Texto

Descripción generada automáticamente

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA

Ingeniería en sistemas de información y

Ciencias de la computación

**Proyecto Sistema de Ventas**

**Documentación Final**

Omar Alejandro Bermúdez Cáceres

Álvaro Vinicio Orellana Méndez

Carlos Eduardo Camey Milian

Herbert Aragón Monzón

Maycol Arnulfo Guerra

Kevin Mauricio Pumay Godoy

Villanueva, octubre de 2024

**Índice**

[Introducción 2](#_gjdgxs)

[Minutas de reuniones 3](#_1fob9te)

[Hoja de ruta 5](#_3znysh7)

[Dercas 7](#_2et92p0)

[Acta de constitución 13](#_tyjcwt)

[Cronograma de actividades 19](#_3dy6vkm)

[Modelado de procesos BPM 20](#_1t3h5sf)

[Caso de uso 22](#_4d34og8)

[Informe de pruebas de seguridad 34](#_2s8eyo1)

[Informe de resultado de pruebas funcionales y no funcionales 36](#_17dp8vu)

[Plantilla de pruebas de aceptación 38](#_3rdcrjn)

[Carta de aceptación del negocio 40](#_26in1rg)

[Receta de publicación en producción 41](#_35nkun2)

[Informe de control de cambios 43](#_1ksv4uv)

[Guía técnica 47](#_44sinio)

[Conclusión 50](#_2jxsxqh)

## Introducción

Este trabajo documenta el desarrollo de un sistema de ventas online, realizado por los presentes antes mencionados. El proyecto se enmarca en el curso de Aseguramiento de la Calidad de Software y tiene como objetivo principal garantizar que el sistema cumpla con los estándares de calidad esperados mediante la implementación de prácticas rigurosas desde las fases iniciales hasta la validación final.

El sistema propuesto busca optimizar los procesos comerciales mediante una plataforma web que permita gestionar inventarios, usuarios y transacciones. La metodología utilizada para el desarrollo incluye prácticas ágiles como Scrum, asegurando una entrega incremental y continua del producto. Además, se han considerado aspectos críticos como la seguridad, el rendimiento y la escalabilidad del sistema.

## Minutas de reuniones

**MINUTA No. 1**

**FECHA: 21/07/2024**

**LUGAR: C204**

**HORA: 10:40 a 11:00**

**Participantes:**

| **No.** | **Nombre** |
| --- | --- |
| 1 | Omar Alejandro Bermúdez Cáceres |
| 2 | Álvaro Vinicio Orellana Méndez |
| 3 | Herbert Alejandro Aragón Monzón |
| 4 | Carlos Eduardo Camey Milian |
| 5 | Maycol Arnulfo Guerra Trigueros |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Proyecto | Sistema de ventas con Angular utilizando Apis. Desarrollo en n capas. Base de datos SQLServer. |
| 2 | Scrum Team | Scrum Master: Omar Bermúdez  Producto Owner: Álvaro Orellana  Developer: Herbert Aragon  Tester: Maycol Guerra  Requerimientos: Carlos Camey |

**Tareas Asignadas:**

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Leer la Guía de Scrum | Todo scrum Team | 27/07/2024 |  |
| 2 | Preparar el entorno de desarrollo | Developer Team | 27/07/2024 |  |

**Decisiones y comentarios:**

1. Se realizará un sistema web de ventas y facturación. Utilizando el framework de Angular.
2. Se estudiará la guía de Scrum.

**MINUTA No. 2**

**FECHA: 28/07/2024**

**LUGAR: C204**

**HORA: 10:40 a 11:00**

**Participantes:**

| **No.** | **Nombre** |
| --- | --- |
| 1 | Omar Alejandro Bermúdez Caceres |
| 2 | Álvaro Vinicio Orellana Méndez |
| 3 | Herbert Alejandro Aragón Monzón |
| 4 | Carlos Eduardo Camey Milian |
| 5 | Maycol Arnulfo Guerra Trigueros |
| 6 | Kevin Mauricio Pumay Godoy |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Proyecto | Sistema de ventas online con Angular utilizando Apis. Desarrollo en n capas. Base de datos SQLServer o Flutter y Firebase. |
| 2 | Scrum Team | Scrum Master: Omar Bermúdez  Producto Owner: Álvaro Orellana  Developer: Herbert Aragon  Tester: Maycol Guerra  Tester: Kevin Pumay  Requerimientos: Carlos Camey |

**Tareas Asignadas:**

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Leer la Guía de Scrum | Todo scrum Team | 04/08/2024 | 1 |
| 2 | Preparar el entorno de desarrollo | Developer Team | 04/08/2024 | 1 |
| 3 | Revisar la documentación/ requerimientos | Todo Scrum Team | 04/0802024 |  |

