

**UNIVERSIDAD POLITÉCNICA DE PUEBLA**  
**Ingeniería en Informática**



**Proyecto de Estadía Profesional**

“Desarrollo de un punto de venta para la gestión de una papelería”

Área temática del CONACYT: VII  
Ingenierías y tecnologías

**Presenta:**

**José Alfredo Tlahuechtl Rodriguez**

**Asesor técnico**

Luis Tepox Romero

**Asesor académico**

Rebeca Rodríguez Huesca

Juan C. Bonilla, Puebla, México.

30 de abril de 2021

# Resumen

---

El presente documento se encuentra dividido por secciones iniciando por la introducción, la cual describe la problemática detectada en la empresa papelería Ingrid, los cuales serán enlistados en las subsecciones de introducción en mayor detalle, de la misma forma se presenta una justificación explicando por qué se propone el uso de un punto de venta, así como los objetivos a alcanzar para la finalización del proyecto.

Se encontrará además una sección dedicada a la metodología y herramientas usadas en construcción del punto de venta. Dando una breve descripción sobre la metodología empleada, dónde se puede usar y el por qué es la mejor opción a implementar en este problema, en las herramientas se enlistan las características, ventajas y desventajas de mismas, además de los lenguajes de programación usados en el desarrollo de este proyecto.

Los resultados, sección dedicada a los productos generados en cada una de las etapas de la metodología, en la presente sección se muestran evidencias de la primera versión y última del punto de venta. Además de pruebas formales realizadas en el uso de la aplicación web.

Conclusiones y recomendaciones sección dedicada a una opinión personal sobre la aplicación además de una serie de recomendaciones a tener en cuenta para mejorar el producto final, en caso de ser necesario.

Apéndices sección en la cual se muestran evidencias de diseño de base de datos, diccionario de datos, el script para generar la base de datos y código generado para el correcto funcionamiento del producto.

# Índice

---

Resumen .....	2
1. Introducción.....	9
1.1. Descripción del problema o necesidad.....	9
1.2 Justificación .....	9
1.3 Objetivo General y Específicos.....	10
El objetivo general .....	10
Objetivos específicos .....	10
2. Metodología y herramientas.....	11
2.1. Metodología de prototipos .....	11
2.2. Herramientas .....	14
3. Resultados .....	20
3.1. Definición de los requisitos .....	20
3.2. Análisis y Diseño.....	32
3.3. Codificación (construcción del prototipo) .....	48
3.4. Pruebas .....	55
3.5. Mantenimiento (retroalimentación).....	59
4. Conclusiones y recomendaciones .....	72
5. Apéndices .....	73
Apéndice A Script para generar BD .....	73
Apéndice B Código .....	75
6. Referencias bibliográficas .....	80

# Índice de figuras

---

Figura 1. Etapas de la metodología por prototipo	12
Figura 2. Caso de uso general	20
Figura 3. Caso de uso catalogar producto	21
Figura 4. Caso de uso catalogar proveedor	22
Figura 5. Caso de uso catalogar deudor	24
Figura 6. Caso de uso registrar compra	26
Figura 7. Caso de uso registrar venta	28
Figura 8. Caso de uso inventario	29
Figura 9. Caso de uso corte de caja	30
Figura 10. Caso de uso reportar venta	31
Figura 11. Diseño de baja fidelidad Pantalla de inicio	33
Figura 12. Diseño de baja fidelidad Pantalla de producto y formulario de modificación/registro	34
Figura 13. Diseño de baja fidelidad Pantalla de proveedor y formulario de modificación/registro	34
Figura 14. Diseño de baja fidelidad Pantalla de compras	35
Figura 15. Diseño de baja fidelidad Pantalla de ventas	35
Figura 16. Diseño de baja fidelidad Pantalla de corte de caja	36
Figura 17. Diseño de baja fidelidad Pantalla de reporte de ventas	36
Figura 18. Diseño de baja fidelidad Pantalla de inventario	37
Figura 19. Diseño de media fidelidad Pantalla de inicio	37
Figura 20. Diseño de media fidelidad Pantalla de productos	38

Figura 21. Diseño de media fidelidad formularios de registro y modificación	38
Figura 22. Diseño de media fidelidad Pantalla de proveedores	39
Figura 23. Diseño de media fidelidad formularios de registro y modificación proveedor	39
Figura 24. Diseño de media fidelidad Pantalla de Compra	40
Figura 25. Diseño de media fidelidad Pantalla de Venta	40
Figura 26. Diseño de media fidelidad Pantalla de corte de caja	41
Figura 27. Diseño de media fidelidad Pantalla de reporte de ventas	41
Figura 28. Diseño de media fidelidad Pantalla de inventario	42
Figura 29. Diseño de base datos	43
Figura 30. Primera versión: Pantalla principal del sistema	48
Figura 31. Primera versión: Pantalla datos del producto	49
Figura 32. Primera versión: Pantalla registro de producto	49
Figura 33. Primera versión: Mensaje de error algún campo vacío	50
Figura 34. Primera versión: Mensaje de error al no completarse un registro	50
Figura 35. Primera versión: Mensaje de registro completado	50
Figura 36. Primera versión: Datos del proveedor	51
Figura 37. Primera versión: Pantalla registro de proveedor	51
Figura 38. Primera versión: Pantalla de compras	52
Figura 39. Primera versión: Pantalla de ventas	52
Figura 40. Primera versión: Pantalla de reportes de venta	53
Figura 41. Primera versión: Pantalla de corte de caja	54
Figura 42. Primera versión: Pantalla de Inventario	54
Figura 43. Prototipo mejorado: Pantalla de productos	59

Figura 44. Prototipo mejorado: Alerta de los productos a agotarse	59
Figura 45. Prototipo mejorado: Formulario de registro para productos	60
Figura 46. Prototipo mejorado: Alerta por si algún campo está vacío	60
Figura 47. Prototipo mejorado: Pantalla de proveedores	60
Figura 48. Prototipo mejorado: Formulario de registro para proveedores	61
Figura 49. Prototipo mejorado: Pantalla de compras	61
Figura 50. Prototipo mejorado: Pantalla de ventas	62
Figura 51. Prototipo mejorado: Pantalla de inventario	62
Figura 52. Prototipo mejorado: Pantalla de corte de caja	63
Figura 53. Prototipo mejorado: Pantalla de reportes de ventas	63
Figura 54. Pantalla de inicio de sección	63
Figura 55. Prototipo mejorado: Pantalla principal	64
Figura 56. Pantalla de ventas	65
Figura 57. Pantalla de reporte de ventas	65
Figura 58. Pantalla de ticket	66
Figura 59. Pantalla de registro de producto	66
Figura 60. Pantalla de modificar producto	67
Figura 61. Pantalla de productos	67
Figura 62. Pantalla de registro deudor	68
Figura 63. Pantalla de deudores	68
Figura 64. Pantalla de confirmación	68
Figura 65. Pantalla de registro de proveedor	69
Figura 66. Pantalla de proveedores	69

Figura 67. Pantalla de actualización de información del proveedor	70
Figura 68. Pantalla de registro de usuarios	70
Figura 69. Pantalla de usuarios	71
Figura 70. Pantalla de actualización de usuario	71
Figura 71. Script para generar la base de datos (1/2)	73
Figura 72. Script para generar la base de datos (2/2)	74
Figura 73. Código fuente del menú	75
Figura 74. Código fuente para registrar nuevos productos.	75
Figura 75. Código fuente para registrar nuevos proveedores	76
Figura 76. Código fuente para crear registro de compras	77
Figura 77. Código fuente para crear folio de compra	78
Figura 78. Código fuente para incrementar la cantidad del producto (1/2)	78
Figura 79. Código fuente para incrementar la cantidad del producto (2/2)	78
Figura 80. Código fuente para crear registro de venta	78
Figura 81. Código fuente para crear un folio de venta	79
Figura 82. Código fuente para decrementar la cantidad del producto en existencia	79

# Índice de tablas

---

Tabla 1. Caso de uso detallado catalogar producto	22
Tabla 2. Caso de uso detallado catalogar proveedor	23
Tabla 3. Caso de uso detallado catalogar deudor	25
Tabla 4. Caso de uso detallado registrar compra	27
Tabla 5. Caso de uso detallado registrar venta	29
Tabla 6. Caso de uso detallado inventario	30
Tabla 7. Caso de uso detallado corte de caja	31
Tabla 8. Caso de uso detallado reportar venta	32
Tabla 9. Caso de prueba CP-001	55
Tabla 10. Caso de prueba CP-002	55
Tabla 11. Caso de prueba CP-003	56
Tabla 12. Caso de prueba CP-004	56
Tabla 13. Caso de prueba CP-005	57
Tabla 14. Caso de prueba CP-006	57
Tabla 15. Caso de prueba CP-007	58
Tabla 16. Caso de prueba CP-008	58

# 1. Introducción

---

El presente capítulo describirá sobre la problemática a resolver y el impacto que tendrá una vez solventada, asimismo los objetivos que se alcanzaran al terminar el desarrollo del presente proyecto.

## 1.1. Descripción del problema o necesidad

Actualmente la empresa Papelería Ingrid cuenta con un control de inventario ineficaz dado que es realizado por el dueño mediante una libreta, provocando que la información no sea totalmente veraz, por los errores humanos debido al cansancio, además de comparar las ventas del día con el inventario actual una por una (venta), para averiguar que las ventas coincidan o descubrir las pérdidas del día. Por otro lado, la falta de reportes en el área de compra/venta de los productos existentes afecta de igual manera al inventario encaminando a un control inexacto del mismo afectando a la adquisición de los productos, puesto que lleva a una colección de un mismo producto o a una decadencia de otro.

## 1.2 Justificación

Los sistemas de punto de venta, en la actualidad no son un lujo, sino una necesidad para la automatización, correlación y sistematización de los procesos de salida y cobro de la mercancía en diversos tipos de establecimientos. Dado la capacidad limitada de las cajas registradoras y la memoria del cerebro humano para el almacenamiento de información sobre colecciones de datos, sin embargo, los sistemas de punto de venta logran reunir, guardar, y generar reportes detallados sobre las compras, ventas o de los productos mejor vendidos en determinadas fechas. Por lo tanto, el software propuesto busca mediante la aplicación de herramientas de desarrollo como los leguajes de programación, agilizar y facilitar las diferentes tareas realizadas manualmente en la papelería, mejorando el control en las áreas de compra/venta e inventario dejando de depender de una libreta como base de datos, además, añadiendo la ventaja del sistema para calcular el total a cobrar por la venta de los diferentes artículos, el cual es un problema mayor al surtir listas escolares dada la cantidad de artículos los cuales son solicitados.

## **1.3 Objetivo General y Específicos**

### **El objetivo general**

Implementar un sistema de punto de venta para Papelería Ingrid y así lograr la estabilidad de la empresa.

### **Objetivos específicos**

- Analizar los requerimientos funcionales para la empresa Papelería Ingrid
- Diseñar una interfaz para un punto de venta
- Desarrollar prototipos para el punto de venta
- Desarrollar y un punto de venta
- Realizar pruebas de usabilidad

## 2. Metodología y herramientas

---

En el presente capítulo se abordará la metodología a usar en el desarrollo del proyecto, describiendo cada una de sus fases, así como las herramientas de apoyo listando ventajas y desventajas.

### 2.1. Metodología de prototipos

La metodología de prototipos, pertenece a los modelos de desarrollo evolutivo. El prototipo se basa en un principio que dicta: “Ser construido en poco tiempo, usando programas necesarios sin la necesidad de utilizar muchos recursos”

Dado la obtención de un diseño rápido, el cual permite visualizar los aspectos del software para el usuario final. Llevando a una construcción de un prototipo, el cual será evaluado para su posterior retroalimentación y mejora del mismo.  
[1]

El modelo se aplica cuando el cliente no delimita los requisitos de entrada procesamiento y salida, es decir, cuando el cliente no está seguro de la forma de interacción del hombre y la máquina. La función del modelo es ayudar al ingeniero de sistemas y al cliente a entender de mejor manera el resultado de la construcción cuando los requisitos estén satisfechos.

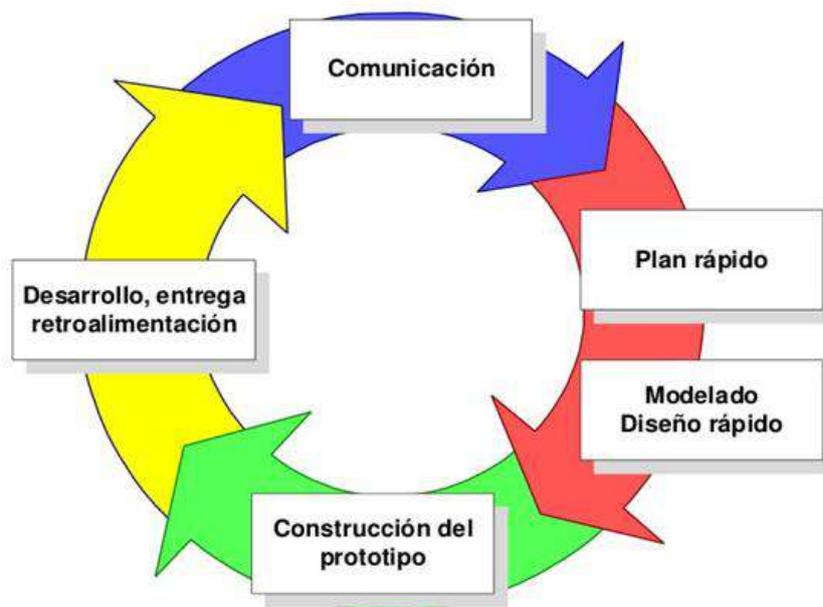
El diseño rápido surge de la recolección de requisitos dados por el usuario final debido a que se definen los objetivos globales para el software y al mismo tiempo se identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición

El diseño rápido lleva a la construcción de un prototipo. El prototipo es evaluado por el usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

El paradigma de creación de prototipos puede ser cerrado o abierto. El enfoque cerrado se denomina a menudo prototipo desechable. Este prototipo sirve únicamente como una basta demostración de los requisitos. Después se desecha y se hace una ingeniería del software con un paradigma diferente. Un enfoque abierto, denominado prototipo evolutivo, emplea el prototipo como primera parte de una actividad de análisis a la que seguirá el diseño y la construcción. El prototipo del software es la primera evolución del sistema terminado.

Antes de elegir un enfoque es necesario determinar si se puede crear un prototipo del sistema a construir. Se pueden definir varios factores candidatos a la creación de prototipos área de aplicación, complejidad, características del cliente y del proyecto.

En el proyecto “Desarrollo de un punto de venta para la gestión de una papelería” se utilizará el modelo de prototipo abierto o prototipo evolutivo ya que se basa en la idea de desarrollar de acuerdo a los comentarios del usuario y al final refinando los detalles a través de diferentes versiones para un software adecuado para el cliente.



**Figura 1.** Etapas de la metodología por prototipo [2]

Como se muestra en la figura 1 la metodología de prototipo consta de 5 etapas como se muestran en el esquema las cuales son:

1. **Definición de los requisitos (comunicación).** Proceso para identificar las necesidades y requerimientos del cliente con respecto al software.
2. **Análisis y Diseño (plan rápido).** Establecer la viabilidad del software desde el punto de vista técnico y económico, planificando las actividades y el presupuesto. El diseño del software se centra en cuatro atributos de un programa: estructura de datos, arquitectura del software, representaciones de interfaz y detalle procedimental.
3. **Codificación (construcción del prototipo):** Traducir en forma legible por la máquina el modelo previamente diseñado. El paso de generación de código lleva a cabo esta tarea. Si lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente.
4. **Pruebas.** Se centra en los procesos lógico internos del software, y en los procesos externos funcionales. Se deben realizar las pruebas para detección de errores y asegurarse que las entradas definidas produzcan resultados reales que concuerden con los resultados requeridos.
5. **Mantenimiento(etroalimentación).** El software indudablemente sufrirá cambios después de ser entregado al cliente. El mantenimiento vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo. [4]

## 2.2. Herramientas

A continuación, se presentan las características, ventajas y desventajas de cada una de las herramientas usadas en el desarrollo del presente proyecto.

### 2.2.1. Sublime Text 3

Sublime Text es un editor de código multiplataforma, ligero y gratuito, el cual su principal función es permitir la interpretación del código y la ejecución del mismo, dependiendo del lenguaje que se opte para escribir en él. Siendo una herramienta para programar sin distracciones por su interfaz sencilla. [11]

#### Características

- Simplicidad
  1. Interfaz sencilla e intuitiva.
  2. Colores para identificar código.
- Plugins
  1. amplia colección de plugins.
- Compatibilidad
  2. Sublime Text 3 es flexible soportando lenguajes de programación como C y Python, por el lado web, CSS, JS, PHP.

#### Ventajas

- Incluye un panel que permite moverse por el código de forma rápida (minimap).
- Resaltado de sintaxis para el código.
- Permite tener varios documentos abiertos mediante pestañas y/o paneles.

