Aplicativo web para la gestión de inventario de microempresarios

Héctor Andrés Restrepo Noguera

Jonathan David Blanco del Toro

David Santiago Mendieta Rozo

Institución Universitaria Uninpahu

Facultad de Ingeniería

Tutor:

Oscar Barragán Páez

Bogotá, Colombia

30 de octubre del 2024

**Tabla de Contenido**

[**Análisis (completar)** 6](#_Toc181298908)

[**Componentes del Sistema:** 6](#_Toc181298909)

[**Requisitos Funciónales** 6](#_Toc181298910)

[**Registro de Usuarios:** 6](#_Toc181298911)

[**Inicio de Sesión:** 6](#_Toc181298912)

[**Gestión de Productos:** 6](#_Toc181298913)

[**Estadísticas de Ventas:** 6](#_Toc181298914)

[**Configuración de la Cuenta:** 7](#_Toc181298915)

[**Requerimientos No Funcionales** 7](#_Toc181298916)

[**Seguridad:** 7](#_Toc181298917)

[**Rendimiento:** 7](#_Toc181298918)

[**Usabilidad:** 7](#_Toc181298919)

[**Compatibilidad:** 7](#_Toc181298920)

[**Mantenimiento:** 7](#_Toc181298921)

[**Diseño** 7](#_Toc181298922)

[**Presentación:** 7](#_Toc181298923)

[**Lógica de Negocio:** 8](#_Toc181298924)

[**Capa de Datos:** 8](#_Toc181298925)

[**Modelado de Casos de Uso** 8](#_Toc181298926)

[**Gestión de Productos:** 8](#_Toc181298927)

[**Visualización de Estadísticas:** 8](#_Toc181298928)

[**Generación de Informes:** 8](#_Toc181298929)

[**Diseño de la Base de Datos** 8](#_Toc181298930)

[**Tabla de Productos:** 9](#_Toc181298931)

[**Tabla de Ventas:** 9](#_Toc181298932)

[**Tabla de Clientes:** 9](#_Toc181298933)

[**Desarrollo** 9](#_Toc181298934)

[**Pruebas** 9](#_Toc181298935)

[**Fundamentos Teóricos** 9](#_Toc181298936)

[**Gestión de Productos:** 9](#_Toc181298937)

[**Estadísticas:** 10](#_Toc181298938)

[**Generación de Informes:** 10](#_Toc181298939)

[**Lista de Tareas Pendientes:** 10](#_Toc181298940)

[**Ámbito del sistema** 10](#_Toc181298941)

[**Consideraciones técnicas** 11](#_Toc181298942)

[**Lenguajes utilizados** 11](#_Toc181298943)

[**HTML (HyperText Markup Language):** 11](#_Toc181298944)

[**CSS (Cascading Style Sheets):** 11](#_Toc181298945)

[**PHP (Hypertext Preprocessor):** 12](#_Toc181298946)

[**JavaScript:** 12](#_Toc181298947)

[**MongoDB:** 13](#_Toc181298948)

[**Análisis y diseño de la aplicación web** 13](#_Toc181298949)

[**Diagrama de casos de uso:** 13](#_Toc181298950)

[**Diagrama de clases:** 13](#_Toc181298951)

[**Requerimientos Funcionales** 14](#_Toc181298952)

[**RF01** 14](#_Toc181298953)

[**RF02** 14](#_Toc181298954)

[**RF03** 15](#_Toc181298955)

[**RF04** 15](#_Toc181298956)

[**RF05** 16](#_Toc181298957)

[**RF06** 17](#_Toc181298958)

[**RF07** 17](#_Toc181298959)

[**Requerimientos No Funcionales** 18](#_Toc181298960)

[**RNF01** 18](#_Toc181298961)

[**RNF02** 18](#_Toc181298962)

[**RNF03** 18](#_Toc181298963)

[**RNF04** 19](#_Toc181298964)

[**RNF05** 19](#_Toc181298965)

[**Diagramas UML** 20](#_Toc181298966)

[**Diagrama de clase:** 20](#_Toc181298967)

[**Diagrama de objetos:** 20](#_Toc181298968)

[**Diagrama de componentes:** 21](#_Toc181298969)

[**Diagrama de despliegue:** 21](#_Toc181298970)

[**Diagrama de paquetes:** 22](#_Toc181298971)

[**Diagrama de Estructura Compuesta:** 22](#_Toc181298972)

[**Diagrama de Perfiles:** 22](#_Toc181298973)