**Decisiones y comentarios:**

1. Se realizará un sistema web de ventas y facturación. Utilizando el framework de Angular.
2. Se estudiará la guía de Scrum.

| **MINUTA No.** | **3** |
| --- | --- |
| **FECHA:** | **04/08/2024** |
| **LUGAR:** | **C204** |
| **HORA:**  **Participantes:** | **10:40 a 11:00** |

| **No.** | **Nombre** |
| --- | --- |
| 1 | Omar Alejandro Bermudez Caceres |
| 2 | Alvaro Vinicio Orellana Mendez |
| 3 | Herbert Alejandro Aragon Monzon |
| 4 | Carlos Eduardo Camey Milian |
| 5 | Maycol Arnulfo Guerra Trigueros |
| 6 | Kevin Mauricio Pumay Godoy |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Definir herramienta para seguimiento de actividades | Notion |
| 2 | Estudiar e investigar que es una hoja de ruta. | Fecha de entrega 24-08-2024 |

**Tareas Asignadas:**

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces**  **Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Preparar el entorno de desarrollo | Developer Team | 05/08/2024 | 2 |
| 2 | Revisar la documentación/ requerimientos | Todo Scrum Team | 05/0802024 | 1 |
| 3 | Organizar las tareas en Notion | Scrum Master | 05/08/2024 |  |

**Decisiones y comentarios:**

1. Se realizará un sistema web de ventas y facturación. Utilizando el framework de Angular.
2. Se estudiará la guía de Scrum.
3. Se usará Notion como herramienta de seguimiento de tareas

**MINUTA No. 4**

**FECHA: 18/08/2024**

**LUGAR: C204**

**HORA: 10:40 a 11:00**

**Participantes:**

| **No.** | **Nombre** |
| --- | --- |
| 1 | Omar Alejandro Bermúdez Cáceres |
| 2 | Álvaro Vinicio Orellana Méndez |
| 3 | Herbert Alejandro Aragón Monzón |
| 4 | Carlos Eduardo Camey Milian |
| 5 | Maycol Arnulfo Guerra Trigueros |
| 6 | Kevin Mauricio Pumay Godoy |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Definir herramienta para seguimiento de actividades | Notion |
| 2 | Estudiar e investigar que es una hoja de ruta. | Fecha de entrega 24-08-2024 |
| 3 | Definir el responsable para hacer hoja de ruta | Scrum Master y PO |

**Tareas Asignadas:**

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Preparar el entorno de desarrollo | Developer Team | 05/08/2024 | 2 |
| 2 | Revisar la documentación/ requerimientos | Todo Scrum Team | 05/0802024 | 1 |
| 3 | Organizar las tareas en Notion | Scrum Master | 05/08/2024 |  |
| 4 | Relizar hoja de ruta | SM Y PO | 22/08/2024 |  |

**Decisiones y comentarios:**

1. Se realizará un sistema web de ventas y facturación. Utilizando el framework de Angular.
2. Se estudiará la guía de Scrum.
3. Se usará Notion como herramienta de seguimiento de tareas.

El Scrum Master y Product Owner realizaran la hoja de ruta.

**MINUTA No. 6**

**FECHA: 01/09/2024**

**LUGAR: C204**

**HORA: 10:40 a 11:00**

**Participantes:**

| **No.** | **Nombre** |
| --- | --- |
| 1 | Omar Alejandro Bermudez Caceres |
| 2 | Alvaro Vinicio Orellana Mendez |
| 3 | Herbert Alejandro Aragon Monzon |
| 4 | Carlos Eduardo Camey Milian |
| 5 | Maycol Arnulfo Guerra Trigueros |
| 6 | Kevin Mauricio Pumay Godoy |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Completar Backlog en Taiga |  |
| 2 | Acta de Constitucion | Se entrega primera semana de septiembre |

**Tareas Asignadas:**

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Completar el Backlog en Taiga | SM | 05-09-2024 |  |
| 2 | Investigar Acta de Constitucion | SM, PO | 06-09-2024 |  |
| 3 | Hacer Acta de Constitucion | SM,PO | 06-09-2024 |  |

**Decisiones y comentarios:**

1. Se investigará sobre cómo hacer un acta de constitución
2. Se usará plantilla de canvas

| **MINUTA No.** | **7** |
| --- | --- |
| **FECHA:** | **08/09/2024** |
| **LUGAR:** | **Universidad Mariano Galvez de Guatemala** |
| **HORA:** | **10:40 a 11:00** |

**Participantes:**

| **No.** | **Nombre** | **Area** |
| --- | --- | --- |
| 1 | Omar Alejandro Bermudez | Scrum Master |
| 2 | Alvaro Orellanda | Product Owner |
| 3 | Herbert Alejandro Aragon | Developer |
| 4 | Carlos Eduardo Camey | Tester |
| 5 | Maycol Guerra Trigueros | Tester |
| 6 | Kevin Pumay Godoy | Tester |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Elaboración de Modelado de procesos BPM | Entregable No.3 |