#### Desventajas

- Fallos en la versión gratuita

### 2.2.2. Hojas de estilo en cascada o Cascading Style Sheets (CSS)

**CSS** es un lenguaje de hojas de estilo en cascada, su principal función es estilizar las etiquetas **HTML** y su contenido. CSS además separa el contenido de la representación visual del sitio. [9]

## **Ventajas**

1. Ahorra esfuerzo y tiempo de edición para cada elemento HTML.
2. Se evita tener que recurrir a trucos para conseguir algunos efectos.
3. Sencillo y fácil de aprender.
4. Los documentos que usan CSS generalmente resultan más compactos.
5. Pueden usarse con otros lenguajes de programación (como JavaScript) para conseguir efectos dinámicos en las páginas.
6. Se pueden especificar Hojas de Estilo para diferentes navegadores y tipos de medios (impresos, braille, auditivos, etc.).

## **Desventajas**

1. Incompatibilidad por parte de los navegadores. Esto puede provocar que:
2. Nuestra página sea visualizada por el lector con un formato no deseado por nosotros.
3. Algunas propiedades de las CSS (como las que afectan la posición o visibilidad de los elementos) pueden provocar que una parte del contenido de nuestra página resulte inaccesible desde ciertos navegadores si no son utilizadas correctamente.

### **2.2.3. JavaScript**

JavaScript es un lenguaje de programación aplicable a un documento HTML con la finalidad de crear una interacción en los sitios web.[5]

## **Ventajas**

1. Lenguaje sencillo de aprender.
2. Ejecuta funciones inmediatamente.
3. Múltiples opciones de efectos visuales.
4. Tiene soporte para navegadores modernos (Mozilla, Opera, Chrome)
5. Desarrollo de páginas dinámicas y aplicaciones web.
6. Permite la validación de datos en un formulario.
7. Permite trabajar modo FullStack en el desarrollo de programación.

## **Desventajas**

1. En el FrontEnd sus códigos son visibles, por lo tanto, pueden ser leídos por cualquier usuario.
2. Tiende a introducir gran cantidad de fragmentos de código en los sitios web.
3. No es compatible en todos los navegadores de manera uniforme.
4. Los usuarios tienen la opción de desactivar JavaScript desde su navegador.
5. Sus scripts son limitados por razones de seguridad y no es posible realizar todo con JavaScript, por lo tanto, es necesario complementarlo con otros lenguajes.

### **2.2.4. PHP**

*“PHP corresponde a las iniciales de personal home page tools (herramientas para páginas iniciales personales). Es un lenguaje de programación tipo script para entornos web con unas funciones muy semejantes a las de ASP y JSP, utilizado, sobre todo, en servidores Linux para personalizar la información enviada a los usuarios que acceden a un sitio web.” [6]*

*“PHP es un lenguaje de programación estructurado con extensiones de orientación a objetos (OO).” [6]*

#### **Características**

1. Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
2. Fácil de aprender
3. Programación segura y confiable.
4. Capacidad de conexión con motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
5. Capacidad de expandir su potencial utilizando módulos
6. Amplia documentación en su sitio web oficial.
7. Permite aplicar técnicas de programación orientada a objetos.
8. No requiere definición de tipos de variables.

## **Ventajas**

1. Lenguaje totalmente libre y abierto.
2. Posee una curva de aprendizaje muy baja.
3. Los entornos de desarrollo son de rápida y fácil configuración.
4. Fácil de instalar.
5. Fácil acceso e integración con las bases de datos.
6. Es un lenguaje multiplataforma.
7. Completamente orientado al desarrollo de aplicaciones web dinámicas y/o páginas web con acceso a una Base de Datos.
8. El código escrito en PHP es invisible al navegador ya que se ejecuta al lado del servidor y los resultados en el navegador es HTML.
9. Posee una versatilidad para la conexión con la mayoría de base de datos que existen en la actualidad.

## **Desventajas**

1. El inconveniente es que el código fuente no pueda ser ocultado de una manera eficiente. La ofuscación es una técnica que puede dificultar la lectura del código y, en ciertos aspectos representa tiempos de ejecución.
2. Nuestro código estará seguro para ejecutar si es nuestro propio servidor. Por lo tanto, si un cliente requiere su código en su pc, tendríamos que dejar el código fuente, sin manera de ocultarlo.
3. Si no lo configuras y/o proteges correctamente dejas abiertas muchas brechas de seguridad que a la larga tendremos problemas.
4. Sólo se ejecuta en un servidor y se necesita un servidor web para que funcione.

### **2.2.5. XAMPP**

*“MySQL es un sistema de gestión de bases de datos (SGBD) SQL. Es un SGBD sencillo y rápido que se adapta perfectamente a entornos en los que el volumen de datos sea del orden de megabytes.” [6]*

## **Características**

- Alojamiento web

1. Proceso de instalación de XAMPP es muy sencillo y rápido.
  2. Adquisición de un servidor local
  3. Entorno adecuado para comprobar aplicaciones MYSQL, PHP, Apache y Perl en una procesadora local.
- Herramientas de XAMPP
    1. Apache
      - Software totalmente gratuito de código abierto.
      - MYSQL
      - Método de gestión de bases de registros relacionales.
    2. PHP
      - Este lenguaje se usa especialmente para desarrollar aplicaciones de software fundadas en web. Es un software de código abierto y actúa bien con MYSQL.
    3. Perl
      - Desarrollo web, adelanto de GUI, gestión de sistemas, entre otros. Perl es competente de trabajar con HTML, XML y otras expresiones de marcas.

### **Ventajas**

- Fácil de estudiar
- Configuraciones mínimas

### **Desventajas**

- No soporta MySQL desde la consola para tareas específicas.
- No se pueden actualizar individualmente las versiones de los programas que instala.
- Dificultad para configurar aplicaciones de terceros.

### **2.2.6. FPDF**

#### **¿Qué es FPDF?**

FPDF es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP, sin necesidad de otras librerías. [12]

## **Ventajas**

- Elección de la unidad de medida, formato de página y márgenes
- Edición de cabeceras y pies de página
- Salto de página automático
- Salto de línea y justificación del texto automáticos
- Admisión de imágenes (JPEG, PNG)

## **Desventajas**

- Incompatibilidad con versiones inferiores de PHP

# 3. Resultados

En el presente capítulo se presentarán los productos generados en cada una de las etapas de la metodología.

## 3.1. Definición de los requisitos

### Requerimientos detallados

La figura 2 muestra el caso de uso del sistema en general, es decir, los módulos a desarrollar en el punto de venta.

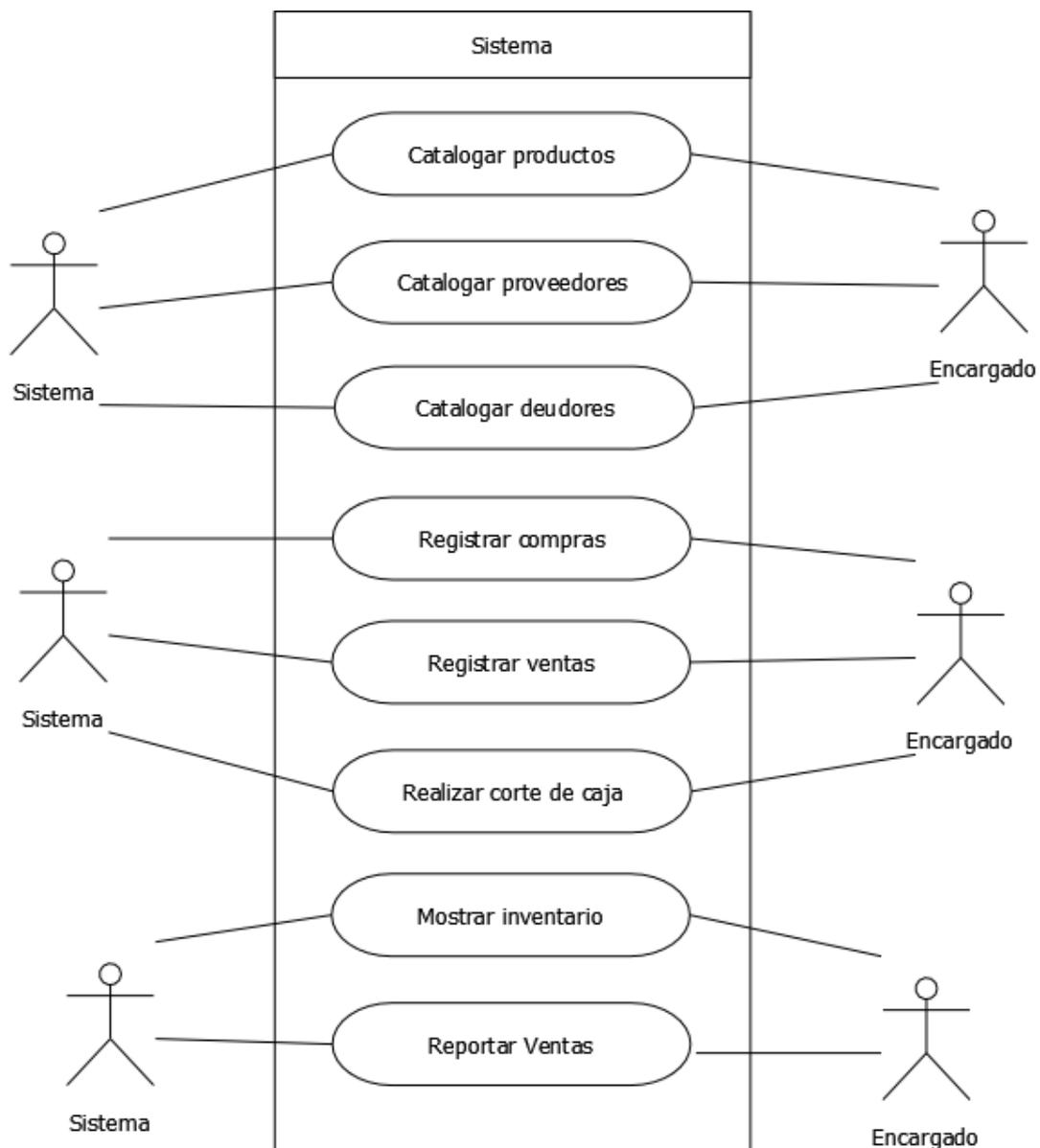
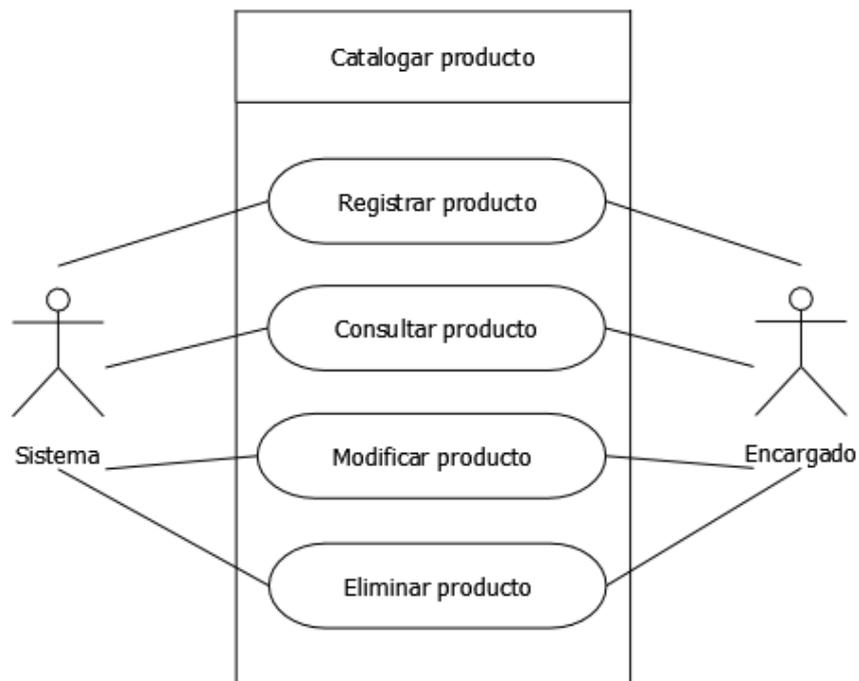


Figura 2. Caso de uso general

La figura 3 muestra el caso de uso del catálogo productos, es decir, los módulos a desarrollar en el presente catálogo.



**Figura 3.** Caso de uso catalogar producto

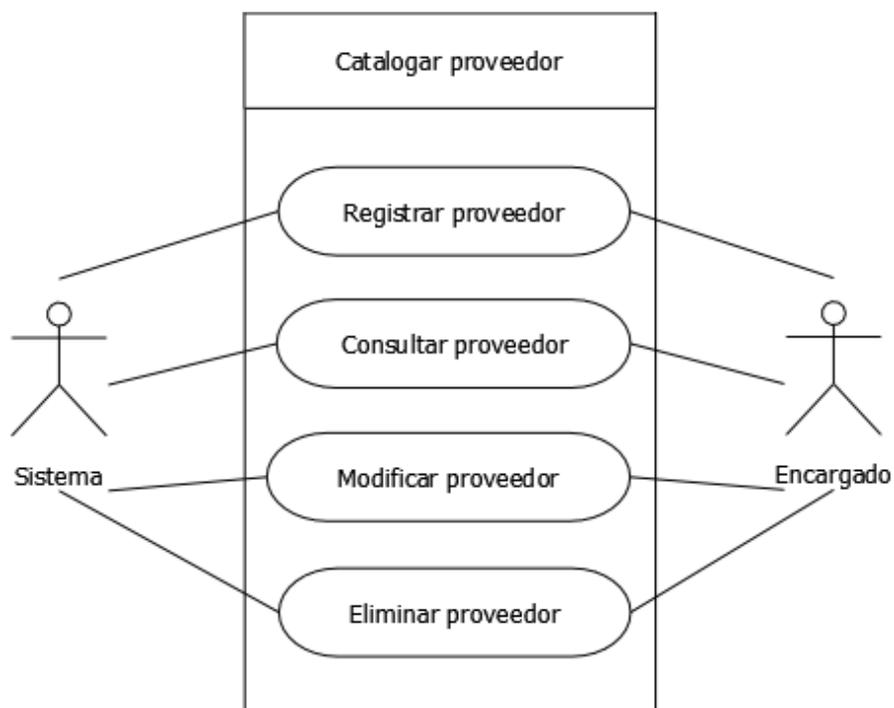
En la tabla 1 se muestra las descripciones paso a paso de lo que se debe de realizar para cada una de las funciones del módulo de productos.

<b>Caso de uso</b>	<b>Catalogar producto</b>
<b>Autores</b>	Encargado, Sistema
<b>Descripción paso a paso</b>	<ol style="list-style-type: none"> <li>1. Registrar producto               <ol style="list-style-type: none"> <li>1.1. El encargado teclea los datos del producto</li> <li>1.2. El sistema guarda la información</li> </ol> </li> <li>2. Consultar producto               <ol style="list-style-type: none"> <li>2.1. El encargado teclea el código de barra</li> <li>2.2. El sistema muestra la información del producto</li> </ol> </li> <li>3. Modificar producto               <ol style="list-style-type: none"> <li>3.1. El encargado selecciona el producto a modificar</li> </ol> </li> </ol>

	<p>3.2. El sistema muestra datos del producto</p> <p>3.3. El encargado modifica los datos del producto</p> <p>3.4. El sistema guarda los cambios</p> <p>4. Eliminar producto</p> <p>4.1. El encargado selecciona el producto a eliminar</p> <p>4.2. El sistema muestra el producto</p> <p>4.3. El encargado confirma el producto a eliminar</p> <p>4.4. El sistema elimina el producto</p>
--	--

**Tabla 1.** Caso de uso detallado catalogar producto

La figura 4 muestra el caso de uso del catálogo proveedores, es decir, los módulos a desarrollar en el presente catálogo.