[**Diagramas de comportamiento:** 23](#_Toc181298974)

[**Diagrama de casos de uso:** 23](#_Toc181298975)

[**Diagrama de Actividades:** 24](#_Toc181298976)

[**Diagrama de Estado:** 24](#_Toc181298977)

[**Diagrama de Secuencia:** 24](#_Toc181298978)

[**Diagrama de Comunicación:** 25](#_Toc181298979)

[**Diagrama de Tiempos:** 25](#_Toc181298980)

[**Ejemplo** 25](#_Toc181298981)

## **Análisis**

### **Componentes del Sistema:**

"BusinessNet" incluye varias funcionalidades clave, como la Gestión de Productos, Estadísticas, Generación de Informes, Lista de Tareas Pendientes y Cierre de Sesión. Durante el análisis, se evidencia cómo cada uno de los componentes interactúa dentro del sistema, así tenemos una experiencia fluida. Por ejemplo, la Gestión de Productos debe estar sincronizada con el apartado de Estadísticas para que cualquier cambio en el inventario se refleje automáticamente en los informes y gráficos.

## **Requisitos Funciónales**

**Registro de Usuarios:** Los usuarios pueden registrarse en la plataforma proporcionando información básica.

**Inicio de Sesión:** Los usuarios pueden iniciar sesión con sus credenciales registradas.

**Gestión de Productos:** Los usuarios pueden agregar, editar y eliminar productos de su inventario.

**Estadísticas de Ventas:** La plataforma proporciona información sobre las ventas realizadas, como el total de ventas y las ventas por período.

**Configuración de la Cuenta:** Los usuarios pueden cambiar su contraseña, actualizar su información personal y eliminar su cuenta.

## **Requerimientos No Funcionales**

**Seguridad:** La plataforma debe proteger los datos de los usuarios mediante el cifrado de contraseñas y la implementación de medidas de seguridad.

**Rendimiento:** La plataforma debe ser capaz de manejar múltiples usuarios simultáneamente sin experimentar tiempos de carga excesivos.

**Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de usar para que los usuarios puedan navegar sin dificultades.

**Compatibilidad:** Debe ser compatible con una variedad de dispositivos y navegadores web para garantizar una experiencia consistente.

**Mantenimiento:** El sistema debe ser fácil de mantener y actualizar, con código limpio y documentación adecuada para facilitar futuras modificaciones.

En conclusión, el análisis incluye una comprobación de la viabilidad de nuestro sistema. Esto lleva asegurarnos de que "BusinessNet" no solo cumpla con los requisitos definidos, sino que también funcione de manera confiable y segura en el entorno definido. La viabilidad técnica y de implementación se considera para asegurarnos que el proyecto pueda ser un éxito y que el sistema cumpla con las expectativas de los usuarios y las necesidades del negocio.

## **Diseño**

Nuestro sistema "BusinessNet" se diseñará como una aplicación web.

**Presentación:** Incluye la interfaz de usuario, accesible a través de un navegador web. Esta se desarrollará con leguajes tales como HTML, CSS y JavaScript.

**Lógica de Negocio:** Contiene la lógica que maneja el funcionamiento del sistema, tales como la gestión de productos, la generación de estadísticas y la creación de informes.

**Capa de Datos:** Gestiona la persistencia de los datos en una base de datos no relacional, como MongoDB. Se diseñarán tablas que almacenen productos, clientes, ventas y usuarios. Hay que asegurarnos que las relaciones estén bien definidas para asegurar la integridad de los datos.

## **Modelado de Casos de Uso**

Para asegurarnos que todas las funcionalidades del sistema sean comprendidas y desarrolladas correctamente, se elaborarán modelos de casos de uso que describan las interacciones entre los usuarios y el sistema:

**Gestión de Productos:** Este caso de uso detallará cómo los usuarios pueden agregar, editar y eliminar productos. Teniendo en cuenta las validaciones necesarias (como la comprobación de campos obligatorios) y los resultados esperados para cada operación.

**Visualización de Estadísticas:** Otro caso de uso describirá cómo los diferentes usuarios pueden acceder y visualizar estadísticas de ventas y clientes.

**Generación de Informes:** Se modelará el proceso de creación de informes, incluyendo la selección de datos a incluir, el formato de salida seria (PDF).

## **Diseño de la Base de Datos**

El diseño de la base de datos será esencial para asegurar que la información sea manejada de manera eficiente y segura. Se crearán diagramas entidad-relación que representen las tablas y sus relaciones.