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces**  **Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Instalación de Bizagi | Omar Bermudez | 10/09/2024 |  |
| 2 | Modela de proceso BPM | Omar Bermudez | 12/09/2024 |  |

**Tareas Asignadas:**

**Decisiones y comentarios:**

1. Se instalará la herramienta Bizagi en la maquina física de Omar Bermudez para elaborar el modelado de datos BPM

**MINUTA No. 8**

**FECHA: 22/09/2024**

**LUGAR: Universidad Mariano Gálvez de Guatemala**

**HORA: 10:40 a 11:00**

**Participantes:**

| **No.** | **Nombre** | **Area** |
| --- | --- | --- |
| 1 | Omar Alejandro Bermudez | Scrum Master |
| 2 | Alvaro Orellanda | Product Owner |
| 3 | Herbert Alejandro Aragon | Developer |
| 4 | Carlos Eduardo Camey | Tester |
| 5 | Maycol Guerra Trigueros | Tester |
| 6 | Kevin Pumay Godoy | Tester |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Elaboración de Casos de Uso | Entregable No.4 |

**Tareas Asignadas:**

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Análisis de Requerimientos | Alvaro Orellana | 25/09/2024 |  |
| 2 | Redacción de casos de uso | Alvaro Orellana | 25/09/2024 |  |

**Decisiones y comentarios:**

1. Se realizará un análisis previo a la redacción de los casos de uso del sistema
2. Se usará la plantilla brindada por el catedrático.

| **MINUTA No.** | **9** |
| --- | --- |
| **FECHA:** | **06/10/2024** |
| **LUGAR:** | **Universidad Mariano Gálvez de Guatemala** |
| **HORA:** | **10:40 a 11:00** |

**Participantes:**

| **No.** | **Nombre** | **Area** |
| --- | --- | --- |
| 1 | Omar Alejandro Bermudez | Scrum Master |
| 2 | Alvaro Orellanda | Product Owner |
| 3 | Herbert Alejandro Aragon | Developer |
| 4 | Carlos Eduardo Camey | Tester |
| 5 | Maycol Guerra Trigueros | Tester |
| 6 | Kevin Pumay Godoy | Tester |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Avance de desarrollo | Entregable No.4 |

**Tareas Asignadas:**

| **No.** | **Tareas Responsable** | | **Fecha Límite** | **Veces**  **Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Revision de Avance de desarrollo | Alvaro Orellana | 06/10/2024 |  |

**Decisiones y comentarios:**

1. Se realizará un análisis previo a la redacción de los casos de uso del sistema
2. Se usará la plantilla brindada por el catedrático.

**MINUTA No. 10**

**FECHA: 13/10/2024**

**LUGAR: Universidad Mariano Gálvez de Guatemala**

**HORA: 10:40 a 11:00**

**Participantes:**

| **No.** | **Nombre** | **Area** |
| --- | --- | --- |
| 1 | Omar Alejandro Bermudez | Scrum Master |
| 2 | Alvaro Orellanda | Product Owner |
| 3 | Herbert Alejandro Aragon | Developer |
| 4 | Carlos Eduardo Camey | Tester |
| 5 | Maycol Guerra Trigueros | Tester |
| 6 | Kevin Pumay Godoy | Tester |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Pruebas Unitarias | Entregable No.6 |

**Tareas Asignadas:**

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Revisión de Avance de desarrollo | Alvaro Orellana | 06/10/2024 |  |
| 2 | Elaboración de pruebas unitarias | Carlos Camey | 13/10/2024 |  |
| 3 | Informe de pruebas unitarias | Carlos Camey | 13/10/2024 |  |

**Decisiones y comentarios:**

1. Se realizarán las pruebas unitarias al software según contenido visto en clase.

**MINUTA No. 11**

**FECHA: 21/10/2024**

**LUGAR: C204**

**HORA: 10:40 a 11:00**

**Participantes:**

| **No.** | **Nombre** |
| --- | --- |
| 1 | Omar Alejandro Bermudez Caceres |
| 2 | Alvaro Vinicio Orellana Mendez |
| 3 | Herbert Alejandro Aragon Monzon |
| 4 | Carlos Eduardo Camey Milian |
| 5 | Maycol Arnulfo Guerra Trigueros |
| 6 | Kevin Mauricio Pumay Godoy |

**Temas:**

| **No.** | **Temas** | **Comentarios** |
| --- | --- | --- |
| 1 | Pruebas Funcionales y No Funcionales | Usar Robot Framework y Jmeter |
| 2 | Pruebas de Seguridad | Herramienta Libre |

**Tareas Asignadas:**

| **No.** | **Tareas** | **Responsable** | **Fecha Límite** | **Veces Pospuesta** |
| --- | --- | --- | --- | --- |
| 1 | Pruebas Func y No Func | Carlos Camey | 24/10/24 | 0 |
| 2 | Pruebas de seguridad | Maycol Guerra | 24/10/24 | 0 |