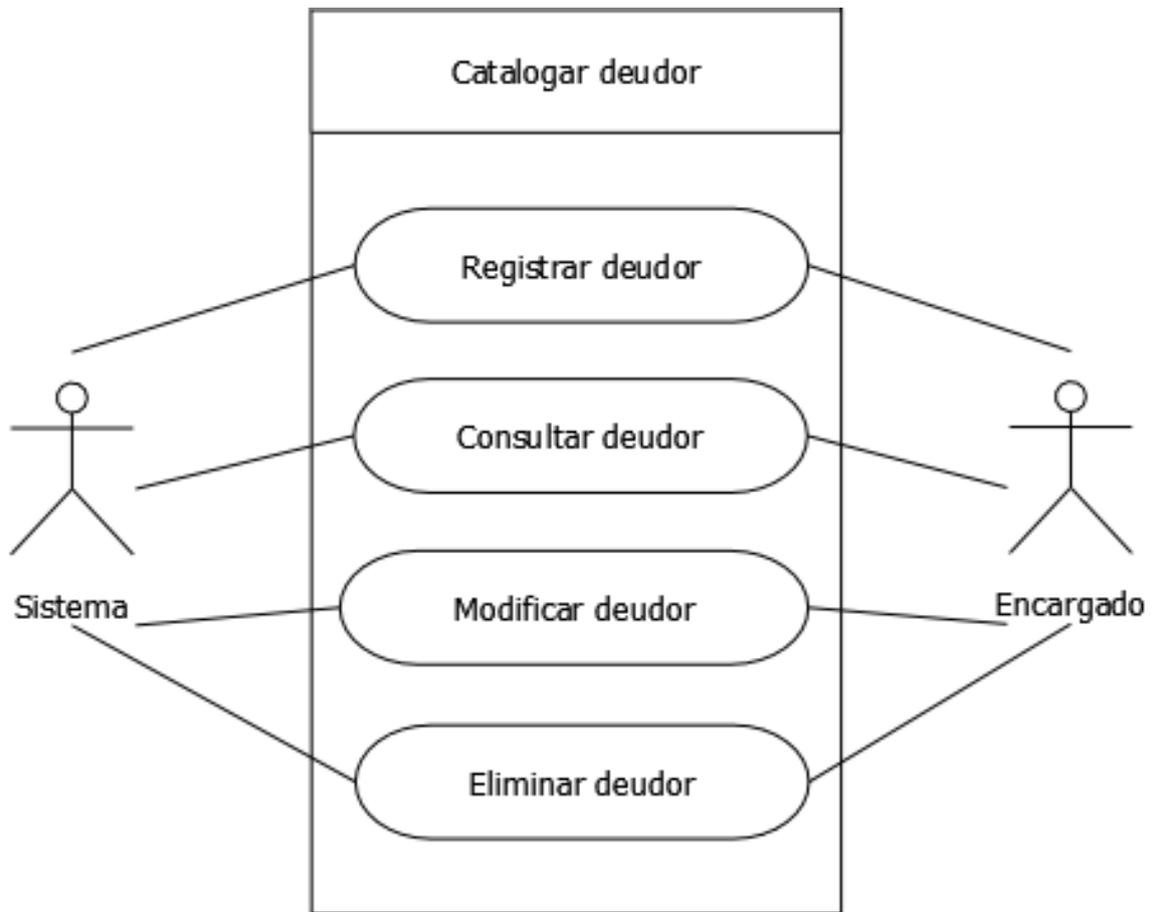
**Figura 4.** Caso de uso catalogar proveedor

En la tabla 2 se muestra las descripciones paso a paso de lo que se debe de realizar para cada una de las funciones del módulo de proveedor.

<b>Caso de uso</b>	<b>Catalogar proveedor</b>
<b>Autores</b>	Encargado, Sistema
<b>Descripción paso a paso</b>	<ol style="list-style-type: none"> <li>1. Registrar proveedor <ol style="list-style-type: none"> <li>1.1. El encargado teclea los datos del proveedor</li> <li>1.2. El sistema guarda la información</li> </ol> </li> <li>2. Consultar proveedor <ol style="list-style-type: none"> <li>2.1. El encargado teclea el ID de proveedor.</li> <li>2.2. El sistema muestra la información del proveedor</li> </ol> </li> <li>3. Modificar proveedor <ol style="list-style-type: none"> <li>3.1. El encargado selecciona el proveedor a modificar</li> <li>3.2. El sistema muestra datos del proveedor</li> <li>3.3. El encargado modifica los datos del proveedor</li> <li>3.4. El sistema guarda los cambios</li> </ol> </li> <li>4. Eliminar proveedor <ol style="list-style-type: none"> <li>4.1. El encargado selecciona el proveedor a eliminar</li> <li>4.2. El sistema muestra el proveedor</li> <li>4.3. El encargado confirma el proveedor a eliminar</li> <li>4.4. El sistema elimina el proveedor</li> </ol> </li> </ol>

**Tabla 2.** Caso de uso detallado catalogar proveedor

La figura 5 muestra el caso de uso del catálogo deudores, es decir, los módulos a desarrollar en el presente catálogo.



**Figura 5.** Caso de uso catalogar deudor

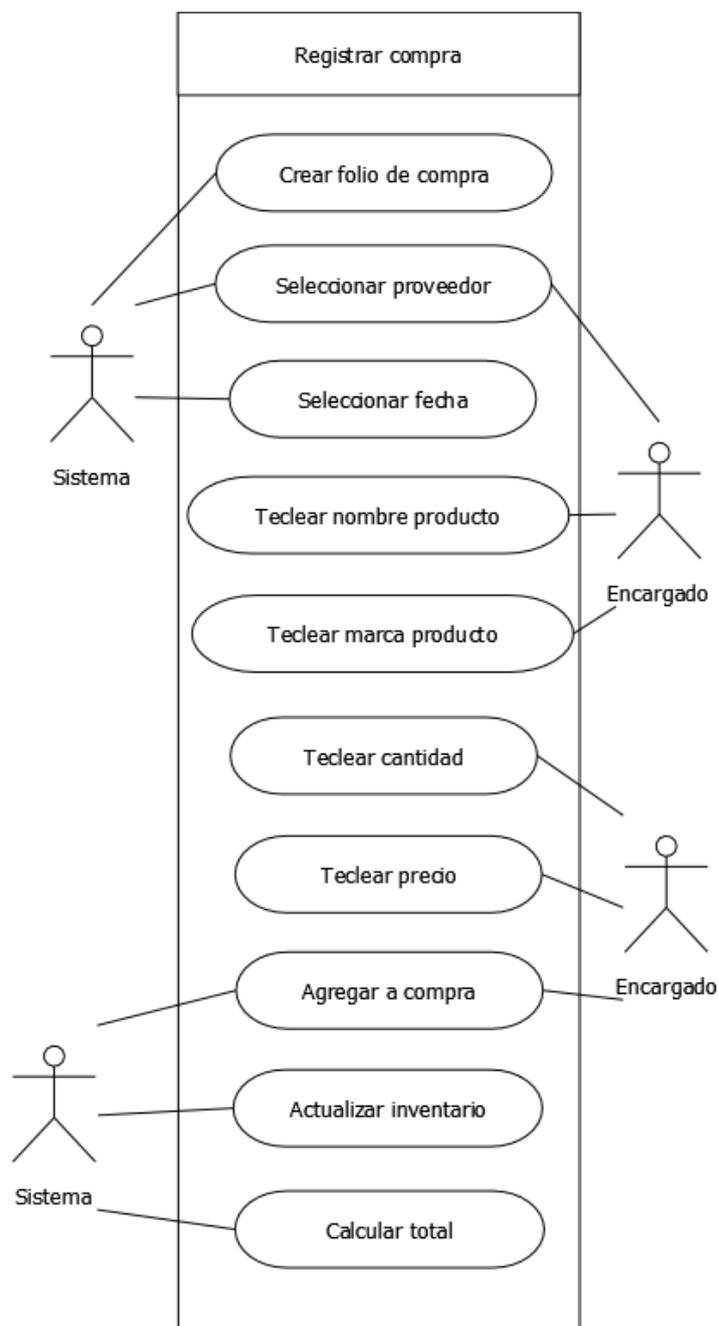
En la tabla 3 se muestra las descripciones paso a paso de lo que se debe de realizar para cada una de las funciones del módulo de deudor.

<b>Caso de uso</b>	<b>Catalogar deudor</b>
<b>Autores</b>	Encargado, Sistema
<b>Descripción paso a paso</b>	<ol style="list-style-type: none"> <li>1. Registrar deudor               <ol style="list-style-type: none"> <li>1.1. El encargado teclea los datos del deudor</li> <li>1.2. El sistema guarda la información</li> </ol> </li> <li>2. Consultar deudor               <ol style="list-style-type: none"> <li>2.1. El encargado teclea el ID de deudor.</li> <li>2.2. El sistema muestra la información del deudor</li> </ol> </li> <li>3. Modificar deudor               <ol style="list-style-type: none"> <li>3.1. El encargado selecciona el deudor a modificar</li> </ol> </li> </ol>

	<ul style="list-style-type: none"> <li>3.2. El sistema muestra datos del deudor</li> <li>3.3. El encargado modifica los datos del deudor</li> <li>3.4. El sistema guarda los cambios</li> <li>4. Eliminar deudor <ul style="list-style-type: none"> <li>4.1. El encargado selecciona el deudor a eliminar</li> <li>4.2. El sistema muestra el deudor</li> <li>4.3. El encargado confirma el deudor a eliminar</li> <li>4.4. El sistema elimina el deudor</li> </ul> </li> </ul>
--	---

**Tabla 3.** Caso de uso detallado catalogar deudor

La figura 6 muestra el caso de uso del módulo registrar compra, es decir, las funciones a desarrollar en el presente módulo.



**Figura 6.** Caso de uso registrar compra

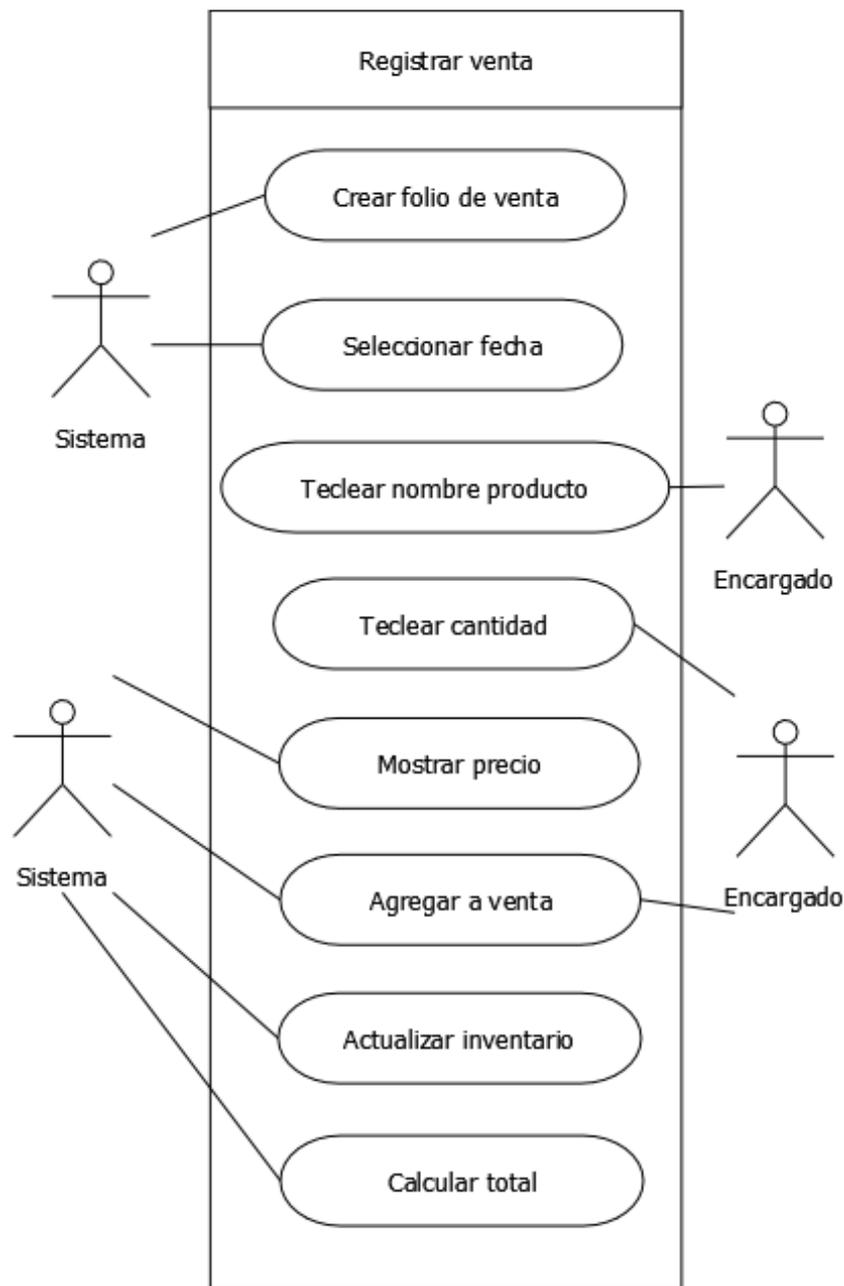
En la tabla 4 se muestra las descripciones paso a paso de lo que se debe de realizar para cada una de las funciones que ofrece el módulo de compras para un uso óptimo.

<b>Caso de uso</b>	<b>Registrar compra</b>
<b>Autores</b>	Encargado, Sistema
<b>Descripción paso a paso</b>	1. El sistema crea folio de compra 2. El encargado selecciona el proveedor

	<ol style="list-style-type: none"> <li>3. El sistema selecciona la fecha</li> <li>4. El encargado teclea el nombre del producto</li> <li>5. El encargado teclea la marca del producto</li> <li>6. El encargado tecla la cantidad del producto</li> <li>7. El encargado teclea el precio del producto</li> <li>8. El encargado y el sistema agregan a compra</li> <li>9. El sistema actualiza el inventario</li> <li>10. El sistema calcula el total</li> </ol>
--	--

**Tabla 4.** Caso de uso detallado registrar compra

La figura 7 muestra el caso de uso del módulo registrar venta, es decir, las funciones a desarrollar en el presente módulo.



**Figura 7.** Caso de uso registrar venta

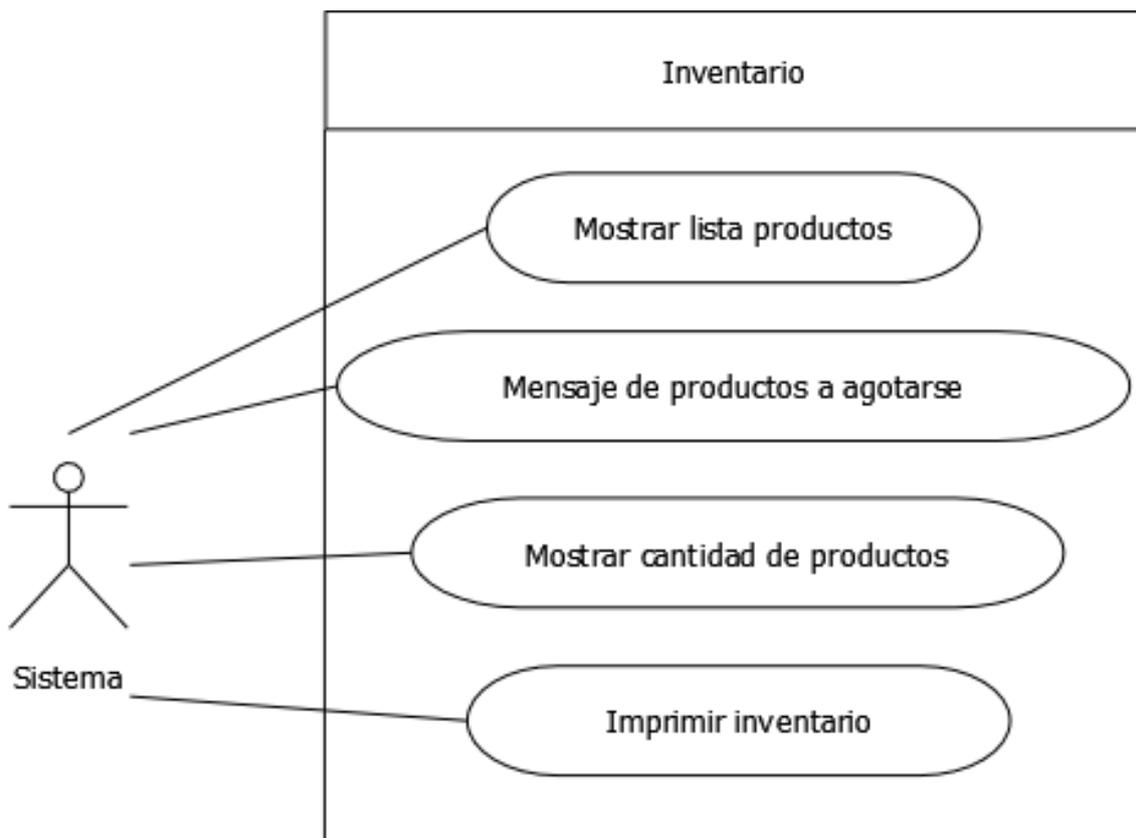
En la tabla 5 se muestra las descripciones paso a paso de como usar el módulo de ventas correctamente.

<b>Caso de uso</b>	<b>Registrar venta</b>
<b>Autores</b>	Encargado, Sistema
<b>Descripción paso a paso</b>	<ol style="list-style-type: none"> <li>1. El sistema crea folio de venta</li> <li>2. El sistema selecciona la fecha</li> <li>3. El encargado selecciona el producto</li> <li>4. El sistema muestra información del</li> </ol>

	producto como precio y marca 5. El encargado teclea la cantidad 6. El encargado y el sistema agregan a venta 7. El sistema actualiza el inventario 8. El sistema calcula el importe 9. El sistema calcula el total
--	---

**Tabla 5.** Caso de uso detallado registrar venta

La figura 8 muestra el caso de uso del módulo inventario, es decir, las funciones a desarrollar en el presente módulo.