**Tabla de Productos:** Incluirá campos como ID, nombre, descripción, precio, imagen, y categoría,

**Tabla de Ventas:** Registrará cada transacción, enlazando productos con clientes y generando estadísticas de inmediato.

**Tabla de Clientes:** Almacenará la información de los clientes, permitiendo un seguimiento detallado del comportamiento de los clientes.

## **Desarrollo**

En la fase de desarrollo de nuestro sistema, se construye el producto siguiendo las especificaciones detalladas en la fase de diseño. Este proceso incluye la implementación de los diagramas de clase, secuencia, modelo entidad-relación, así como la realización de pruebas para asegurar que nuestro producto esté listo para ser utilizado por los usuarios finales.

## **Pruebas**

En esta parte se realiza la codificación de los componentes, se realiza la evaluación de la entrega del ejecutable, se realizan las pruebas de funcionamiento y se elaboran los manuales de usuario y manuales de sistema.

## **Fundamentos Teóricos**

**Gestión de Productos:** La funcionalidad de Gestión de Productos permite a los usuarios añadir, editar y eliminar productos dentro del sistema. Cada producto se le pueden agregar imágenes, precios y detalles específicos. La lógica de esta funcionalidad se basa en un sistema de base de datos no relacional trabajada con MongoDB. Los productos se organizan de manera que sean fácilmente accesibles y modificables por el usuario, lo que facilita la administración del inventario.

**Estadísticas:** Este apartado proporciona un análisis detallado de las ventas y la gestión de clientes. La aplicación recoge datos sobre el registro de ventas y clientes, calculando el total de ventas, los productos más vendidos y clientes más frecuentes, esta información se presenta a través de gráficos. Las métricas clave utilizadas en esta sección permiten a los usuarios tomar decisiones informadas sobre el rendimiento de su negocio.

**Generación de Informes:** La funcionalidad de Generación de Informes permite la creación de reportes detallados basados en los datos recopilados por el aplicativo. Estos informes pueden ser exportados en formatos PDF y son diseñados para ayudar en el análisis y seguimiento de la productividad de la microempresa.

**Lista de Tareas Pendientes:** La lista de Tareas Pendientes es una herramienta que ayuda a los usuarios a organizar y priorizar sus actividades diarias. Esta funcionalidad permite la creación, actualización y eliminación de tareas, lo que mejora la eficiencia en la gestión del tiempo.

## **Ámbito del sistema**

El aplicativo del sistema "Business Net" permite a los usuarios gestionar eficientemente sus productos, incluyendo la posibilidad de agregar, editar y eliminar productos con imágenes, precios y detalles completos. Además, ofrece un apartado de estadísticas que registra las ventas y los diferentes clientes, muestra el total de ventas, las ganancias obtenidas, y permite visualizar y generar informes detallados sobre las estadísticas de ventas. También incluye una lista de tareas pendientes y la opción de cerrar sesión para mantener la seguridad del usuario.

## **Consideraciones técnicas**

La base fundamental que se requiere del sistema informático debe tener una comunicación entre cliente y servidor, se daría mediante los estándares de la www y el estándar básico de comunicación por el protocolo http

## **Lenguajes utilizados**

Para el programa se van a utilizar los siguientes Lenguajes (HTML, CSS, PHP, JAVA SCRIPT, MONGODB,).

**HTML (HyperText Markup Language):** HTML es el lenguaje de marcado fundamental para la creación de páginas web. Se utiliza para estructurar el contenido y organizar la información en un formato que los navegadores web pueden interpretar y mostrar. HTML está compuesto por una serie de elementos o etiquetas que definen diferentes tipos de contenido, como encabezados, párrafos, enlaces, listas, tablas e imágenes. Cada etiqueta HTML tiene una función específica y puede tener atributos que proporcionan información adicional o modifican el comportamiento de la etiqueta. Por ejemplo, <a href="url">Enlace</a> crea un enlace que lleva al usuario a una URL especificada. HTML también proporciona una estructura jerárquica mediante el uso de contenedores como <div> y <span>, que permiten agrupar y organizar el contenido. A pesar de ser un lenguaje de marcado y no de programación, HTML es esencial para cualquier desarrollo web, ya que establece la base sobre la cual se aplica el estilo con CSS y se añade funcionalidad con JavaScript.