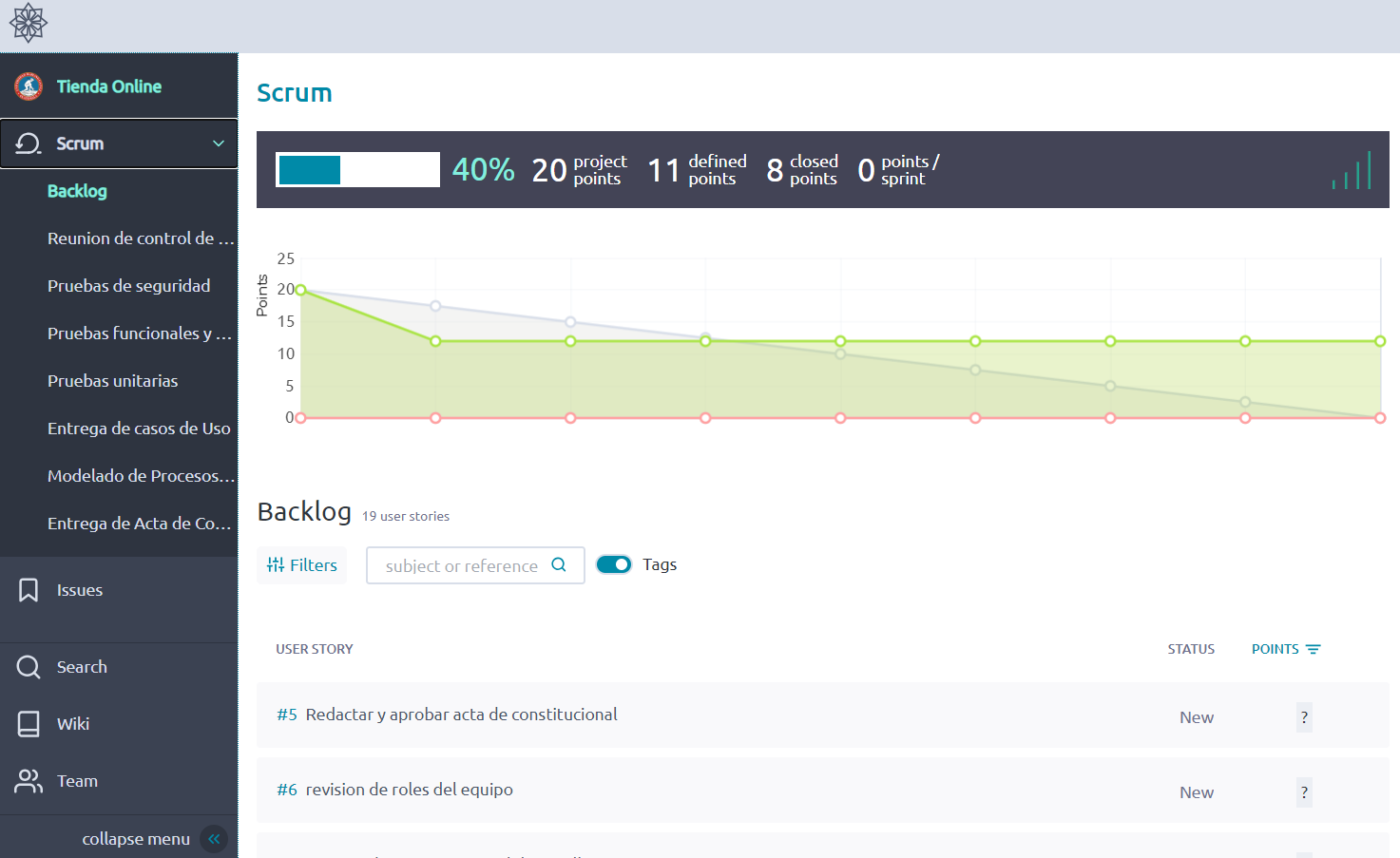
**Decisiones y comentarios:**

1. Se debe definirá la herramienta a utilizar para las pruebas de seguridad.

## Hoja de ruta

La hoja de ruta del proyecto detalla los pasos y actividades necesarias para completar el desarrollo del sistema de ventas online. Incluye fechas clave, responsables de cada tarea y los entregables esperados. Esta planificación es esencial para asegurar que el proyecto se mantenga en el camino correcto y se cumplan los plazos establecidos.

En la hoja de ruta están las fechas de entrega de cada uno de los componentes del proyecto, como la creación de la hoja de ruta, el acta de constitución, el modelado de procesos BPM, los casos de uso, las pruebas unitarias, las pruebas funcionales y no funcionales, y las pruebas de seguridad. Cada tarea tiene asignado un responsable, lo que facilita el seguimiento y la gestión del proyecto.