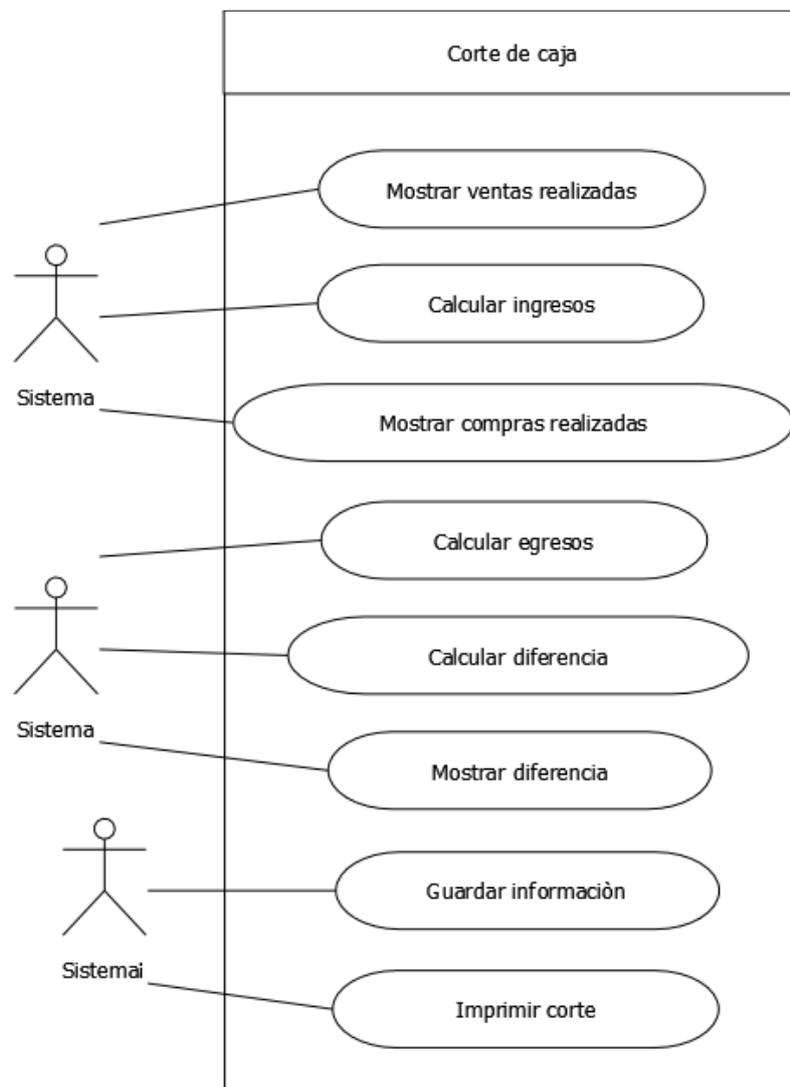
**Figura 8.** Caso de uso inventario

En la tabla 6 se muestra las descripciones paso a paso de lo que el sistema realiza en el módulo de inventario.

Caso de uso	Inventario
<b>Autores</b>	Sistema
<b>Descripción paso a paso</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra una lista de los productos</li> <li>2. El sistema notifica al encargado de los productos a agotarse</li> <li>3. El sistema muestra la cantidad de productos.</li> <li>4. El sistema imprime el inventario</li> </ol>

**Tabla 6.** Caso de uso detallado inventario

La figura 9 muestra el caso de uso del módulo corte de caja, es decir, las funciones a desarrollar en el presente módulo.



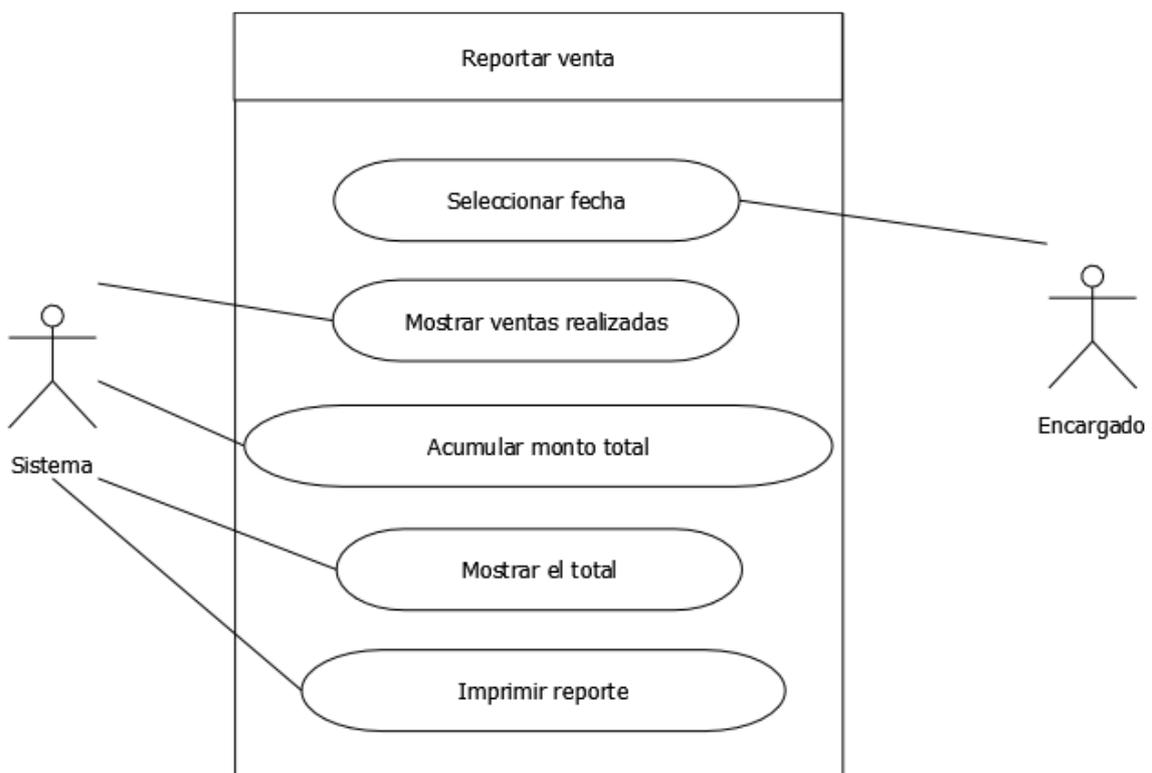
**Figura 9.** Caso de uso corte de caja

En la tabla 7 se muestra las descripciones paso a paso de lo que el sistema realiza en el módulo de corte de caja.

Caso de uso	Corte de caja
Autores	Sistema
Descripción paso a paso	<ol style="list-style-type: none"> <li>1. El sistema muestra las ventas realizadas en el día</li> <li>2. El sistema calcula los ingresos</li> <li>3. El sistema muestra las compras realizadas en el día</li> <li>4. El sistema calcula los egresos</li> <li>5. El sistema calcula la diferencia</li> <li>6. El sistema muestra la diferencia</li> <li>7. El sistema guarda la información</li> <li>8. El sistema imprime el corte</li> </ol>

**Tabla 7.** Caso de uso detallado corte de caja

La figura 10 muestra el caso de uso del módulo reportar venta, es decir, las funciones a desarrollar en el presente módulo.



**Figura 10.** Caso de uso reportar venta

En la tabla 8 se muestra las descripciones paso a paso de lo que el encargado y el sistema deben de realizar para generar los reportes de venta.

<b>Caso de uso</b>	<b>Reportar venta</b>
<b>Autores</b>	Encargado, Sistema
<b>Descripción paso a paso</b>	<ol style="list-style-type: none"> <li>1. El encargado selecciona una fecha</li> <li>2. El sistema muestra las ventas realizadas en esa fecha</li> <li>3. El sistema calcula el monto total</li> <li>4. El sistema muestra el total</li> <li>5. El sistema imprime el reporte</li> </ol>

**Tabla 8.** Caso de uso detallado reportar venta

### **3.2. Análisis y Diseño**

En la etapa de análisis se identificaron los problemas y/o necesidades que el cliente requiere en los módulos del sistema de punto de venta.

- **Falta de organización**

Control inexacto de la información de los materiales que ofrece la papelería, debido a que no es posible actualizar eficazmente el inventario del establecimiento. Por lo tanto, el proceso para localizar algún material dentro del almacén es aburrido, costoso y tedioso.

- **Inexistencia de reportes**

La falta de reportes en la compra/venta de los productos existentes afecta de igual manera al inventario encaminando a un control inexacto del inventario afectando a la adquisición de los productos, puesto que lleva a una colección de un mismo producto o a una decadencia de otro.

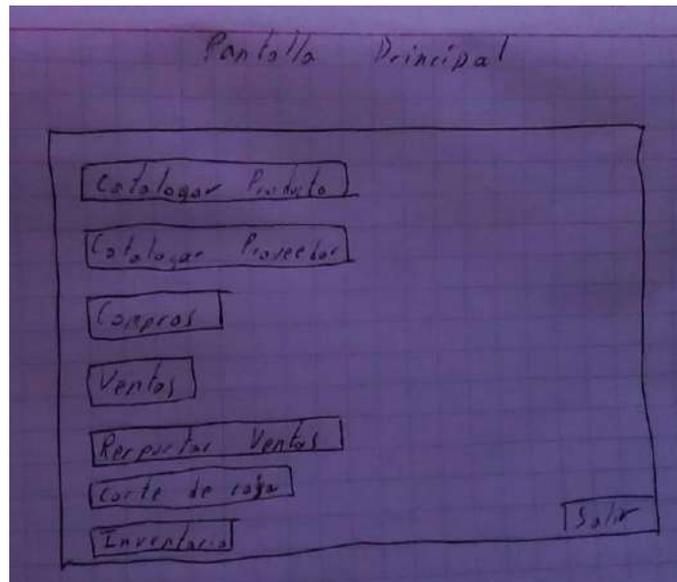
- **Publicidad**

En un mundo tecnológico donde la publicidad en línea ha aumentado, llevando a los carteles físicos a ser obsoletos para atraer a clientes nuevos a consumir los productos que ofrece la empresa para obtener mayores ganancias.

## Diagramas de baja fidelidad

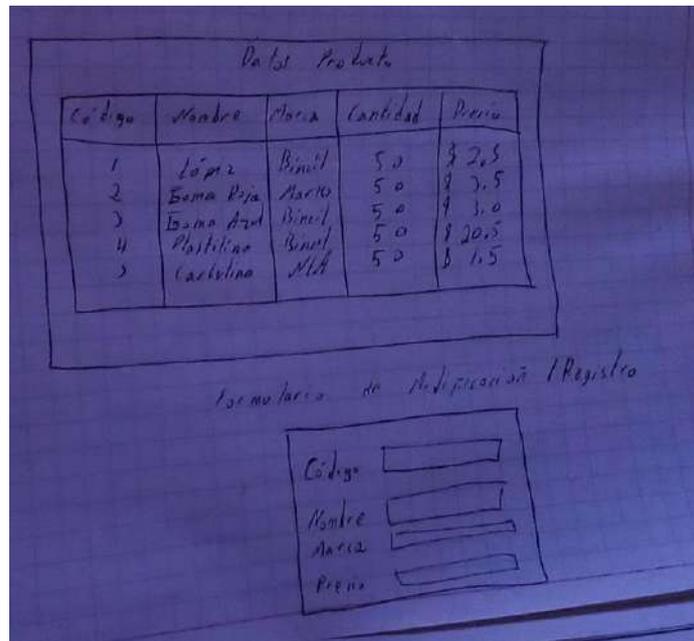
Los diagramas de baja fidelidad, son una traducción simple de los conceptos de diseño. A menudo se usa para convertir las ideas de diseño en artefactos tangibles para recopilar y analizar las demandas del usuario en la etapa inicial.

La figura 11 muestra un dibujo a lápiz de la posible interfaz principal del sistema a desarrollar como aplicación de escritorio.



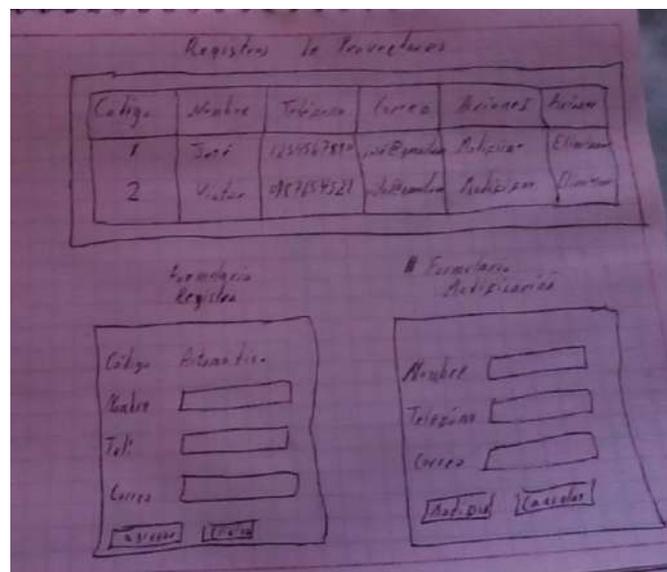
**Figura 11.** Diseño de baja fidelidad Pantalla de inicio

La figura 12 muestra un dibujo a lápiz de diseño de interfaz en el catálogo de productos del sistema además del diseño de los formularios de registro y modificación del mismo.



**Figura 12.** Diseño de baja fidelidad Pantalla de producto y formulario de modificación/registro

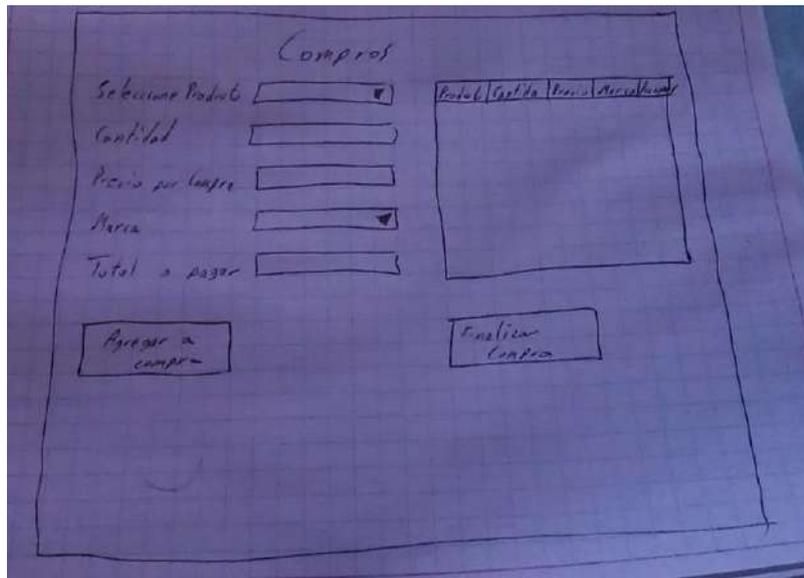
La figura 13 muestra un dibujo a lápiz de diseño de interfaz en el catálogo de proveedores del sistema además del diseño de los formularios de registro y modificación del mismo.



**Figura 13.** Diseño de baja fidelidad Pantalla de proveedor y formulario de modificación/registro

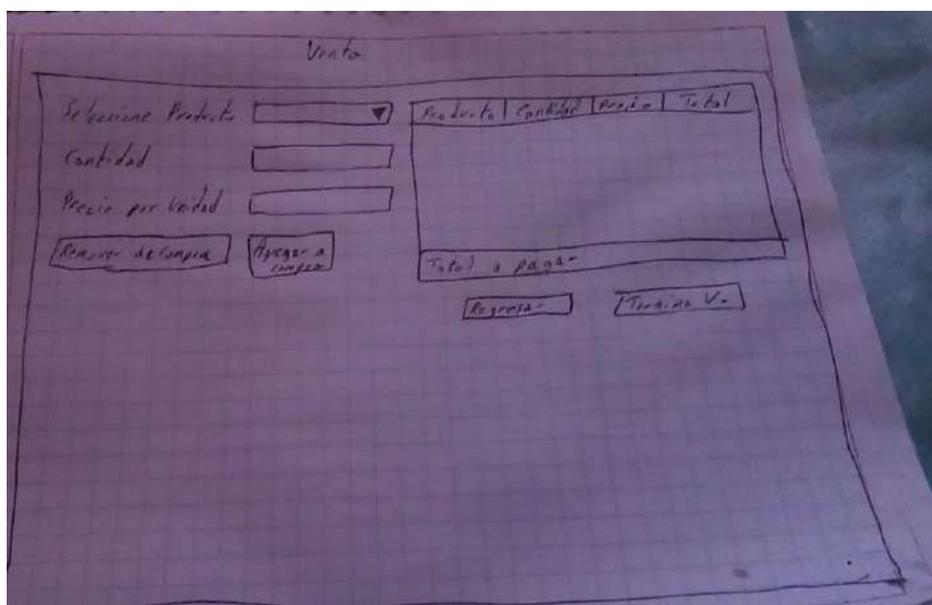
La figura 14 contiene un dibujo a lápiz de diseño de interfaz en el área de compras de productos del sistema, dividida en dos secciones una de registro

de compras, la otra es una tabla donde se agrega los productos a comprar al proveedor.