**CSS (Cascading Style Sheets):** CSS es un lenguaje de hojas de estilo que se utiliza para definir la presentación y el diseño de los documentos HTML. A través de CSS, los desarrolladores pueden controlar aspectos visuales como colores, fuentes, márgenes, espaciado, alineación y disposición de los elementos en una página web. CSS se basa en reglas y selectores; cada regla se aplica a elementos HTML específicos y define cómo deben ser estilizados. Las reglas CSS pueden aplicarse de tres formas: en línea, en el <head> del documento HTML mediante una etiqueta <style>, o en un archivo externo vinculado a través de una etiqueta <link>. El uso de CSS permite separar el contenido (HTML) de la presentación, lo que facilita la gestión y el mantenimiento de la apariencia del sitio web. CSS también incluye características avanzadas como media queries para diseño responsivo, animaciones y transiciones que mejoran la experiencia del usuario en diferentes dispositivos y tamaños de pantalla.

**PHP (Hypertext Preprocessor):** PHP es un lenguaje de scripting del lado del servidor que se utiliza para desarrollar aplicaciones web dinámicas y gestionar la interacción entre el servidor y el cliente. A diferencia de los lenguajes de programación que se ejecutan en el navegador del cliente, PHP se ejecuta en el servidor, generando HTML que luego se envía al navegador del usuario. PHP es especialmente útil para tareas como la manipulación de bases de datos, la gestión de sesiones de usuario y la generación de contenido dinámico basado en la interacción del usuario. El lenguaje se integra fácilmente con bases de datos como MySQL y PostgreSQL, permitiendo a los desarrolladores crear aplicaciones que pueden almacenar, recuperar y manipular datos. PHP también ofrece una amplia gama de bibliotecas y frameworks que simplifican el desarrollo y aumentan la eficiencia. Además, es compatible con la mayoría de los servidores web y sistemas operativos, lo que lo convierte en una opción versátil para el desarrollo web.

**JavaScript:** JavaScript es un lenguaje de programación del lado del cliente que se utiliza para crear interactividad y dinamismo en las páginas web. A diferencia de HTML y CSS, que definen la estructura y el estilo, JavaScript permite a los desarrolladores añadir comportamientos y funcionalidades que responden a las acciones del usuario. JavaScript puede manipular el DOM (Document Object Model) para modificar el contenido de la página, responder a eventos como clics y teclas, y realizar solicitudes asíncronas al servidor sin necesidad de recargar la página (AJAX). También se puede utilizar para crear animaciones, validar formularios y gestionar datos en tiempo real. Con la introducción de ECMAScript 6 (ES6) y versiones posteriores, JavaScript ha evolucionado significativamente, ofreciendo nuevas características como clases, módulos y funciones flecha que facilitan el desarrollo de aplicaciones complejas. Además, con la llegada de Node.js, JavaScript también se puede utilizar en el lado del servidor, permitiendo a los desarrolladores usar el mismo lenguaje en ambas partes de la aplicación web.

**MongoDB:** MongoDB es una base de datos NoSQL orientada a documentos que almacena datos en formato BSON (Binary JSON). A diferencia de las bases de datos relacionales tradicionales que utilizan tablas y filas, MongoDB organiza los datos en documentos similares a JSON, que se agrupan en colecciones. Esta estructura flexible permite almacenar datos semiestructurados y facilita la adaptación a cambios en el esquema. MongoDB es conocido por su capacidad de escalabilidad horizontal, permitiendo a los desarrolladores distribuir los datos en múltiples servidores para manejar grandes volúmenes de información y alta carga de trabajo. Además, ofrece características como consultas ad hoc, índices y agregaciones, lo que facilita la recuperación y el análisis de datos. MongoDB también se integra bien con lenguajes de programación populares y marcos de trabajo, proporcionando una solución robusta para aplicaciones web modernas que requieren flexibilidad y rendimiento.

## **Análisis y diseño de la aplicación web**

**Diagrama de casos de uso:** Define las interacciones entre los usuarios (actores) y el sistema, ayudando a identificar los requisitos funcionales.

**Diagrama de clases:** Modela las entidades, sus atributos, métodos y las relaciones entre ellas, representando la estructura del sistema.