**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Teams

Descripción generada automáticamente**

## Dercas

**Propósito:** Es describir de forma clara concisa, específica, detallada e inequívoca los requerimientos del software para el sistema de ventas online.

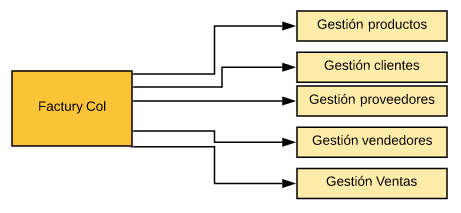
**Ámbito del sistema:** este producto se implementará en un entorno web.

**Definiciones acrónicas y abreviaturas**

| SGBD | Sistema gestor de Base de datos. |
| --- | --- |
| UML | Lenguaje unificado para modelamiento de sistemas |
| HTTPS | Protocolo de internet para indicar que el sitio es seguro |

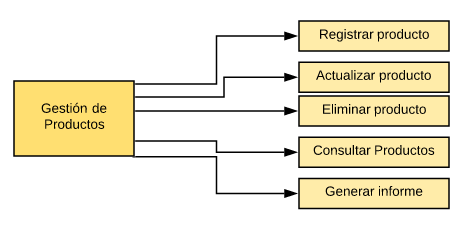
**Perspectiva del producto:** El sistema de ventas online permitirá administrar los inventarios de productos, usuarios, contraseñas y elaboración de reportes.

**Funciones del producto**



**Figura 1: Diagrama de descomposición de requerimientos**

**Gestión de producto ejemplo**



**Figura 2: Diagrama de descomposición Gestión de Productos**

Es un conjunto de requisitos relacionados con la gestión de productos, se centra en los registros, actualización, eliminación, consulta y generar informes de los productos

**Registrar producto**

**Descripción, entrada, proceso, salida, requerimientos no funcionales**

**Descripción**

Su función consiste en registrar un producto nuevo en la base de datos. Para ello el usuario debe ingresar sus datos en un formulario el cual esta sincronizado con la BD.

**Entrada**

* Id del producto
* Nombre del producto
* Descripción del producto
* cantidad
* Tamaño
* Precio unitario 12.000

**Proceso**

Una vez haya accedido a esta utilidad del sistema se introducen todos los datos del producto, estos serán validados por el sistema, comprobando la existencia previa del producto en la BD, o en su efecto generando un nuevo registro, Una vez realizada se presenta en pantalla un mensaje de confirmación de la operación.

Cualquier error que se produzca durante el proceso debe ser notificado al usuario mediante un mensaje descriptivo en pantalla.

**Salida**

Registro de la BD: Mensaje de confirmación con el resultado del nuevo registro (OK)

Formato de factura para imprimir

**Requerimientos específicos no funcionales**

Base de datos: El registro de los datos en la BD se debe realizar en un máximo de 2 segundos

Seguridad: una vez almacenados los datos estos serán encriptados por la BD

**Actualizar producto**

**Introducción:** esta funcionalidad permite actualizar los productos de la BD del software

**Entrada**

* Id producto
* Nombre
* Precio unitario 12.123

**Proceso**

Se procede a actualizar el producto en el software

**Salida**

Mensaje el producto fue actualizado

**Requerimientos no funcionales**

El proceso de actualización en el sistema no tardará más de 2 segundos

**Características de los usuarios**

Tener conocimientos básicos en informática

**Restricciones**

**Limitaciones del hardware**

Disco duro espacio 26 GB

RAM 2 GB

Tarjeta gráfica

**Operaciones paralelas**

No se contemplan

**Funciones de auditoría**

Para el cálculo o condensación del IVA

**Funciones de control**

Generar reportes diarios

**Lenguajes de programación**

MySQL

.net

Angular

TypeScript

Html

**Protocolos de comunicación**

Los protocolos de comunicaciones entre los diferentes nodos de la infraestructura hardware de soporte serán los siguientes:

* HTTPS para conexiones con el servidor web.
* TCP/IP a nivel físico.
* Ethernet 802.3 a nivel eléctrico.