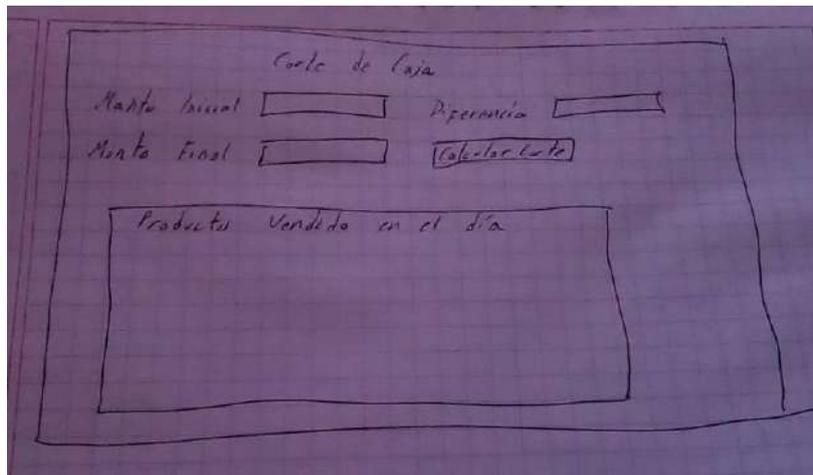
**Figura 14.** Diseño de baja fidelidad Pantalla de compras

La figura 15 contiene un dibujo a lápiz de diseño de interfaz en el área de ventas de productos del sistema, dividida en dos secciones una de registro de ventas, la otra es una tabla donde se agrega los productos a vender al cliente.



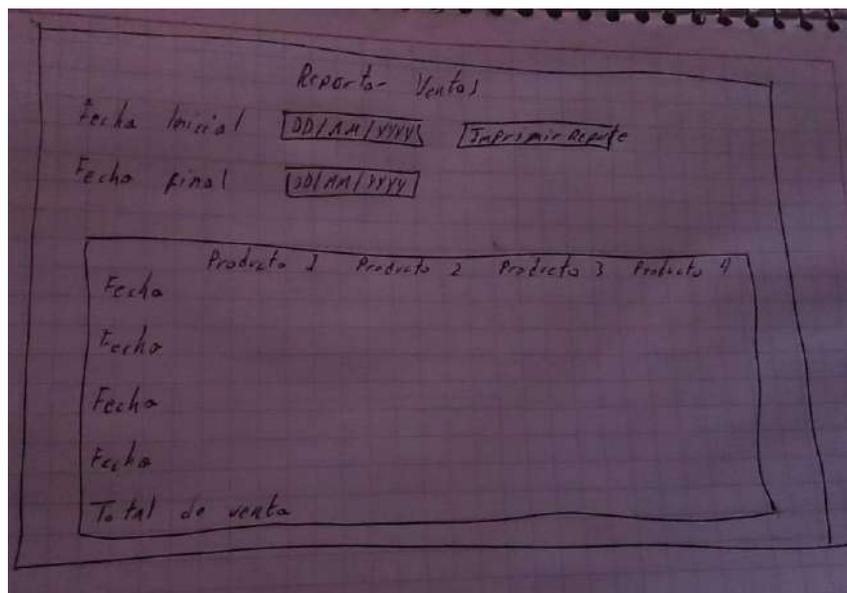
**Figura 15.** Diseño de baja fidelidad Pantalla de ventas

La figura 16 contiene un dibujo a lápiz de diseño de interfaz en el área de corte de caja.



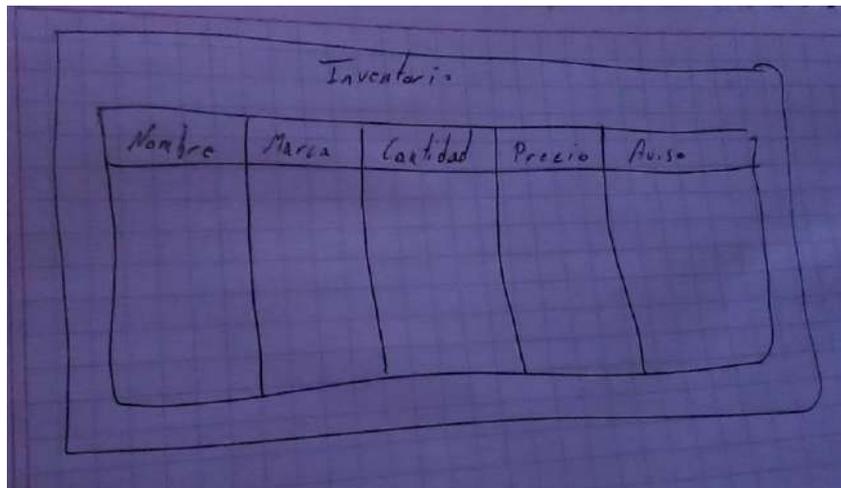
**Figura 16.** Diseño de baja fidelidad Pantalla de corte de caja

La figura 17 muestra un dibujo a lápiz de diseño de interfaz en el área de reportar ventas de productos del sistema, la cual mostrará un informe detallado por día.



**Figura 17.** Diseño de baja fidelidad Pantalla de reporte de ventas

La figura 18 muestra un dibujo a lápiz de diseño de interfaz en el área de inventario, listando los nombres, marca, cantidad y precio de los productos en existencia además de una columna de aviso en caso de que el producto se esté agotando.

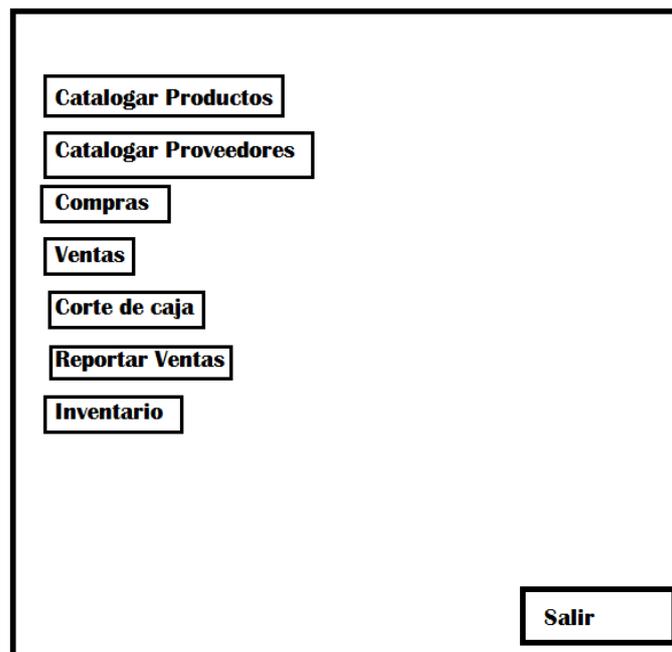


**Figura 18.** Diseño de baja fidelidad Pantalla de inventario

### Diagramas de media fidelidad

Después de haber corregido y mejorado el prototipo de baja fidelidad, considerando la retroalimentación del usuario. De esta manera, se desarrolla un prototipo de media fidelidad en el cual define diseño, tipografía, tamaños y formas, con el fin de hacer más amena y agradable visualmente el producto.

La figura 19 proyecta un dibujo sobre un diseño realista a la pantalla de inicio del punto de venta, en este caso siendo la pantalla principal del sistema.



**Figura 19.** Diseño de media fidelidad Pantalla de inicio

La figura 20 proyecta un diseño mejor definido en el catálogo de productos en el cual se listarán los datos de los productos en existencia, además de contar con un botón de “agregar producto”.

Datos producto						
Código	Nombre	Marca	Cantidad	Precio	Acciones	

**Agregar producto**

**Figura 20.** Diseño de media fidelidad Pantalla de productos

La figura 21 proyecta un diseño definido en los formularios de registro y modificación de los productos, en el formulario de modificación no podrá ser editado el código del producto.

**Formulario de registro**

**Código**

**Nombre**

**Marca**

**Precio**

**Agregar** **Cancelar**

**Formulario de modificaciones**

**Nombre**

**Marca**

**Precio**

**Modificar** **Cancelar**

**Figura 21.** Diseño de media fidelidad formularios de registro y modificación

La figura 22 proyecta un diseño mejor definido en el catálogo de proveedores, así como las posibles acciones permitidas a realizar en algún registro.

Registros Proveedores					
Código	Nombre	Teléfono	Correo	Acciones	
				Modificar	Eliminar

**Figura 22.** Diseño de media fidelidad Pantalla de proveedores

La figura 23 proyecta un diseño definido en los formularios de registro y modificación de los productos, al igual que con los formularios de producto no se permite la edición del campo código.

**Formulario de registro**

**Código**

**Nombre**

**Teléfono**

**Correo**

**Formulario de modificaciones**

**Nombre**

**Teléfono**

**Correo**

**Figura 23.** Diseño de media fidelidad formularios de registro y modificación proveedor

La figura 24 proyecta un diseño definido en el módulo de compras del proyecto con las opciones de agregar y remover productos de la compra a realizar, además cuenta con un campo no editable del total a pagar.

**Compra**

**Seleccione un producto**

**Cantidad**

**Precio**

**Remover de compra** **Agregar a venta**

**Total a pagar**

Producto	Cantidad	Precio	Total

**Regresar**

**Finalizar Venta**

**Figura 24.** Diseño de media fidelidad Pantalla de Compra

La figura 25 proyecta un diseño definido en el módulo de ventas del proyecto con las opciones de agregar y remover productos de la venta a realizar, además cuenta con un campo no editable del total a pagar.

**Ventas**

**Seleccione un producto**

**Cantidad**

**Precio Unitario**

**Remover de compra** **Agregar a venta**

**Total a pagar**

Producto	Cantidad	Precio	Total

**Regresar**

**Finalizar Venta**

**Figura 25.** Diseño de media fidelidad Pantalla de Venta

La figura 26 proyecta un diseño definido en el módulo de corte de caja del proyecto con un registro de los productos vendidos en el día, cuenta con un pequeño formulario donde se ingresará el monto inicial del día y el dinero al final del día de ventas para posteriormente ser calculado una diferencia y obtener las ganancias.

**Corte de caja**

**Monto inicial**  **Diferencia**

**Monto final**  **Calcular corte**

**Productos vendidos en el día**

**Figura 26.** Diseño de media fidelidad Pantalla de corte de caja

La figura 27 proyecta un diseño definido en el módulo de reportar ventas del proyecto, en el espacio en blanco se mostrará el reporte en formato con las fechas indicadas por el usuario.

**Reportar ventas**

**Fecha Inicial**  **Imprimir Reporte**

**Fecha Final**

**Figura 27.** Diseño de media fidelidad Pantalla de reporte de ventas

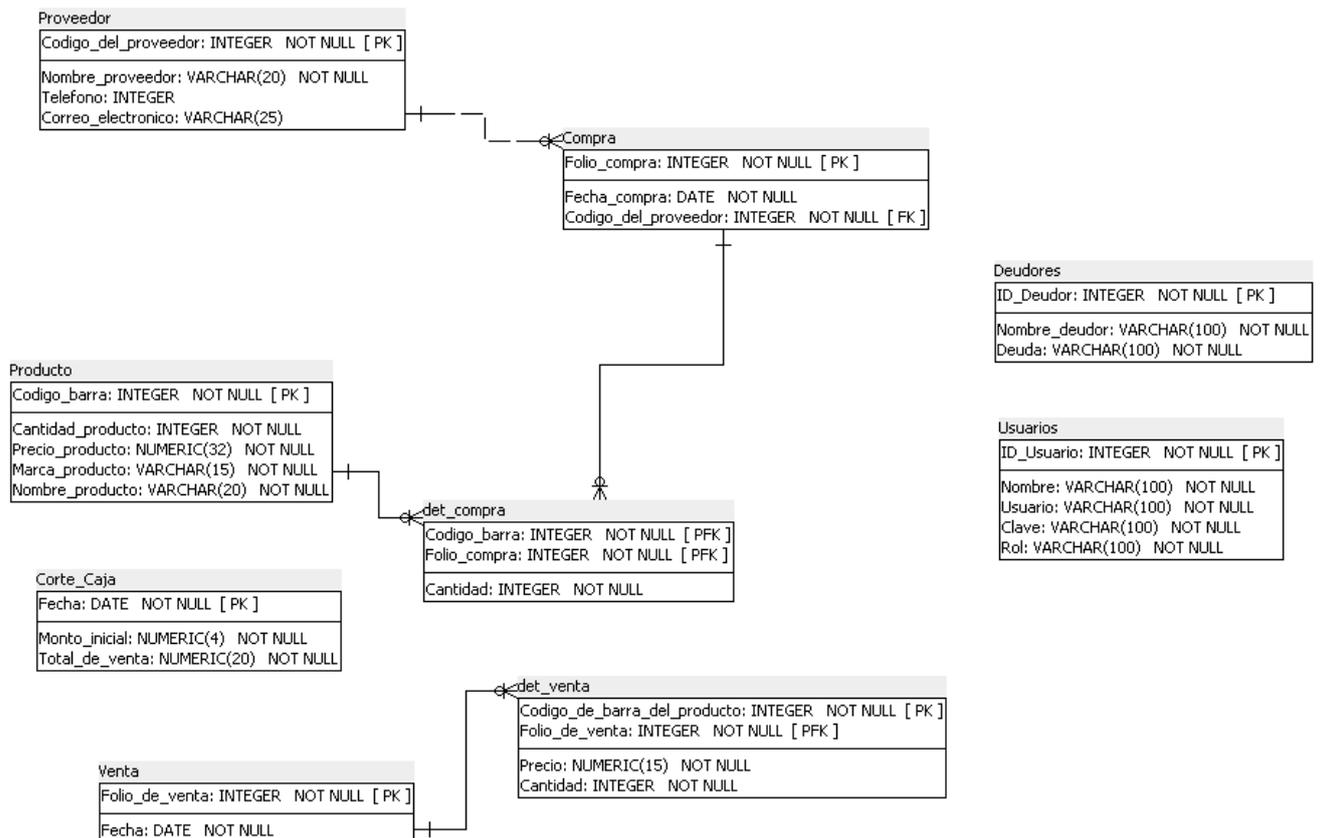
En la figura 28 muestra un diseño detallado del módulo de inventario listando las características de los productos en existencia.

<b>Inventario</b>				
<b>Nombre</b>	<b>Marca</b>	<b>Cantidad</b>	<b>Precio</b>	<b>Aviso</b>

**Figura 28.** Diseño de media fidelidad Pantalla de inventario

### **Diseño de la base de datos**

En la figura 29 se presenta el diseño para la base de datos a implementar en el desarrollo del proyecto.



**Figura 29.** Diseño de base datos

En la sección de apéndices se encuentra el script para generar la base de datos.

## Diccionario de datos

### Lista de mesas

- Compra
- Corte\_caja
- det\_compra
- det\_venta
- Producto
- Proveedor
- Deudor
- Venta

### Compra

(Nombre físico: Compra)

Se enuncian los elementos que contienen una compra de mercancía

Nombre de columna lógica	Nombre de columna física	Tipo	PK	NULO
Folio_compra	Folio_compra	ENTERO	PK	NO
Es el identificador de una compra				
Fecha_compra	Fecha_compra	FECHA		NO
Guarda la fecha de la compra				
Código_del_proveedor (FK)	Código_del_proveedor	ENTERO		NO
Es el identificador del proveedor				

## Referencias

- Proveedor a través de (Código\_del\_proveedor)

## Referenciado por

- det\_compra haciendo referencia (Folio\_compra)

## Corte\_caja

(Nombre físico: Corte\_caja)

Almacena el monto vendido al final del día

Nombre de columna lógica	Nombre de columna física	Tipo	PK	NULO
Fecha (PK)	Fecha	FECHA	PK	NO
Guarda la fecha del corte de caja				
Monto_inicial	Monto_inicial	NUMÉRICO (4,0)		NO
Guarda el monto al inicio del día				
Total_de_venta	Total_de_venta	NUMÉRICO (20)		NO
Almacena el total de las ventas				

## det\_compra

(Nombre físico: det\_compra)

Aquí se muestra los detalles al realizar una compra

Nombre de columna lógica	Nombre de columna física	Tipo	PK	NULO
Código_barra (PK) (FK)	Código_barra	ENTERO	PK	NO
Número que identifica al producto				
Folio_compra (PK) (FK)	Folio_compra	ENTERO	PK	NO
Es el identificador de una compra				
Cantidad	Cantidad	ENTERO		NO
La cantidad del producto				

### Referencias

- Producto a través de (Código\_barra)
- Compra a través de (Folio\_compra)

### det\_venta

(Nombre físico: det\_venta)

Aquí se muestra los detalles al realizar en venta

Nombre de columna lógica	Nombre de columna física	Tipo	PK	NULO
Código_de_barra_del_producto (PK)	Código_de_barra_del_producto (PK)	ENTERO	PK	NO
Es el identificador del producto				
Folio_de_venta (PK) (FK)	Folio_de_venta	ENTERO	PK	NO
Es el identificador de una venta				
Precio	Precio	NUMÉRICO (15,0)		NO
El precio de cada artículo				
Cantidad	Cantidad	ENTERO		NO
Es el número o cantidad del producto				

### Referencias

- Venta a través (Folio\_de\_venta)

## Producto

(Nombre físico: Producto)

Se mencionan las características del producto

Nombre de columna lógica	Nombre de columna física	Tipo	PK	NULO
Código_barra (PK)	Código_barra	ENTERO	PK	NO
Número que identifica al producto				
Cantidad_producto	Cantidad_producto	ENTERO		NO
El número de piezas del producto				
Precio_producto	Precio_producto	NUMÉRICO (32,0)		NO
Guarda el precio del producto				
Marca_producto	Marca_producto	VARCHAR (15)		NO
Es la marca del producto				
Nombre_producto	Nombre_producto	VARCHAR (20)		NO
Guarda el nombre del producto				

### Referenciado por

- referencia (Código\_barra)

## Proveedor

(Nombre físico: Proveedor)

Contiene la información del proveedor

Nombre de columna lógica	Nombre de columna física	Tipo	PK	NULO
Código_del_proveedor (PK)	Código_del_proveedor	ENTERO	PK	NO
Es el identificador del proveedor				

Nombre_proveedor	Nombre_proveedor	VARCHAR (20)	NO
------------------	------------------	--------------	----

Guarda el nombre del proveedor

Teléfono	Teléfono	ENTERO	SI
----------	----------	--------	----

Guarda el teléfono del proveedor

Correo_electrónico	Correo_electrónico	VARCHAR (25)	SI
--------------------	--------------------	--------------	----

Guarda el correo electrónico del proveedor

### Referenciado por

- Compra referencia (Código\_del\_proveedor)

### Deudor

(Nombre físico: Deudor)

Contiene la información del deudor

Nombre de columna lógica	Nombre de columna física	Tipo	PK	NULO
--------------------------	--------------------------	------	----	------

ID_deudor (PK)	ID_deudor	ENTERO	PK	NO
----------------	-----------	--------	----	----

Es el identificador del deudor

Nombre_deudor	Nombre_deudor	VARCHAR (100)		NO
---------------	---------------	---------------	--	----

Guarda el nombre del deudor

Deuda	Deuda	NUMÉRICO (32,0)		SI
-------	-------	-----------------	--	----

Guarda el monto del deudor

### Referenciado por

- Compra referencia (ID\_deudor)

### Venta

(Nombre físico: Venta)

### Características de una venta

Nombre de columna lógica	Nombre de columna física	Tipo	PK	NULO
Folio_de_venta (PK)	Folio_de_venta	ENTERO	PK	NO
Es el identificador de una venta				
Fecha	Fecha	FECHA		NO
Guarda la fecha de la venta				

### Referenciado por

- det\_venta referenciando (Folio\_de\_venta)

### 3.3. Codificación (construcción del prototipo)

La figura 30 presenta la ventana funcional de la pantalla principal del punto de venta listando la mayoría de los módulos desarrollados, cuenta con un botón de salir al no existir la clásica “x” que cierra los programas.