# **Requerimientos Funcionales**

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RF01** |
| **Nombre del**  **Requerimiento:** | Diseño de Interfaz de Usuario (UI) |
| **Características:** | * Atractivo Visual: El diseño debe ser estéticamente agradable y atractivo para los usuarios. * Coherencia: La UI debe ser coherente en todas las páginas web que forman parte de la aplicación. * Usabilidad: La navegación dentro de la aplicación debe ser intuitiva y fácil de entender para los usuarios. |
| **Descripción del**  **requerimiento:** | Es un componente esencial en el desarrollo de cualquier aplicación web, ya que es la primera impresión que los usuarios tendrán de la misma. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RF02** |
| **Nombre del Requerimiento:** | Páginas y Formularios |
| **Características:** | * Las páginas deben permitir la navegación fácil entre secciones o páginas relacionadas a través de menús, botones o enlaces. |
| **Descripción del**  **requerimiento:** | Se debe diseñar un conjunto de páginas web y formularios que sean necesarios para las funciones clave de la aplicación, |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RF03** |
| **Nombre del**  **Requerimiento:** | Experiencia de Usuario (UX) |
| **Características:** | * Navegación Intuitiva: Los usuarios deben poder encontrar fácilmente las funciones que necesitan, con una navegación lógica y menús claros. * Retroalimentación Visual: Proporcionar retroalimentación visual a través de mensajes de éxito o error, animaciones suaves y transiciones fluidas. * Diseño Minimalista: Evitar la sobrecarga de información y elementos en pantalla para mantener la simplicidad y claridad. |
| **Descripción del**  **requerimiento:** | La interfaz debe centrarse en proporcionar una experiencia del usuario (UX) amigable y eficiente en todas las páginas y formularios. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RF04** |
| **Nombre del**  **Requerimiento:** | Integración con Lógica PHP |
| **Características:** | * Comunicación Eficiente: Asegurarse de que las páginas HTML y la lógica PHP se comuniquen de manera eficiente. * Validación de Datos: Implementar validaciones en ambas partes (HTML y PHP) para garantizar la seguridad y la integridad de los datos transmitidos y almacenados. * Pruebas de Integración: Realizar pruebas exhaustivas para garantizar que la integración entre la interfaz y la lógica sea sólida y sin errores. |
| **Descripción del**  **requerimiento:** | Integrar las páginas HTML diseñadas con la lógica PHP desarrollada por otro compañero, asegurando una interacción fluida entre la interfaz y la lógica. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RF05** |
| **Nombre del Requerimiento:** | Registro de Usuarios |
| **Características:** | * El formulario de registro debe recopilar la información necesaria de los usuarios, como nombre, dirección de correo electrónico y contraseña; permitiendo modificar, consultar y eliminar el registro |
| **Descripción del**  **requerimiento:** | Se debe diseñar un apartado en las páginas web capaz de realizar la función de registro y almacenarla en la base de datos. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RF06** |
| **Nombre del Requerimiento:** | Inicio de Sesión |
| **Características:** | * La página de inicio de sesión debe permitir a los usuarios ingresar sus credenciales de manera segura. |
| **Descripción del**  **requerimiento:** | Se debe diseñar un apartado en las páginas web capaz de realizar la función de validación de las credenciales almacenadas en la base de datos para que deje al usuario realizar actividades dentro de la página. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RF07** |
| **Nombre del Requerimiento:** | Módulo de Reporte |
| **Características:** | * La página debe contener un apartado que le permita al usuario llevar el control de los diferentes movimientos y procesos del inventariado y mostrar los diferentes reportes. |
| **Descripción del**  **requerimiento:** | La plataforma proporciona información sobre las ventas realizadas, como el total de ventas y las ventas por período. La plataforma proporciona información sobre las ventas realizadas, como el total de ventas y las ventas por período. |
| **Prioridad del requerimiento:**  Alta | |

# **Requerimientos No Funcionales**

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RNF01** |
| **Nombre del**  **Requerimiento:** | sistema y transacción |
| **Características:** | El sistema presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema. |
| **Descripción del**  **requerimiento:** | Toda funcionalidad del negocio debe responder al usuario en menos de 5 segundos. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RNF02** |
| **Nombre del Requerimiento:** | Actualización de Datos |
| **Características:** | La Base de datos debe estar normalizada y actualizada diariamente. |
| **Descripción del**  **requerimiento:** | No se puede Perder información y debe tener un backup diario de toda la base. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RNF03** |
| **Nombre del**  **Requerimiento:** | Seguridad Informática |
| **Características:** | El sistema debe de contar con roles de usuario y un súper usuario a administrador el cual podrá modificar los datos de usuario si es necesario realizarlos. |
| **Descripción del**  **requerimiento:** | Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | **RNF04** |
| **Nombre del Requerimiento:** | Interfaz Grafica |
| **Características:** | El diseño debe ser los más complejo posible. |
| **Descripción del requerimiento:** | El usuario debe sentirse cómodo con el aplica. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del**  **requerimiento:** | **RNF05** |
| **Nombre del Requerimiento:** | Carga Rápida de Páginas |
| **Características:** | * Comprimir y minimizar recursos, como imágenes, CSS y JavaScript, para reducir el tiempo de carga. * Utilizar técnicas de almacenamiento en caché para acelerar la entrega de contenido. * Minimizar solicitudes de servidor y reducir el tamaño de los archivos para una carga más eficiente. |

