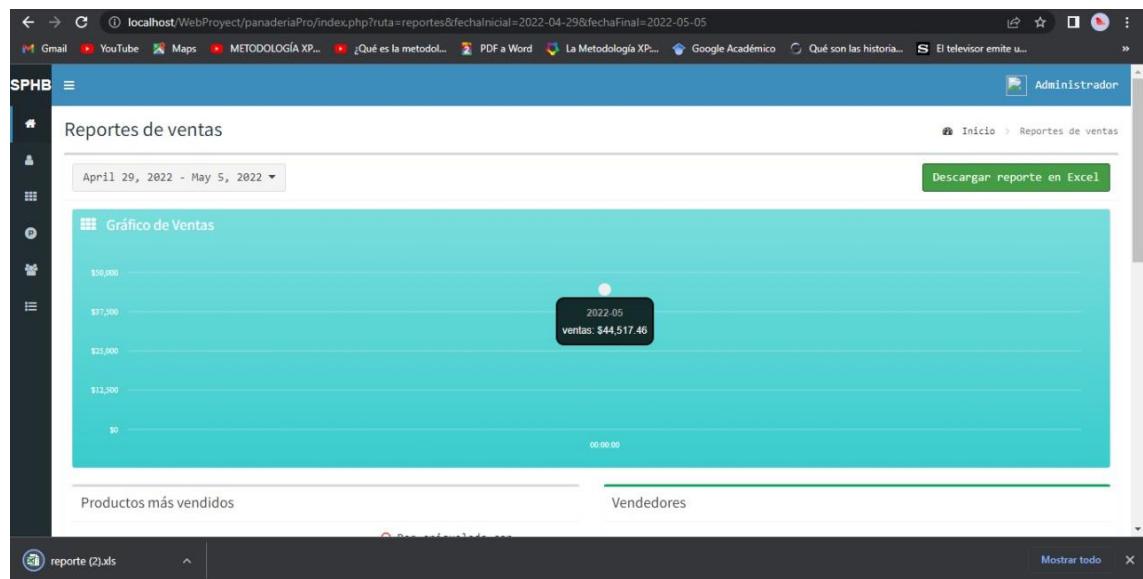


Ilustración 92 Reporte de ventas parte V

Luego si queremos descargar nuestro reporte en formato Excel solo daremos clic en el botón de descargar.

A	B	C	D	E	F	G	H	I	J
CÓDIGO	CLIENTE	VENDEDOR	CANTIDAD	PRODUCTOS	IMPUESTO	NETO	TOTAL	METODO DE PAGO	FECHA
2	10001 Marcos	Administrador	15	Pan de ajonjoli	\$4.26	\$426.00	\$430.26	Efectivo	05/05/2022
3			700	Pan aniquelado con mantequilla					
4			68	Pan de ajonjoli					
5			40	Pan dietético					
6	10002 Christopher Paniagua	Administrador	425	Polvorones de canela	\$0.00	\$41,462.20	\$41,462.20	Efectivo	05/05/2022
7	10003 Christopher Paniagua	Administrador	25	Pan aniquelado con mantequilla	\$0.00	\$525.00	\$525.00	Efectivo	05/05/2022
8	10004 Marie Sanchez	Administrador	100	Pan aniquelado con mantequilla	\$0.00	\$2,100.00	\$2,100.00	Efectivo	05/05/2022

Ilustración 93 Archivo Excel del reporte de ventas

Al final observaremos que todos los procesos se visualizan en nuestro panel de control.

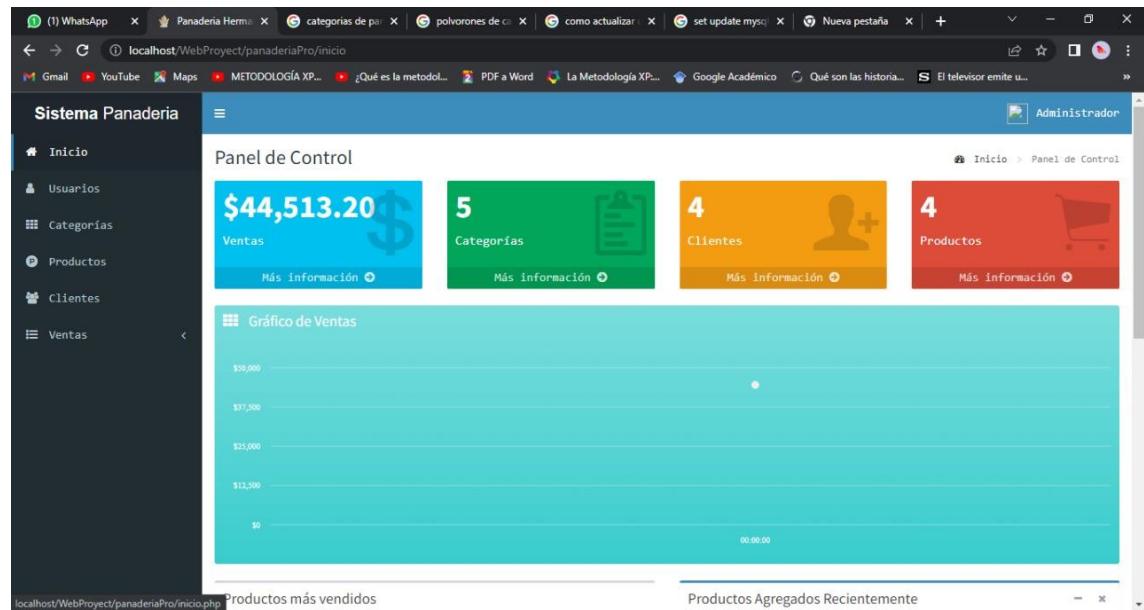


Ilustración 94 Procesos en el panel de control del sistema

Anexo D: Diccionario de datos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media type
id (<i>Primaria</i>)	int(11)	No				
categoria	text	No				
fecha	timestamp	No	current_timestamp()			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

clientes

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media type
id (<i>Primaria</i>)	int(11)	No				
nombre	text	No				
documento	int(11)	No				
email	text	No				
telefono	text	No				
direccion	text	No				
fecha_nacimiento	date	No				
compras	int(11)	No				
ultima_compra	datetime	No				
fecha	timestamp	No	current_timestamp()			

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

productos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media type
id (<i>Primaria</i>)	int(11)	No				
id_categoria	int(11)	No				
codigo	text	No				
descripcion	text	No				
imagen	text	No				
stock	int(11)	No				
precio_compra	float	No				
precio_venta	float	No				
ventas	int(11)	No				
fecha	timestamp	No	current_timestamp()			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

usuarios

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media type
id (<i>Primaria</i>)	int(11)	No				
nombre	text	No				
usuario	text	No				
password	text	No				
perfil	text	No				
foto	text	No				
estado	int(11)	No				
ultimo_login	datetime	No				
fecha	timestamp	No	current_timestamp()			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	4	A	No	

ventas

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media type
id (<i>Primaria</i>)	int(11)	No				
codigo	int(11)	No				
id_cliente	int(11)	No				
id_vendedor	int(11)	No				
productos	text	No				
impuesto	float	No				
neto	float	No				
total	float	No				
metodo_pago	text	No				
fecha	timestamp	No	current_timestamp()			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

Ilustración 95 Diccionario de datos