**Consideraciones de seguridad**

* Para acceso al sistema aplicar claves robustas
* tokens

**Criticalidad de la aplicación**

* máximo permite 1000 usuarios conectados a la vez

**Suposiciones y dependencias**

* contabilidad
* facturación
* ventas

**Requerimientos futuros**

No Aplica

**Interfaces de Usuario**

Administrador, Cliente

**Interfaces Software**

Aplicación orientada a la web

**Interfaces de Comunicación**

N/A

**Requerimientos de rendimiento**

Número de usuarios conectados 10

**Requerimientos específicos**

* Usuarios -Perfiles (cliente, administrador) o Recuperación de contraseñas
* Empresa
* Contactos
* Catálogo de Artículos (categoría, subcategoría) o Clientes, Proveedores
* Tipos de pago (depósitos, tarjeta (paypal) )
* Pedidos (Kardex)
* Carrito de compras (Kardex)
* Historio de transacciones

**Reportes**

* Existencia de artículos (búsqueda por id)
* Pedidos y Ventas (búsqueda por id de transacción) o Kardex (filtro por fecha y por id articulo)
* Histórico de transacciones (filtrar por id cliente)

## Acta de constitución

| 1. **IDENTIFICACIÓN DEL PROYECTO** | |
| --- | --- |
| **Nombre del proyecto:** | Sistema de ventas online |
| **Patrocinador:** | Universidad Mariano Gálvez |
| **Tipo de Proyecto:** | Desarrollo / Mantenimiento |

| 1. **PERFIL DEL PROYECTO** |
| --- |
| 1. **DESCRIPCIÓN DEL PROYECTO:** |
| Sistema de ventas online que permite la gestión de usuarios, CRUD para inventarios y ventas y generación de reportes de ventas. |
| 1. **JUSTIFICACION DEL PROYECTO** |
| Este proyecto, desarrollado en el contexto del curso de Aseguramiento de la Calidad de Software, aborda la necesidad crítica de garantizar que los sistemas de ventas en línea cumplan con los estándares de calidad esperados. Al implementar un sistema web de ventas y realizar pruebas exhaustivas, se busca demostrar la relevancia de aplicar prácticas de aseguramiento de calidad desde las primeras fases del desarrollo, hasta su validación final. |
| 1. **ALCANCE DEL PROYECTO:** |
| Usuarios (Cliente – administrador), Recuperación de contraseña, Empresa, Contactos, catálogo de artículos, clientes y proveedores, tipos de pago, pedidos (Kardex), carrito de compras, histórico de transacciones, Reportes: Existencia de artículos, pedidos y ventas, Kardex, Histórico de transacciones. |
| 1. **LÍMITES AL ALCANCE DEL PROYECTO** |
| **Pruebas de Calidad de Alto nivel.** |
| 1. **REQUISITO DE TIEMPO** |
| Entrega final sábado 26 de Octubre. Entregas parciales de elementos y documentación de Control de calidad según Sprints. |
| 1. **REQUISITO DE CALIDAD** |
| El avance del proyecto será medido por medio del marco SCRUM, Requisitos y documentación de calidad:  Dercas: 31/08/2024  Acta de constitución  Modelado de procesos BPM  Casos de uso  Pruebas unitarias  Pruebas funcionales y no funcionales  Pruebas de seguridad  Reunión de Control de Cambios (Proyecto finalizado, publicado en producción) |
| 1. **COSTOS DEL PROYECTO** |
| Este proyecto será desarrollado con tecnologías Open Source por lo que no tendrá costos asociados a licencias de software. |
| 1. **RIESGO INICIAL DE ALTO NIVEL** |
| Tiempo de desarrollo. |

| 1. **REQUERIMIENTOS GLOBALES** | |
| --- | --- |
| 1. **REQUERIMIENTOS NO FUNCIONALES:** | |
| ∙  El sistema debe ser capaz de manejar hasta 10 transacciones simultáneas sin afectar su rendimiento.  ∙  Las consultas a la base de datos (MySQL) deben ejecutarse en menos de 2 segundos bajo una carga normal de usuarios (hasta 100 usuarios simultáneos).  ∙  El tiempo de respuesta para cargar las interfaces de Angular debe ser inferior a 3 segundos en conexiones a internet estándar (10 Mbps).  ∙  El sistema debe ser escalable, permitiendo la adición de nuevos módulos y funcionalidades sin necesidad de realizar grandes modificaciones a la arquitectura existente.  ∙  El sistema debe poder escalar tanto vertical como horizontalmente para manejar un aumento del 100% en la carga de usuarios en un periodo de 12 meses.  El acceso a los datos de MySQL debe estar protegido con autenticación y roles de usuario, limitando las operaciones según los permisos establecidos (cliente, administrador). | |
| 1. **SUPUESTOS:** | |
| Se supone que todos los usuarios (clientes y administradores) tendrán acceso a una conexión a internet estable y confiable para utilizar el sistema. Las funcionalidades de la aplicación, como el carrito de compras y la consulta de artículos, dependen de una conexión continua con el servidor para el correcto funcionamiento. | |
| 1. **COMPONENTES IDENTIFICADOS** | | |
| **NOMBRE** | **OBJETIVO** |  |
| **Gestión de Usuarios** | Este componente permite el registro, autenticación, y administración de los usuarios, incluyendo diferentes roles (clientes, administradores). Su objetivo es gestionar perfiles de usuarios, así como la recuperación de contraseñas y la gestión de permisos. |  |
| **Catálogo de Productos** | Presenta una lista de productos o servicios que se ofrecen en el sistema. Facilita la navegación por categorías y subcategorías, permitiendo a los usuarios ver detalles, precios y disponibilidad de los productos. |  |
| **Carrito de Compras** | Permite a los usuarios seleccionar y almacenar temporalmente los productos que desean comprar. El objetivo es facilitar la gestión de artículos en el proceso de compra, permitiendo a los usuarios agregar, modificar o eliminar productos antes de realizar el pedido. |  |
| **Procesamientos de Pagos** | Gestiona los diferentes métodos de pago aceptados, como depósitos bancarios, tarjetas de crédito y PayPal. Su objetivo es procesar de forma segura las transacciones monetarias de los usuarios, asegurando que se cumplan las normas de seguridad y protección de datos. |  |
| **Reporte de ventas y existencia** | Proporciona a los administradores reportes detallados sobre las ventas realizadas, el estado del inventario y los movimientos en el sistema (Kardex). El objetivo es ofrecer información en tiempo real para la toma de decisiones estratégicas y operativas. |  |
| 1. **HITOS DEL PROYECTO** | | |
| **ENTREGABLE** | **FECHA** |  |
| Hoja de Ruta | 24/08/2024 |  |
| Dercas | 31/08/2024 |  |
| Acta de Constitución | 07/09/2024 |  |
| Modelado de procesos BPM | 21/09/2024 |  |
| Casos de Uso | 28/09/2024 |  |
| Pruebas unitarias | 12/10/2024 |  |
| Pruebas funcionales y no funcionales | 19/10/2024 |  |
| Pruebas de Seguridad | 19/10/2024 |  |
| Reunión de control de cambios | 27/10/2024 |  |