**Figura 30.** Primera versión: Pantalla principal del sistema

La figura 31 presenta la ventana funcional de catálogo de productos, cuenta con un pequeño formulario de modificación del producto al dar clic sobre los productos registrados, en la sección en blanco es donde se listarán los productos registrados en la Base de datos (BD), al seleccionar algún registro este podrá ser borrado con el botón eliminar.

**Datos del producto**

Código de barras  Marca del producto

Nombre del producto  Precio del producto

Cantidad del producto

Código	Cantidad	Precio	Marca	Nombre	Proveedor

**Figura 31.** Primera versión: Pantalla datos del producto

La figura 32 presenta la ventana funcional de la pantalla registrar nuevos productos, se tiene un formulario donde se deberán ingresar los datos relacionados al nuevo producto como su nombre, precio, marca y el proveedor al que se compró.

**Registrar nuevo producto**

Código de barra  Se asigna automáticamente

Nombre del producto

Cantidad del producto  La cantidad se asigna en compras

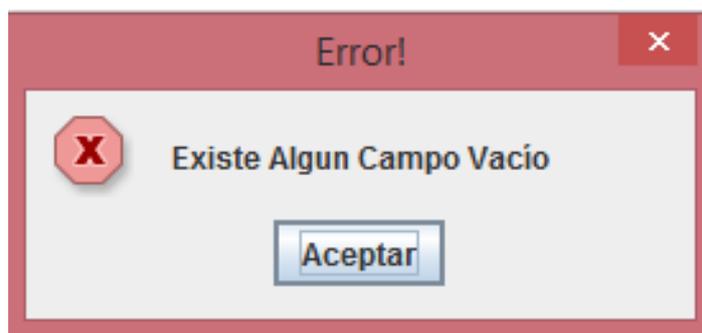
Precio del producto

Marca del producto

Proveedor del producto  ▼

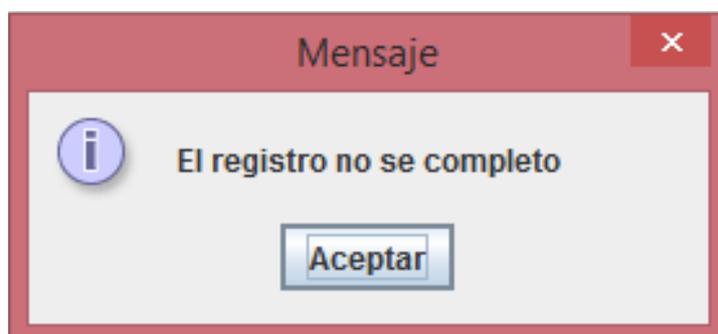
**Figura 32.** Primera versión: Pantalla registro de producto

La figura 33 presenta la ventana de mensaje de error al dejar campos vacíos en los formularios que existen en el sistema.



**Figura 33.** Primera versión: Mensaje de error algún campo vacío

La figura 34 presenta la ventana de mensaje de advertencia al no completarse el proceso de envío y registro de datos en la base de datos, es decir, si existió algún problema.



**Figura 34.** Primera versión: Mensaje de error al no completarse un registro

La figura 35 presenta la ventana de mensaje de éxito al completarse el proceso de envío y registro de datos en la base de datos de un nuevo registro ya se producto o proveedor.



**Figura 35.** Primera versión: Mensaje de registro completado

La figura 36 presenta la ventana funcional de catálogo de proveedor, cuenta con un pequeño formulario de modificación a la información del proveedor al dar clic sobre los registros, en la sección en blanco es donde se listarán los

proveedores registrados en la BD y su respectiva información, al seleccionar algún registro este podrá ser borrado con el botón eliminar.

Datos del proveedor

Código del proveedor  Teléfono

Nombre del proveedor  Correo electrónico

Código	Nombre	Teléfono	Correo
--------	--------	----------	--------

**Figura 36.** Primera versión: Datos del proveedor

La figura 37 presenta la ventana funcional de la pantalla registrar nuevos proveedores, se tiene un formulario donde se deberán ingresar los datos relacionados al nuevo proveedor como su nombre, teléfono y el correo del proveedor al que se compró.

Registrar nuevo proveedor

Código del proveedor Se asigna automáticamente

Nombre del proveedor

Teléfono

Correo electrónico

**Figura 37.** Primera versión: Pantalla registro de proveedor

La figura 38 presenta la ventana funcional de módulo de compras, se divide en dos secciones la del lado izquierdo es un formulario que permite seleccionar los productos existentes e ingresar la cantidad a comprar, además cuenta con un campo para ingresar el precio de los productos a comprar, del lado derecho se irán listando la información de los productos a comprar.

**Compras**

Seleccionar el producto:

Seleccionar proveedor:

Precio del producto:

Cantidad del producto:

Marca:

Total a pagar:

Producto	Proveedor	Precio	Cantidad	Marca

**Figura 38.** Primera versión: Pantalla de compras

La figura 39 presenta la ventana funcional de módulo de ventas, se divide en dos secciones la del lado izquierdo es un formulario que permite seleccionar los productos existentes con algún ejemplar disponible e ingresar la cantidad a vender, además cuenta con un campo para mostrar el precio a pagar de los productos a vender, del lado derecho se irán listando la información de los productos a vender como su nombre, precio por unidad y su total de acuerdo al precio y la cantidad vendida.

**Ventas**

Seleccione producto:

Cantidad del producto:

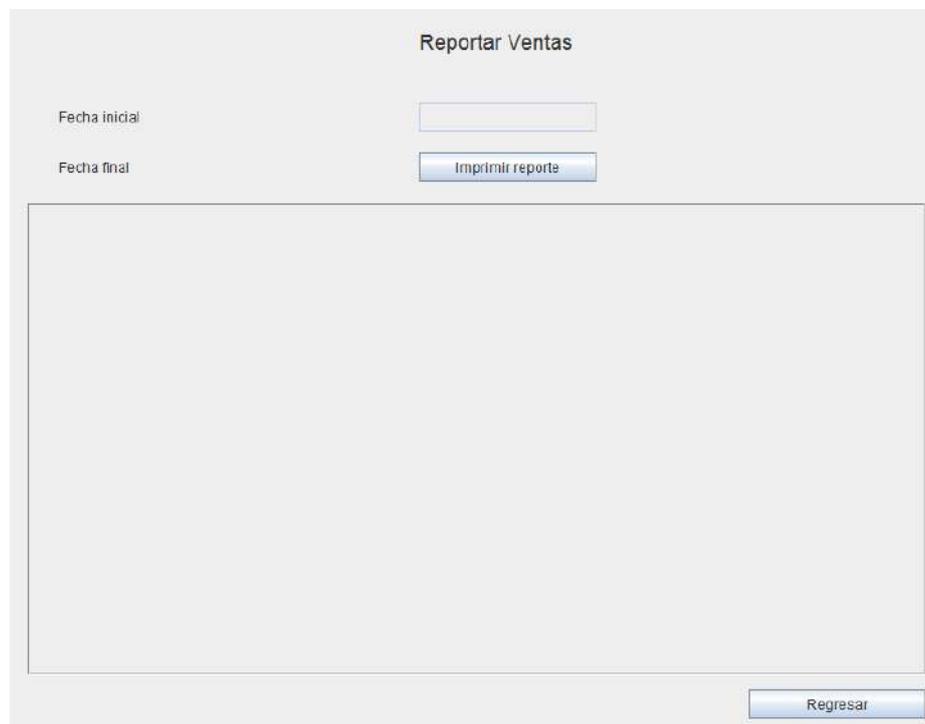
Precio por unidad:

Total a pagar:

--	--	--	--	--

**Figura 39.** Primera versión: Pantalla de ventas

La figura 40 presenta la ventana funcional de módulo de reportar, en el espacio en blanco se mostrará el informe de ventas de acuerdo a las fechas escritas en el formulario de arriba del espacio en blanco.



**Figura 40.** Primera versión: Pantalla de reportes de venta

La figura 41 presenta la ventana funcional de módulo de corte de caja, se ingresará el monto inicial y final del día, posteriormente el botón calcular ventas se encargará de realizar las operaciones correspondientes, en la parte en blanco se mostrará una lista de los productos vendidos en el día así como sus cantidades y totales.

**Corte de caja**

Monto inicial  Diferencia

Monto final del día

**Figura 41.** Primera versión: Pantalla de corte de caja

La figura 42 presenta la ventana funcional de módulo de inventario con los datos de los productos en existencia.

**Inventario**

Código	Cantidad	Precio	Marca	Nombre
13	0	1.5	Bincil	Lapiz

**Figura 42.** Primera versión: Pantalla de Inventario

### 3.4. Pruebas

En la tabla 9 se muestra un caso de prueba de las funciones del módulo de productos como registrar, editar y eliminar la información de los productos.

<b>Nombre del proyecto: Desarrollo de un punto de venta para la gestión de una papelería</b>		<b>ID Caso de prueba: CP-001</b>	
<b>Ambiente de prueba: Aplicación web</b>		ID Historia de Usuario: HU-001	
<b>Autor Caso de Prueba: José Alfredo Tlahuechtl Rodríguez</b>			
<b>Propósito</b>			
Verificar registro, modificar y eliminar productos			
#	Acciones	Salida Esperada	Salida obtenida
01	Registrar datos del producto	Mostrar lista de los productos registrados	Muestra datos de los productos registrados
02	Modificar datos del producto	Mostrar lista de los productos actualizados	Muestra lista de los productos actualizados.
03	Eliminar producto	Mostrar una alerta de eliminar producto	Muestra la alerta de eliminar producto
<b>Resultado: Aprobado</b>			

**Tabla 9.** Caso de prueba CP-001

En la tabla 10 se muestra un caso de prueba de las funciones del módulo de proveedor como registrar, editar y eliminar la información de los proveedores.

<b>Nombre del proyecto: Desarrollo de un punto de venta para la gestión de una papelería</b>		<b>ID Caso de prueba: CP-002</b>	
<b>Ambiente de prueba: Aplicación web</b>		ID Historia de Usuario: HU-002	
<b>Autor Caso de Prueba: José Alfredo Tlahuechtl Rodríguez</b>			
<b>Propósito</b>			
Verificar registro, modificar y eliminar proveedor			
#	Acciones	Salida Esperada	Salida obtenida
01	Registrar datos del proveedor	Mostrar lista de los proveedores registrados	Muestra datos de los proveedores registrados
02	Modificar datos del proveedor	Mostrar lista de los proveedores actualizados	Muestra lista de los proveedores actualizados.
03	Eliminar proveedor	Mostrar una alerta de eliminar proveedor	Muestra la alerta de eliminar proveedor
<b>Resultado: Aprobado</b>			

**Tabla 10.** Caso de prueba CP-002

En la tabla 11 se muestra un caso de prueba de las funciones del módulo de deudor como registrar, editar y eliminar la información de los deudores.

<b>Nombre del proyecto: Desarrollo de un punto de venta para la gestión de una papelería</b>		<b>ID Caso de prueba: CP-003</b>	
<b>Ambiente de prueba: Aplicación web</b>		ID Historia de Usuario: HU-003	
<b>Autor Caso de Prueba:</b> José Alfredo Tlahuechtl Rodríguez			
<b>Propósito</b>			
Verificar registro, modificar y eliminar deudor			
#	Acciones	Salida Esperada	Salida obtenida
01	Registrar datos del deudor	Mostrar lista de los deudores registrados	Muestra datos de los deudores registrados
02	Modificar datos del deudor	Mostrar lista de los deudores actualizados	Muestra lista de los deudores actualizados.
03	Eliminar deudor	Mostrar una alerta de eliminar deudor	Muestra la alerta de eliminar deudor
<b>Resultado:</b> Aprobado			

**Tabla 11.** Caso de prueba CP-003

En la tabla 12 se muestra un caso de prueba de las funciones del módulo de compra las cuales son realizar la compra y actualización de los artículos en el inventario.

<b>Nombre del proyecto: Desarrollo de un punto de venta para la gestión de una papelería</b>		<b>ID Caso de prueba: CP-004</b>	
<b>Ambiente de prueba: Aplicación web</b>		ID Historia de Usuario: HU-004	
<b>Autor Caso de Prueba:</b> José Alfredo Tlahuechtl Rodríguez			
<b>Propósito</b>			
Realizar una compra y actualización de cantidad del producto			
#	Acciones	Salida Esperada	Salida obtenida
01	Seleccionar un proveedor	Listar los proveedores registrados	Lista los proveedores registrados
02	Teclar datos de la compra	Agregar a compra	Agrega a compra
04	Mostrar total a pagar	Mostrar total a pagar	Muestra total a pagar
03	Alerta de compra realizada	Mostrar una alerta de compra exitosa	Muestra la alerta de compra exitosa
<b>Resultado:</b> Aprobado			

**Tabla 12.** Caso de prueba CP-004

En la tabla 13 se muestra un caso de prueba de las funciones del módulo de ventas las cuales son realizar la venta y actualización de los artículos en el inventario.

<b>Nombre del proyecto: Desarrollo de un punto de venta para la gestión de una papelería</b>		<b>ID Caso de prueba: CP-005</b>	
<b>Ambiente de prueba: Aplicación web</b>		ID Historia de Usuario: HU-005	
<b>Autor Caso de Prueba: José Alfredo Tlahuechtl Rodríguez</b>			
<b>Propósito</b>			
Realizar una venta y actualización de cantidad del producto			
#	Acciones	Salida Esperada	Salida obtenida
01	Teclear datos de la venta	Agregar a venta	Agrega a venta
02	Mostrar total a cobrar	Mostrar total a cobrar	Muestra el total a cobrar
03	Alerta de venta realizada	Mostrar una alerta de venta exitosa	Muestra la alerta de venta exitosa
<b>Resultado: Aprobado</b>			

**Tabla 13.** Caso de prueba CP-005

En la tabla 14 se muestra un caso de prueba del módulo de inventario para mostrar en pantalla datos de los productos.

<b>Nombre del proyecto: Desarrollo de un punto de venta para la gestión de una papelería</b>		<b>ID Caso de prueba: CP-006</b>	
<b>Ambiente de prueba: Aplicación web</b>		ID Historia de Usuario: HU-006	
<b>Autor Caso de Prueba: José Alfredo Tlahuechtl Rodríguez</b>			
<b>Propósito</b>			
Verificar los productos en existencia			
#	Acciones	Salida Esperada	Salida obtenida
01	Listar los productos	Mostrar datos de productos en una tabla	Muestra datos de productos en una tabla
<b>Resultado: Aprobado</b>			

**Tabla 14.** Caso de prueba CP-006

En la tabla 15 se muestra un caso de prueba de las funciones del módulo de corte de caja para calcular la diferencia entre ingresos y egresos.