# **Diagramas UML**

Este modelo dispone de varios diagramas que ayudan a comprender la complejidad del futuro sistema, permitiéndonos mostrar en un lenguaje estándar aquellas funcionalidades,requisitos,y demás características que hemos detectado en el sistema. UML define una gran cantidad de diagramas para representar los diferentes aspectos del desarrollo de la aplicación entre estos tenemos

**Diagrama de clase:** es el cual representa la estructura estática del sistema, mostrando clases, atributos y relaciones.

**Diagrama

Descripción generada automáticamente**

**Diagrama de objetos:** Es una instancia de un diagrama de clases, mostrando objetos específicos y sus relaciones.

**Diagrama

Descripción generada automáticamente**

**Diagrama de componentes:** Representa la organización física de un sistema en componentes y sus sistemas.

Diagrama

Descripción generada automáticamente

**Diagrama de despliegue:** muestra la configuración física de un sistema en tiempo de ejecución.

Diagrama

Descripción generada automáticamente

**Diagrama de paquetes:** Organiza elementos en gropos lógicos.

Diagrama, Forma, Rectángulo

Descripción generada automáticamente

**Diagrama de Estructura Compuesta:** Muestra la estructura interna de un clasificador.

Diagrama

Descripción generada automáticamente

**Diagrama de Perfiles:** Define extensiones en el metamodelo UML

Imagen que contiene Diagrama

Descripción generada automáticamente

Los anteriores hacen parte de los diagramas estructurales.

**Diagramas de comportamiento:** Estos diagramas muestran la dinámica del sistema, incluyendo los procesos y las interacciones entre objetos entre estos tenemos:

Diagrama

Descripción generada automáticamente

**Diagrama de casos de uso:** Muestra como los actores interactúan con el sistema

Diagrama

Descripción generada automáticamente

**Diagrama de Actividades:** Representan el flujo de trabajo de un proceso.

Diagrama

Descripción generada automáticamente

**Diagrama de Estado:** Muestra los posibles estados de un objeto y las transiciones entre ellos.

Diagrama

Descripción generada automáticamente

**Diagrama de Secuencia:** Muestra la interacción entre objetos a lo largo del tiempo.

Diagrama

Descripción generada automáticamente

**Diagrama de Comunicación:** Es una variante del diagrama de secuencia que enfatiza las comunicaciones entre objetos

Diagrama

Descripción generada automáticamente

**Diagrama de Tiempos:** Muestra la interacción entre varios objetos a lo largo del tiempo.

Gráfico

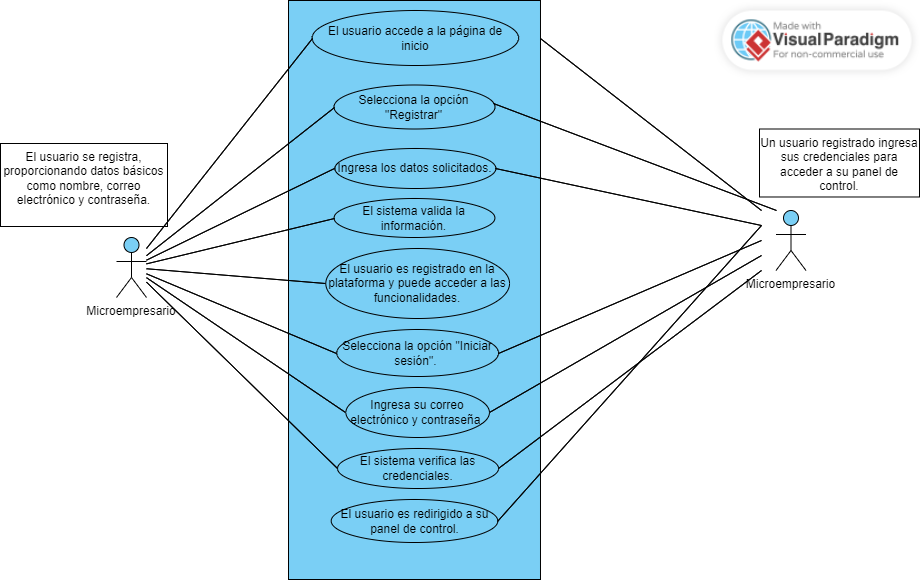
Descripción generada automáticamente con confianza baja