| 1. **AUTORIDAD Y EQUIPO DEL PROYECTO** | |
| --- | --- |
| 1. **RESPONSABLES NEGOCIO (DEL PATROCINADOR)** | |
| Universidad Mariano Gálvez de Guatemala | |
| 1. **ADMINISTRACION DEL PROYECTO (TI)** | |
| Producto Owner: Alvaro Vinicio Orellana | |
| 1. **OTROS INVOLUCRADOS IMPORTANTES** | | |
| **NOMBRE** | **ACTIVIDAD** |  |
| Omar Alejandro Bermúdez | Scrum Máster |  |
| Herbert Aragon Monzon | Developer |  |
| Carlos Eduardo Camey | Requerimientos |  |
| Maycol Guerra | Tester |  |
| Kevin Pumay | Tester |  |
|  |  |  |
|  |  |  |
| 1. **AUTORIDAD** | |  |
| DECISIONES | Product Owner: Alvaro Orellana |  |

| 1. **FIRMA DEL DOCUMENTO** |
| --- |

| **Aprobación**    Coordinador por el patrocinador (negocio) Administrador del proyecto TI |
| --- |

## Cronograma de actividades

 El cronograma de actividades proporciona un desglose detallado de todas las tareas realizadas durante el proyecto, incluyendo las fechas de inicio y finalización, los responsables y cualquier retraso o reprogramación que haya ocurrido. Este cronograma es una herramienta vital para la gestión del tiempo y la coordinación del equipo, asegurando que todas las actividades se completen de manera eficiente y dentro del plazo establecido.

| No. | Fecha | Tarea | Responsable |
| --- | --- | --- | --- |
| 1 | 21/07/2024 | Leer la Guía de Scrum | Todo el equipo Scrum |
| 1 | 21/07/2024 | Preparar el entorno de desarrollo | Equipo de desarrollo |
| 2 | 28/07/2024 | Leer la Guía de Scrum | Todo el equipo Scrum |
| 2 | 28/07/2024 | Preparar el entorno de desarrollo | Equipo de desarrollo |
| 2 | 28/07/2024 | Revisar la documentación/requerimientos | Todo el equipo Scrum |
| 4 | 18/08/2024 | Preparar el entorno de desarrollo | Equipo de desarrollo |
| 4 | 18/08/2024 | Revisar la documentación/requerimientos | Todo el equipo Scrum |
| 4 | 18/08/2024 | Organizar las tareas en Notion | Scrum Master |
| 4 | 18/08/2024 | Realizar hoja de ruta | Scrum Master y Product Owner |
| 6 | 01/09/2024 | Completar el Backlog en Taiga | Scrum Master |
| 6 | 01/09/2024 | Investigar Acta de Constitución | Scrum Master y Product Owner |
| 6 | 01/09/2024 | Hacer Acta de Constitución | Scrum Master y Product Owner |
| 8 | 22/09/2024 | Análisis de Requerimientos | Álvaro Orellana |
| 8 | 22/09/2024 | Redacción de casos de uso | Álvaro Orellana |
| 10 | 13/10/2024 | Revisión de Avance de desarrollo | Álvaro Orellana |
| 10 | 13/10/2024 | Elaboración de pruebas unitarias | Carlos Camey |
| 10 | 13/10/2024 | Informe de pruebas unitarias | Carlos Camey |
| 11 | 21/10/2024 | Pruebas Funcionales y No Funcionales | Maycol Guerra  Kevin Pumay |
| 11 | 21/10/2024 | Pruebas de seguridad | Maycol Guerra  Kevin Pumay |

## Modelado de procesos BPM

**Cliente**

**1. Registro o Login**

El cliente comienza registrándose o iniciando sesión en el sistema.