<b>Nombre del proyecto: Desarrollo de un punto de venta para la gestión de una papelería</b>		<b>ID Caso de prueba: CP-007</b>	
<b>Ambiente de prueba: Aplicación web</b>		ID Historia de Usuario: HU-007	
<b>Autor Caso de Prueba: José Alfredo Tlahuechtl Rodríguez</b>			
<b>Propósito</b>			
Verificar ventas y compras realizadas en el día, mostrar la diferencia entre egresos e ingresos			
#	Acciones	Salida Esperada	Salida obtenida
01	Mostrar ventas del día	Mostrar detalles de venta en el día	Muestra detalles de venta en el día
02	Mostrar ingresos	Mostrar ingresos	Muestra ingresos
03	Mostrar compras del día	Mostrar detalles de compra en el día	Muestra detalles de compra en el día
04	Mostrar egresos	Mostrar egresos	Muestra egresos
05	Mostrar diferencia de egresos e ingresos	Mostrar detalles del corte	Muestra detalles del corte
<b>Resultado: Aprobado</b>			

**Tabla 15.** Caso de prueba CP-007

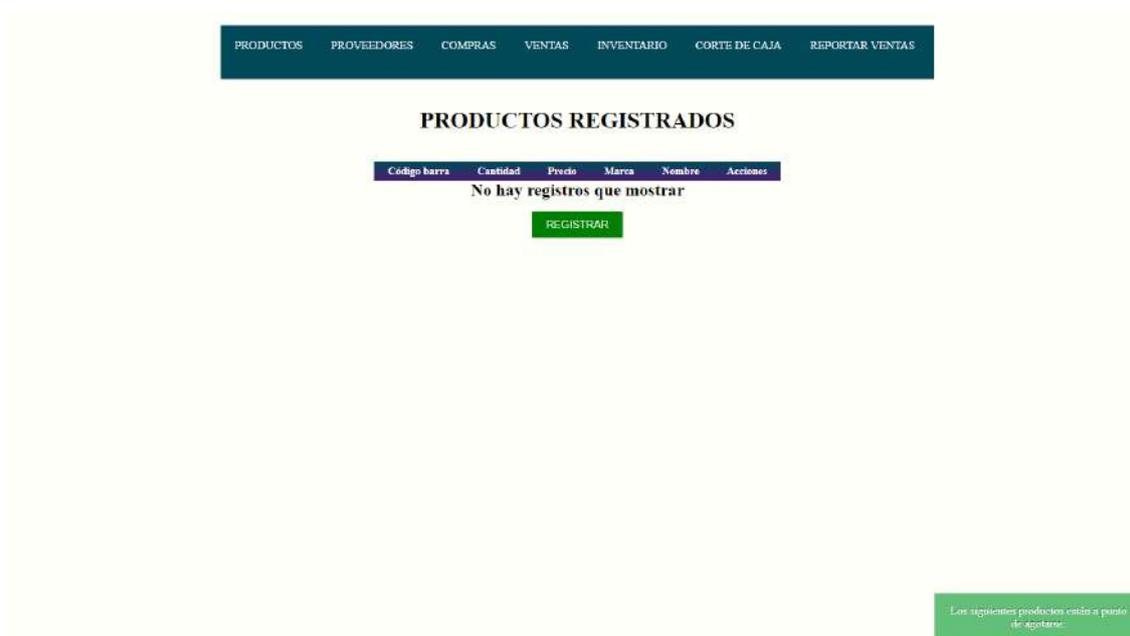
En la tabla 16 se muestra un caso de prueba de las funciones del módulo de reporte de ventas.

<b>Nombre del proyecto: Desarrollo de un punto de venta para la gestión de una papelería</b>		<b>ID Caso de prueba: CP-008</b>	
<b>Ambiente de prueba: Aplicación web</b>		ID Historia de Usuario: HU-008	
<b>Autor Caso de Prueba: José Alfredo Tlahuechtl Rodríguez</b>			
<b>Propósito</b>			
Comprobar reportes de ventas			
#	Acciones	Salida Esperada	Salida obtenida
01	Seleccionar fecha	Mostrar las ventas realizadas en la fecha	Muestra las ventas realizadas en la fecha
02	Mostrar ganancias del día	Mostrar ganancias	Muestra los ingresos del día
<b>Resultado: Aprobado</b>			

**Tabla 16.** Caso de prueba CP-008

### 3.5. Mantenimiento (retroalimentación)

La figura 43 muestra la ventana funcional del catálogo de productos, en esta ventana se mostrará la información de los productos registrados en la BD, en la parte de arriba de la tabla “Productos registrados” se encuentra un menú el cual se mostrará en todas las ventanas y servirá para navegar entre los diferentes módulos que contiene este sistema



**Figura 43.** Prototipo mejorado: Pantalla de productos

La figura 44 presenta un mensaje emergente del prototipo, el cual advierte sobre los productos próximos a agotarse, siempre y cuando existan menos de 15 unidades. Se presentará el mensaje en los módulos de inventario, productos y ventas



**Figura 44.** Prototipo mejorado: Alerta de los productos a agotarse

La figura 45 presenta la ventana funcional de un formulario de registro de productos, en el que se deberá agregar la información pertinente al producto como nombre, marca, precio, cantidad.

PRODUCTOS    PROVEEDORES    COMPRAS    VENTAS    INVENTARIO    CORTE DE CAJA    REPORTAR VENTAS

### FORMULARIO DE REGISTRO

Nombre

Marca

Precio

Cantidad

**Figura 45.** Prototipo mejorado: Formulario de registro para productos

La figura 46 presenta un mensaje emergente cuando los campos de algún formulario no han sido completados.



**Figura 46.** Prototipo mejorado: Alerta por si algún campo está vacío

La figura 47 muestra la ventana funcional del catálogo de proveedor, en esta ventana se mostrará la información de los proveedores como código, nombre, correo y una columna de acciones que permitirá modificar o eliminar el registro del proveedor.

PRODUCTOS    PROVEEDORES    COMPRAS    VENTAS    INVENTARIO    CORTE DE CAJA    REPORTAR VENTAS

### PROVEEDORES REGISTRADOS

Código de Proveedor	Nombre	Teléfono	Correo Electrónico	Acciones
No hay registros que mostrar				

**Figura 47.** Prototipo mejorado: Pantalla de proveedores

La figura 48 presenta la ventana funcional de un formulario de registro de proveedores, en el que se deberá agregar la información pertinente al proveedor como nombre, teléfono y correo.

PRODUCTOS    PROVEEDORES    COMPRAS    VENTAS    INVENTARIO    CORTE DE CAJA    REPORTAR VENTAS

### FORMULARIO DE REGISTRO

Nombre

Teléfono

Correo electrónico

**REGISTRAR**    **CANCELAR**

**Figura 48.** Prototipo mejorado: Formulario de registro para proveedores

La figura 49 muestra la ventana funcional del módulo de compras, dividida en dos secciones la del lado izquierdo permitirá seleccionar al proveedor que se le compran los productos, luego se ingresará la información del producto una vez finalizado en la sección a la derecha se agregaran los artículos a comprar calculando automáticamente el total de la compra.

PRODUCTOS    PROVEEDORES    COMPRAS    VENTAS    INVENTARIO    CORTE DE CAJA    REPORTAR VENTAS

### COMPRAS

Selección proveedor

Nombre del producto

Marca

Cantidad

Precio de compra

**FINALIZAR COMPRA**

NOMBRE	MARCA	CANTIDAD	PRECIO	REMOVED
Total de compra: \$0				

**AGREGAR A COMPRA**    **VACIAR COMPRA**

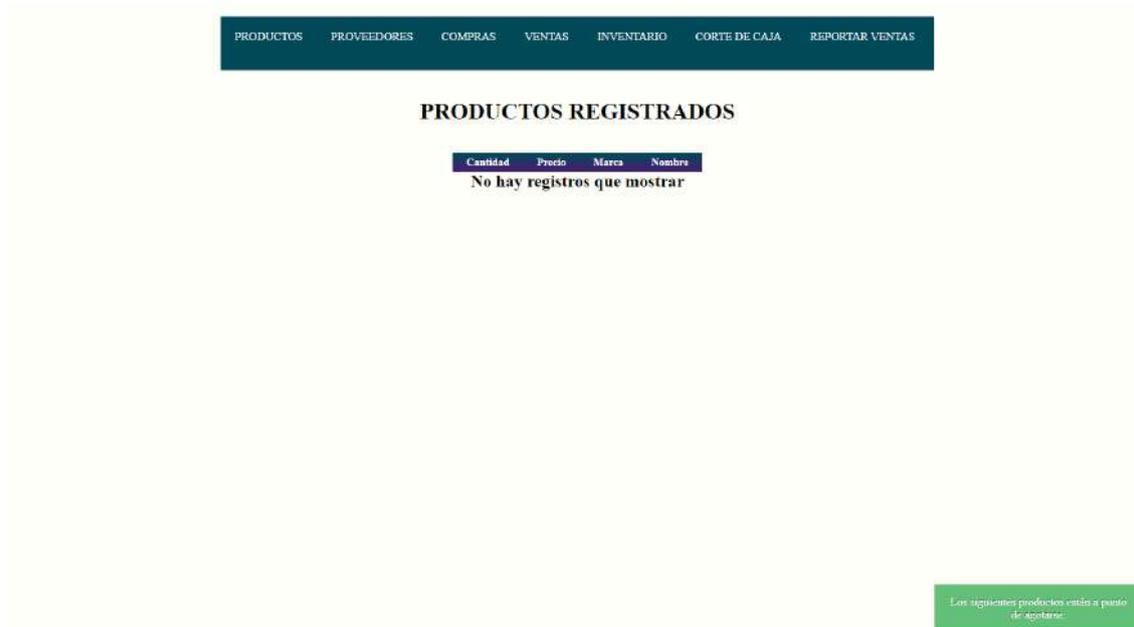
**Figura 49.** Prototipo mejorado: Pantalla de compras

La figura 50 muestra la ventana funcional del módulo de ventas, dividida en dos secciones la del lado izquierdo permitirá seleccionar al producto a vender, posteriormente se ingresará la cantidad del producto, la sección del precio unitario no será editable, una vez finalizado en la sección a la derecha se agregará la información de los artículos a vender calculando automáticamente el total de la venta.



**Figura 50.** Prototipo mejorado: Pantalla de ventas

La figura 51 muestra la ventana funcional del módulo de inventario donde se mostrará la información de todos los productos registrados.



**Figura 51.** Prototipo mejorado: Pantalla de inventario

La figura 52 muestra la ventana funcional del módulo de corte de caja, en la cual se mostrarán los productos vendidos en el día y compras realizadas en la misma fecha, además de calcular la diferencia entre egresos e ingresos.



**Figura 52.** Prototipo mejorado: Pantalla de corte de caja

La figura 53 muestra la ventana funcional del módulo de reportes de venta, en la cual se listarán los productos vendidos en la fecha especificada por el usuario.



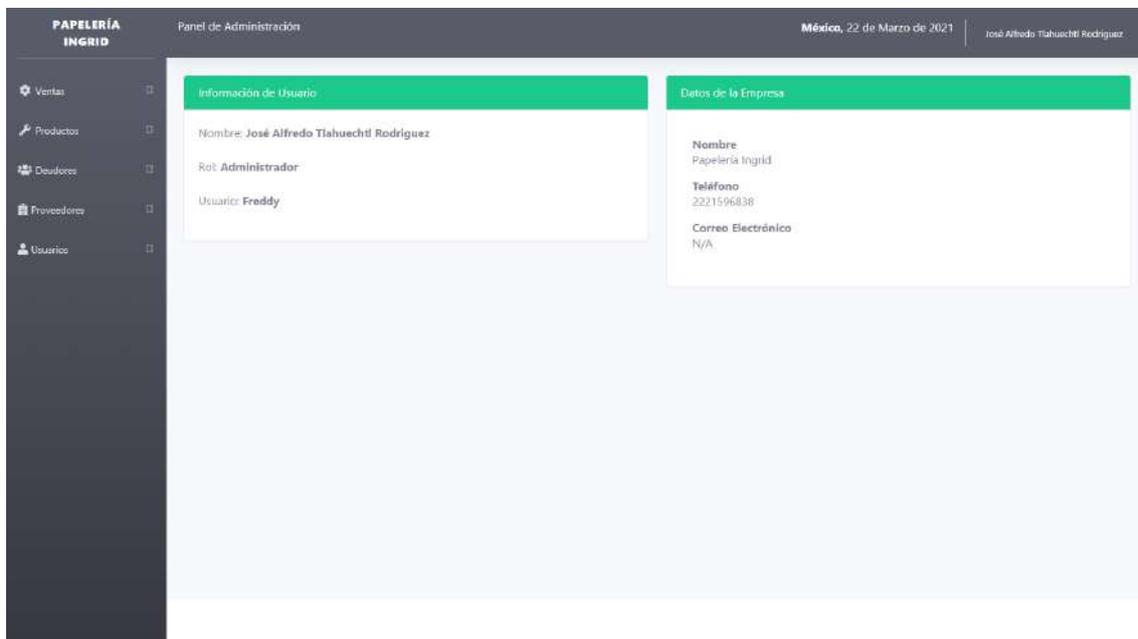
**Figura 53.** Prototipo mejorado: Pantalla de reportes de ventas

En la figura 54 Se muestra un inicio de sesión, en el cual se ingresa el nombre de usuario y contraseña del mismo



**Figura 54.** Pantalla de inicio de sección

En la figura 55 Se muestra la pantalla principal del sistema, del lado izquierdo se ubicarán los módulos existentes en el sistema, mientras que en el espacio en blanco se mostrará la información de usuario y datos de la empresa. En la parte superior se visualizará la fecha del día y nombre real de la persona que ha iniciado sesión.



**Figura 55.** Prototipo mejorado: Pantalla principal

En la figura 56 Se podrán realizar las ventas, cuenta con una lista desplegable que permite seleccionar el producto a agregar a la venta donde se mostrará la información, los mismos productos agregados podrán ser retirados con la opción de eliminar, o si bien la venta se realizara el botón generar venta mandará un ticket de la venta.

Datos Venta

VENDEDOR: JOSÉ ALFREDO TLAHUECHTL RODRIGUEZ

Acciones  
Anular Generar Venta

Productos	Existencia	Cantidad	Precio	Precio Total	Acciones
<input type="text"/>	-	<input type="text" value="0"/>	0.00	0.00	

Código	Descripción	Cantidad	Precio	Precio Total	Acciones
7	Gaseosa	1	1500.00	1500	<span>Eliminar</span>
1	lenovo	8	10.00	80	<span>Eliminar</span>
Total \$/:				1580	

**Figura 56.** Pantalla de ventas

En la figura 57 Se visualizarán las ventas realizadas en las diferentes fechas y con hora exacta además de contar con la opción de generar el reporte o ticket de esa venta.

Reportes de ventas

Mostrar  registros

Buscar:

id	Ti	Fecha	Ti	Total	Ti	Acciones
1		2021-03-21 23:37:14		3000.00		<span>Generar reporte</span>

Mostrando 1 a 1 de 1 registros

Anterior 1 Siguiente

**Figura 57.** Pantalla de reporte de ventas

En la figura 58 Se visualiza los datos de la papelería Ingrid así como los productos vendidos, sus cantidades y total a pagar



**Figura 58.** Pantalla de ticket

En la figura 59 Se muestra el formulario de registro de los productos con los campos necesarios de la información a ingresar del producto.

**Registro productos**

Proveedor  
Lineo

Producto  
Ingrese nombre del producto

Precio  
Ingrese precio

Cantidad  
Ingrese cantidad

**Guardar Producto** **Regresar**

**Figura 59.** Pantalla de registro de producto

En la figura 60 Se muestra el formulario para modificar datos de los productos

**Figura 60.** Pantalla de modificar producto

En la figura 61 Se muestran los registros de los productos en existencia y otro tipo de información de los mismos

ID	PRODUCTO	PRECIO	EXISTENCIA	ACCIONES
1	Lápiz	10.00	48	[Refresh] [Edit] [Delete]
2	Sacapuntas	2.00	69	[Refresh] [Edit] [Delete]
7	Cartulina Azul	1.50	1	[Refresh] [Edit] [Delete]

**Figura 61.** Pantalla de productos

En la figura 62 Se muestra el formulario de registro para deudores.

Registro deudor

Nombre

Teléfono

Deuda

Guardar información
Regresar

**Figura 62.** Pantalla de registro deudor

En la figura 63 Se visualizarán los clientes que tienen alguna deuda con la papelería

Nuevo

Mostrar 10 registros Buscar:

ID	NOMBRE	TELEFONO	DEUDA	ACCIONES
3	Víctor	16555	254.7	<span style="color: #27ae60; font-size: 1.2em;">✎</span> <span style="color: #e74c3c; font-size: 1.2em;">✖</span>
4	Alfredo	2147483647	14	<span style="color: #27ae60; font-size: 1.2em;">✎</span> <span style="color: #e74c3c; font-size: 1.2em;">✖</span>
5	House	12345	34.46	<span style="color: #27ae60; font-size: 1.2em;">✎</span> <span style="color: #e74c3c; font-size: 1.2em;">✖</span>

Mostrando 1 a 3 de 3 registros Anterior 1 Siguiente

**Figura 63.** Pantalla de deudores

En la figura 64 Se muestra un mensaje de confirmación



**Figura 64.** Pantalla de confirmación

En la figura 65 se muestra un formulario para el registro de nuevos proveedores.