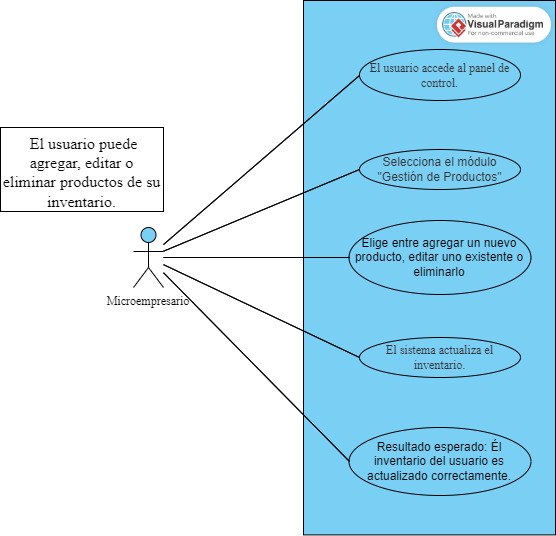
Ponerles nombre a los diagramas es para mantener una mejor comunicación entre los miembros de un equipo:

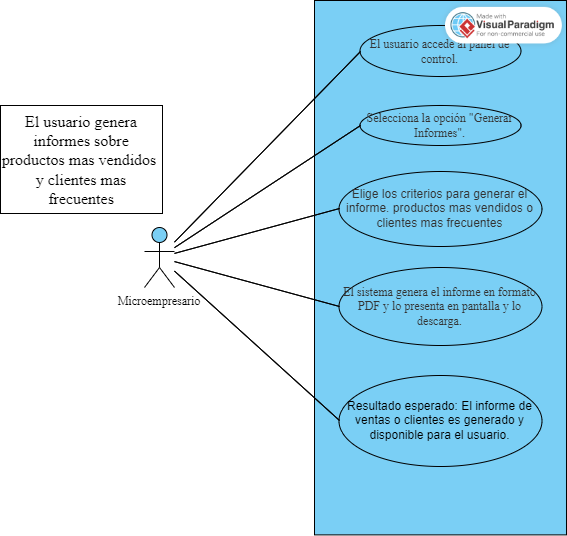
## **Ejemplo**

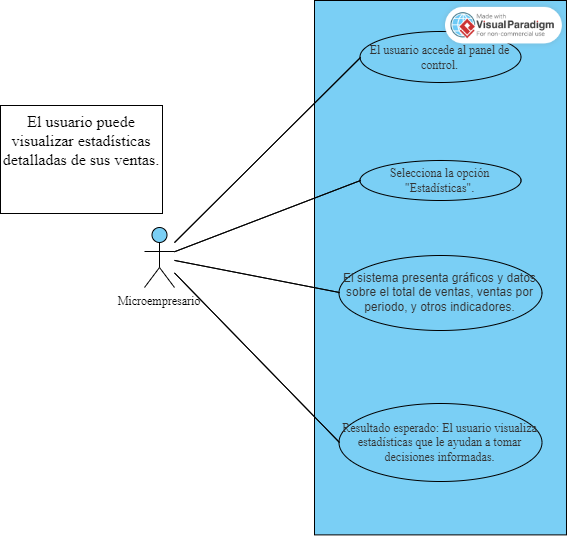
Si estamos modelando un sistema de comercio electrónico, podríamos hacer los siguientes diagramas: “diagramas de casos de uso, diagramas de clases de productos, diagramas de secuencias de compra, diagrama de estados de un pedido, diagramas de despliegue de un servidor web”

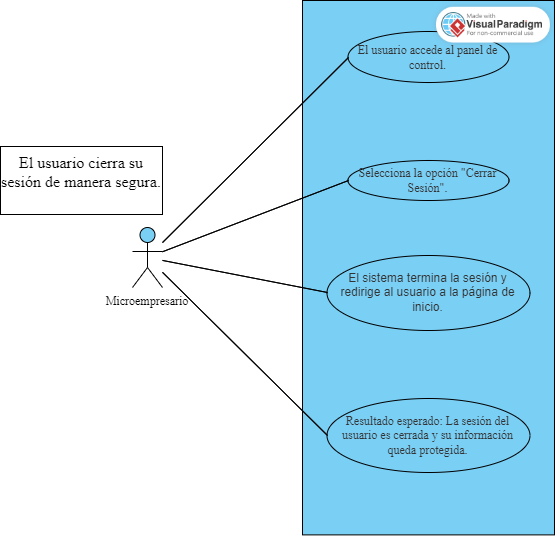
**Diagrama de Caso de Uso General**



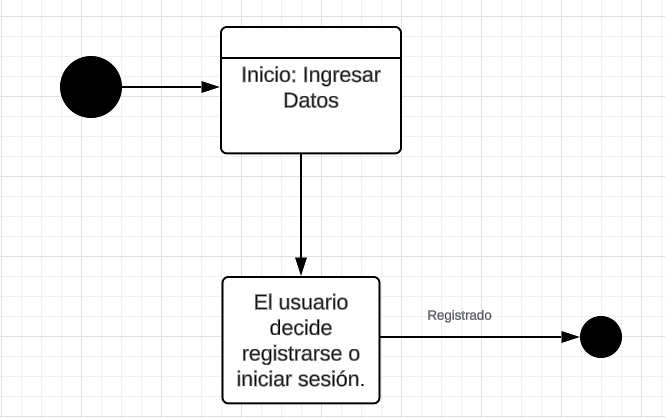


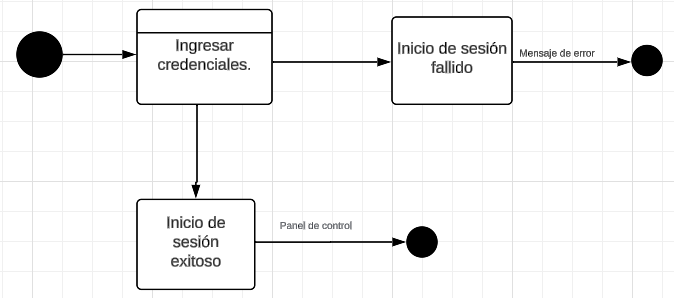


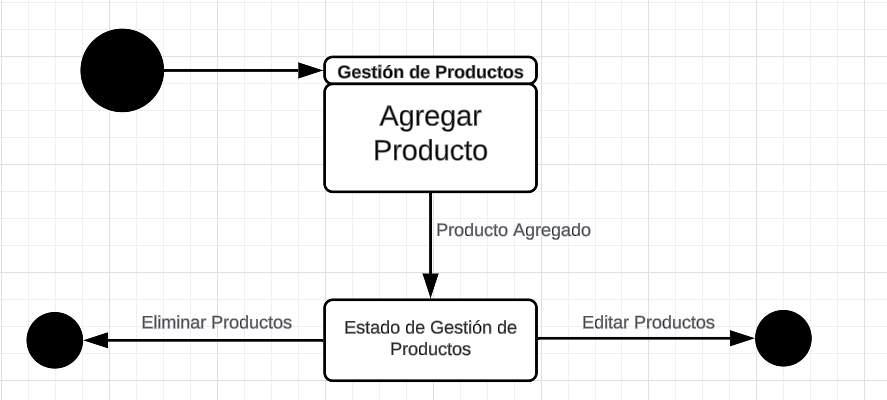




**Realizar Diagrama De Estado**







**Bibliografía**

**Mozilla Developer Network (MDN) - HTML**:

[MDN Web Docs: HTML](https://developer.mozilla.org/en-US/docs/Web/HTML)

Una guía completa y actualizada sobre HTML, incluyendo elementos, atributos y prácticas recomendadas.

**Mozilla Developer Network (MDN) - CSS**:

[MDN Web Docs: CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)

Documentación detallada sobre CSS, incluyendo propiedades, selectores y ejemplos.

**PHP Official Documentation**:

[PHP Manual](https://www.php.net/manual/en/)

Documentación oficial de PHP, que cubre funciones, clases y buenas prácticas para el desarrollo en PHP.

**W3Schools - JavaScript Tutorial**:

W3Schools: JavaScript

Tutorial interactivo de JavaScript con ejemplos y ejercicios para aprender el lenguaje.

**MongoDB Official Documentation**:

[MongoDB Documentation](https://www.mongodb.com/docs/)

Documentación oficial sobre MongoDB, incluyendo instalación, configuración y uso de la base de datos.