Registro de Proveedor

NOMBRE

Ingrese nombre

TELÉFONO

Ingrese teléfono

DIRECCIÓN

Ingrese Direccion

Guardar Proveedor Regresar

**Figura 65.** Pantalla de registro de proveedor

En la figura 66 Se visualiza una lista de los proveedores y su información de contacto

Proveedores

Mostrar 10 registros

Buscar:

Nuevo

ID	PROVEEDOR	TELEFONO	DIRECCION	ACCIONES
1	Open	9	LJ	Editar
2	Lineo	9865412	Lima	Editar
7	Rosa	1234567890	Montalvo	Editar

Mostrando 1 a 3 de 3 registros

Anterior 1 Siguiente

**Figura 66.** Pantalla de proveedores

En la figura 67 Se muestra el formulario de modificación de información del proveedor

Proveedor Actualizado correctamente

Proveedor

Teléfono

Dirección

[Editar Proveedor](#) [Regresar](#)

**Figura 67.** Pantalla de actualización de información del proveedor

En la figura 68 Se visualiza un formulario de registro de usuarios con la información necesaria a ser completada para darse de alta.

Registro de usuarios

Nombre

Usuario

Contraseña

Rol

[Registrar Usuario](#) [Regresar](#)

**Figura 68.** Pantalla de registro de usuarios

En la figura 69 se listan los usuarios registrados y activos que laboren en la papelería

Usuarios Nuevo

Mostrar 10 registros Buscar:

ID	NOMBRE	USUARIO	ROL	ACCIONES
9	José Alfredo Tahuecht Rodríguez	Freddy	1	<a href="#">Editar</a> <a href="#">Eliminar</a>
11	House	House	2	<a href="#">Editar</a> <a href="#">Eliminar</a>

Mostrando 1 a 2 de 2 registros Anterior **1** Siguiente

**Figura 69.** Pantalla de usuarios

En la figura 70 se muestra el formulario de actualización de datos de los usuarios.

Usuario Actualizado

Nombre

Usuario

Rol

[Editar Usuario](#) [Regresar](#)

**Figura 70.** Pantalla de actualización de usuario

En la sección de apéndices se encontrará el código fuente del desarrollo del sistema.

## 4. Conclusiones y recomendaciones

---

Realizar el proyecto de “Desarrollo de un punto de venta para la gestión de una papelería” ha sido una experiencia desafiante y productiva debido a la experiencia insuficiente en el área de desarrollo web y elaboración de sistemas de punto de venta, adquiriendo conocimiento nuevo en cada fase para mejorar mis habilidades en las herramientas implementadas.

Las recomendaciones para este proyecto son:

- Realizar más pruebas
- Generar un reporte .pdf del corte caja
- El menú debe de usar el 100% de ancho del dispositivo

# 5. Apéndices

---

## Apéndice A Script para generar BD

En la figura 71 se muestra parte del script necesario para la generación de la base de datos, que se usara en el punto de venta

```
CREATE TABLE Proveedor (  
    Codigo_del_proveedor INT NOT NULL,  
    Nombre_proveedor VARCHAR(35) NOT NULL,  
    Telefono VARCHAR NOT NULL,  
    Correo_electronico VARCHAR(35),  
    PRIMARY KEY (Codigo_del_proveedor)  
);  
  
CREATE TABLE Producto (  
    Codigo_barra INT NOT NULL,  
    Cantidad_producto INT NOT NULL,  
    Precio_producto DOUBLE PRECISION NOT NULL,  
    Marca_producto VARCHAR(25) NOT NULL,  
    Nombre_producto VARCHAR(20) NOT NULL,  
    PRIMARY KEY (Codigo_barra)  
);  
  
CREATE TABLE Compra (  
    Folio_compra INT NOT NULL,  
    Fecha_compra DATE NOT NULL,  
    Codigo_del_proveedor INT NOT NULL,  
    PRIMARY KEY (Folio_compra)  
);  
  
CREATE TABLE det_compra (  
    Codigo_barra INT NOT NULL,  
    Folio_compra INT NOT NULL,  
    Cantidad INT NOT NULL,  
    Precio DOUBLE PRECISION NOT NULL,  
    PRIMARY KEY (Codigo_barra, Folio_compra)  
);  
  
CREATE TABLE Venta (  
    Folio_de_venta INT NOT NULL,  
    Fecha DATE NOT NULL,  
    PRIMARY KEY (Folio_de_venta)  
);  
  
CREATE TABLE det_venta (  
    Codigo_de_barra_del_producto INT NOT NULL,  
    Folio_de_venta INT NOT NULL,  
    Precio DOUBLE PRECISION NOT NULL,  
    Cantidad INT NOT NULL,  
    PRIMARY KEY (Codigo_de_barra_del_producto, Folio_de_venta)  
);
```

Figura 71. Script para generar la base de datos (1/2)

En la figura 72 se muestra un segmento del script necesario para la generación de la base de datos, que se usara en el punto de venta

```
CREATE TABLE Corte_Caja (  
    Fecha DATE NOT NULL,  
    Monto_inicial DOUBLE PRECISION NOT NULL,  
    Total_de_venta DOUBLE PRECISION NOT NULL,  
    PRIMARY KEY (Fecha)  
);  
  
ALTER TABLE Compra ADD CONSTRAINT proveedor_compra_fk  
FOREIGN KEY (Codigo_del_proveedor)  
REFERENCES Proveedor (Codigo_del_proveedor)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;  
  
ALTER TABLE det_compra ADD CONSTRAINT producto_det_compra_fk  
FOREIGN KEY (Codigo_barra)  
REFERENCES Producto (Codigo_barra)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;  
  
ALTER TABLE det_compra ADD CONSTRAINT compra_det_compra_fk  
FOREIGN KEY (Folio_compra)  
REFERENCES Compra (Folio_compra)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;  
  
ALTER TABLE det_venta ADD CONSTRAINT venta_det_venta_fk  
FOREIGN KEY (Folio_de_venta)  
REFERENCES Venta (Folio_de_venta)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;
```

Figura 72. Script para generar la base de datos (2/2)

## Apéndice B Código

En la figura 73 se presenta el código generado para la interfaz del menú desarrollado en el lenguaje de marcas de hipertexto (HTML, HyperTextMarkup).

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>PUNTO DE VENTA</title>
  <link rel="stylesheet" type="text/css" href="estilos_principal.css">
  <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <link rel="stylesheet" type="text/css" href="alertifyjs/css/alertify.css">
  <link rel="stylesheet" type="text/css" href="alertifyjs/css/themes/default.css">
</head>
<body>
  <div class="contenedor">
    <header>
      <div class="menu_bar">
        <a href="#" class="bt-menu"><span></span>MENÚ</a>
      </div>
      <nav>
        <ul>
          <li><a href="productos.php">PRODUCTOS</a><span></span></li>
          <li><a href="proveedores.php">PROVEEDORES</a><span></span></li>
          <li><a href="Compras.php">COMPRAS</a><span></span></li>
          <li><a href="ventas.php">VENTAS</a><span></span></li>
          <li><a href="Inventario.php">INVENTARIO</a><span></span></li>
          <li><a href="corte_caja.php">CORTE DE CAJA</a><span></span></li>
          <li><a href="Reportar_ventas.php">REPORTAR VENTAS</a><span></span></li>
        </ul>
      </nav>
    </header>
  </div>
</body>
</html>
<script src="jquery-3.5.1.min.js"></script>
<script src="menu.js"></script>
<script src="confirmacion.js"></script>
<script src="alertifyjs/alertify.js"></script>
<script src="alertifyjs/alertify.min.js"></script>
```

Figura 73. Código fuente del menú

En la figura 74 se proyecta el código generado para la inserción de datos de un nuevo producto a la base de datos, desarrollado en el lenguaje de Hypertext Preprocessor (PHP)

```
<?php
include("conexion.php");
$nombre = "";
$marca = "";
$precio = "";
$cantidad = "";

if(isset($_POST['btn_registrar'])){
  //INSERTAR
  $nombre = $_POST['nombre'];
  $marca = $_POST['marca'];
  $precio = $_POST['precio'];
  $cantidad = $_POST['cantidad'];
  $nombre = mb_convert_case($nombre, MB_CASE_UPPER, "UTF-8");

  IF ($nombre=="" || $marca=="" || $precio=="" || $cantidad=="") {
    echo "<script>alertify.error('Todos los campos con obligatorios');</script>";
  } else {
    mysqli_query($conexion, "INSERT INTO $tabla_Producto (Codigo barra, Cantidad_producto, precio_producto, marca_producto, nombre_producto)
    values ('','$cantidad', '$precio', '$marca', '$nombre')");
    echo "<script>alertify.error('Todos los campos con obligatorios');</script>";
    header("Location: productos.php");
  }
}

if(isset($_POST['btn_cancelar'])){
  header("Location: productos.php");
}
?>
```

Figura 74. Código fuente para registrar nuevos productos.

En la figura 75 se proyecta el código generado para la inserción de datos de proveedores a la base de datos, desarrollado en el lenguaje de PHP.

```
<?php
include("conexion.php");
$nombre = "";
$telefono = "";
$correo = "";

if(isset($_POST['btn_registrar'])){
//INSERTAR
$nombre = $_POST['nombre'];
$telefono = $_POST['telefono'];
$correo = $_POST['correo'];
$nombre = mb_convert_case($nombre, MB_CASE_UPPER, "UTF-8");

if ($nombre=="" || $telefono=="" || $correo=="") {
    echo "<script>alert('todos los campos con obligatorios');</script>";
} else {
    mysqli_query($conexion, "INSERT INTO Stabla_Proveedor (Codigo_del_proveedor, Nombre_proveedor, Telefono, Correo_electronico) values ('','$nombre', '$telefono', '$correo')");
    echo "Registro completo";
    header("Location: proveedores.php");
}
}

if(isset($_POST['btn_cancelar'])){
    header("Location: proveedores.php");
}
?>
```

**Figura 75.** Código fuente para registrar nuevos proveedores

En la figura 76 se proyecta el código generado para el registro de datos del módulo de compras a la base de datos.

```

public function crearCompras(){
    include("conexion.php");
    $fecha=date('Y-m-d');
    $idCompra=self::creaFolio();
    $datos=$_SESSION['tablaComprasTemp'];
    $r=0;

    for ($i=0; $i < count($datos) ; $i++) {
        $d=explode("||", $datos[$i]);
        $sql="INSERT INTO compra (Folio_compra,
                                Fecha_compra,
                               Codigo_del_proveedor)
                                VALUES ('$idCompra',
                                '$fecha',
                                '$d[0]')";
        $r=$r + $result=mysqli_query($conexion,$sql);
        $idProducto=self::buscaProducto($d[1], $d[2]);
        if ($idProducto==0) {
            $Producto=self::insertarProducto($d[1], $d[2], $d[3]);
            $idProducto=self::buscaProducto($d[1], $d[2]);
            $sql2="INSERT INTO det_compra (Codigo_barra,
                                            Folio_compra,
                                            Cantidad,
                                            Precio)
                                            VALUES ('$idProducto',
                                            '$idCompra',
                                            '$d[3]',
                                            '$d[4]')";
            $resultado=mysqli_query($conexion,$sql2);
        } else {
            $sql2="INSERT INTO det_compra (Codigo_barra,
                                            Folio_compra,
                                            Cantidad,
                                            Precio)
                                            VALUES ('$idProducto',
                                            '$idCompra',
                                            '$d[3]',
                                            '$d[4]')";
            $resultado=mysqli_query($conexion,$sql2);
            $suma=self::buscarCantidad($idProducto, $d[3]);
            $sql3="UPDATE producto SET Cantidad_producto='$suma' WHERE Codigo_barra= '$idProducto'";
            $resultado2=mysqli_query($conexion,$sql3);
        }
    }
    return $r;
}

```

**Figura 76.** Código fuente para crear registro de compras

En la figura 77 se proyecta el código generado para la creación de un nuevo folio de compra

```

public function creaFolio(){
    include("conexion.php");
    $sql="SELECT Folio_compra from compra group by Folio_compra desc";

    $resul=mysqli_query($conexion,$sql);
    $id=mysqli_fetch_row($resul)[0];

    if($id=="" or $id==null or $id==0){
        return 1;
    }else{
        return $id + 1;
    }
}

```

**Figura 77.** Código fuente para crear folio de compra

En la figura 78 se proyecta el código generado para la actualización de cantidad de un producto al realizar una compra.

```
$suma=$self::buscarCantidad($idProducto, $d[3]);  
$sql3="UPDATE producto SET Cantidad_producto='$suma' WHERE Codigo_barra= '$idProducto';"  
$resultado2=mysqli_query($conexion,$sql3);
```

**Figura 78.** Código fuente para incrementar la cantidad del producto (1/2)

En la figura 79 se proyecta el código generado para la actualización de cantidad de un producto al realizar una compra.

```
public function buscarCantidad($e, $can){  
    include("conexion.php");  
    $sql="SELECT Cantidad_producto FROM $tabla_Producto WHERE '$e'= Codigo_barra;";  
  
    $resul=mysqli_query($conexion,$sql);  
    $id=mysqli_fetch_row($resul)[0];  
    $id= $id+$can;  
    return $id;  
}
```

**Figura 79.** Código fuente para incrementar la cantidad del producto (2/2)

En la figura 80 se proyecta el código generado para el registro de datos del módulo de ventas a la base de datos

```
public function crearVentas(){  
    include("conexion.php");  
    $fecha=date('Y-m-d');  
    $idVenta=$self::creaFolioVenta();  
    $datos=$_SESSION['tablaVentasTemp'];  
    $r=0;  
  
    for ($i=0; $i < count($datos) ; $i++) {  
        $d=explode("||", $datos[$i]);  
        $sql="INSERT INTO venta (Folio_de_venta,  
                                Fecha)  
                VALUES ('$idVenta',  
                            '$fecha')";  
  
        $r=$r + $result=mysqli_query($conexion,$sql);  
        $idProducto=$self::buscaProducto($d[0], $d[1]);  
        $sql2="INSERT INTO det_venta (Codigo_de_barra_del_producto,  
                                    Folio_de_venta,  
                                    Precio,  
                                    Cantidad)  
                VALUES ('$idProducto',  
                            '$idVenta',  
                            '$d[3]',  
                            '$d[2]')";  
  
        $resultado=mysqli_query($conexion,$sql2);  
        $suma=$self::decrementarCantidad($idProducto, $d[2]);  
        $sql3="UPDATE producto SET Cantidad_producto='$suma' WHERE Codigo_barra= '$idProducto';"  
        $resultado2=mysqli_query($conexion,$sql3);  
    }  
    return $r;  
}
```

**Figura 80.** Código fuente para crear registro de venta

En la figura 81 se proyecta el código generado para la creación de un nuevo folio de venta.

```
public function creaFolioVenta(){
    include("conexion.php");
    $sql="SELECT Folio_de_venta from venta group by Folio_de_venta desc";

    $resul=mysqli_query($conexion,$sql);
    $id=mysqli_fetch_row($resul)[0];

    if($id==" " or $id==null or $id==0){
        return 1;
    }else{
        return $id + 1;
    }
}
```

**Figura 81.** Código fuente para crear un folio de venta

En la figura 82 se proyecta el código generado para la actualización de cantidad de un producto al realizar una venta.

```
public function decrementarCantidad($e, $can){
    include("conexion.php");
    $sql="SELECT Cantidad_producto FROM $tabla_Producto WHERE '$e'=Codigo_barra;";

    $resul=mysqli_query($conexion,$sql);
    $id=mysqli_fetch_row($resul)[0];
    $id= $id-$can;
    return $id;
}
```

**Figura 82.** Código fuente para decrementar la cantidad del producto en existencia

## 6. Referencias bibliográficas

---

- [1] URL: <https://metodologia.es/prototipado/> blog en el cual se puede consultar información acerca de distintas metodologías.
- [2] Shari Lawrence Peleeger. Ingeniería de software, editorial MC Graw Hill, Madrid España, 2002.
- [3] URL: <https://helisulbaransistemas.blogspot.com/2014/09/paradigmas-en-el-desarrollo-de-software.html> blog personal donde se pueden encontrar temas relacionados con ingeniería de software.
- [4] Ing. Marcelo Carrera. Modelo de Desarrollo PROTOTIPADO, septiembre, 2005.
- [5] Marijn Haverbeke. Eloquent Javascript 3rd Edition, Autoedición, febrero, 2018.
- [6] Millet & Gil de la Iglesia, Laboratorio de PHP y MySQL, 2009.
- [7] Reginald Braithwaite, JavaScript Allongé (Free Edition), Leanpub, mayo, 2015.
- [8] Eguíluz Pérez, Javier, Introducción AJAX, Autoedición, junio, 2008.
- [9] Schifreen, Robert, How to create Web sites and applications with HTML, CSS, Javascript, PHP and MySQL, Oakworth Business, septiembre, 2010.
- [10] Bahit, Eugenia, POO y MVC en PHP, Autoedición, 2011.
- [11] URL: <https://www.genbeta.com/herramientas/sublime-text-un-sofisticado-editor-de-codigo-multiplataforma>, blog donde se consulta acerca de herramientas de diseño y desarrollo web.
- [12] URL: <http://www.fpdf.org> pagina oficial de la clase FPDF



Universidad Politécnica de Puebla  
Ingeniería en Informática

*José Alfredo Tlahuechtl Rodríguez*  
*Luis Tepox Romero*

*Rebeca Rodríguez Huesca*

Este documento se distribuye para los términos de la  
Licencia 2.5 Creative Commons (CC-BC-NC-ND 2.5 MX)