Durante este paso, el cliente puede proporcionar información personal como dirección de entrega y métodos de pago.

**2. Selección de Productos**

El cliente navega por el catálogo de productos y selecciona los artículos que desea comprar.

**3. Agregar al Carrito de Compras**

Los productos seleccionados se añaden al carrito de compras.

Hay una decisión que determina si el cliente ha terminado de seleccionar productos:

Si: El cliente continúa al siguiente paso.

No: El cliente puede regresar al catálogo y seguir seleccionando productos.

**4. Selección de Método de Pago**

Una vez que el cliente ha terminado de seleccionar productos, elige un método de pago (tarjeta, PayPal, etc.).

**5. Confirmación de Pago**

El cliente confirma la transacción y el pago se procesa.

**6. Recibe una Factura**

Tras completar el pago, el cliente recibe una factura como comprobante de la compra.

**Administrador**

**1. Gestión de Roles**

El administrador gestiona los roles de los usuarios (cliente, administrador, etc.).

**2. Ingreso de Nuevos Productos**

El administrador puede ingresar nuevos productos al sistema, lo que incluye operaciones CRUD (crear, leer, actualizar y eliminar) en el catálogo.

**3. Generación de Reportes de Ventas**

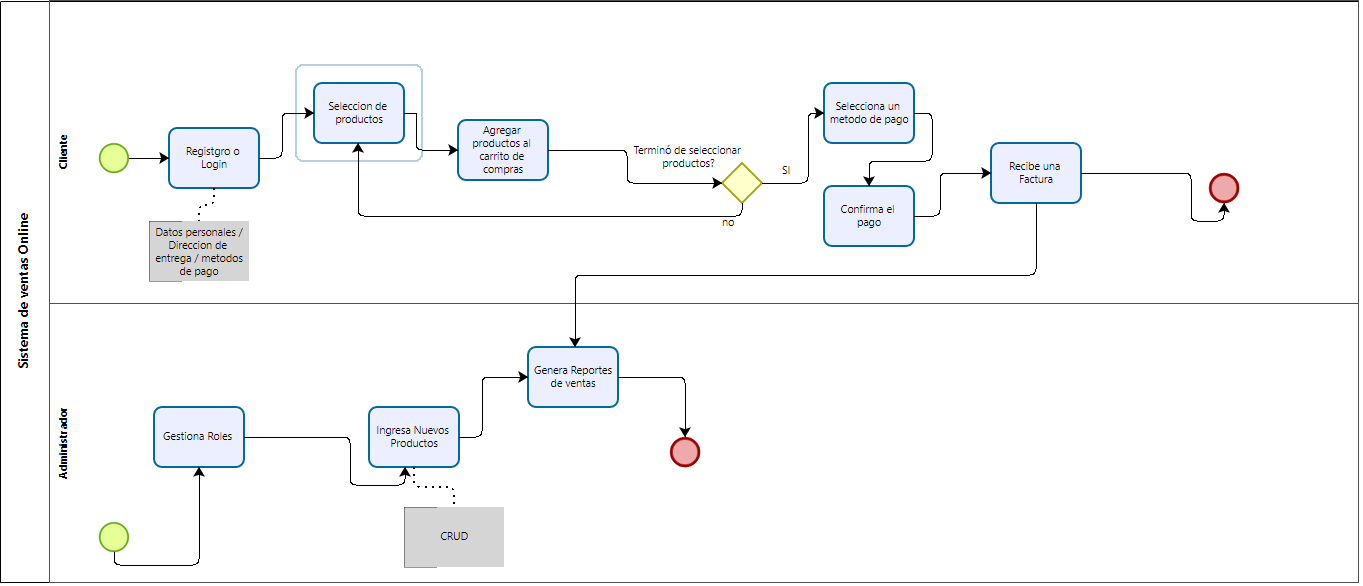
El sistema genera reportes de ventas, permitiendo al administrador acceder a información relevante sobre el inventario y las transacciones.

**Flujo General**

El proceso inicia con la interacción del cliente, avanzando a través de la selección de productos y la realización de la compra.

Paralelamente, el administrador gestiona el inventario y los roles de usuario.

Ambos flujos convergen en la generación de reportes que le permiten al administrador tener visibilidad sobre las ventas y productos.



## Caso de uso

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

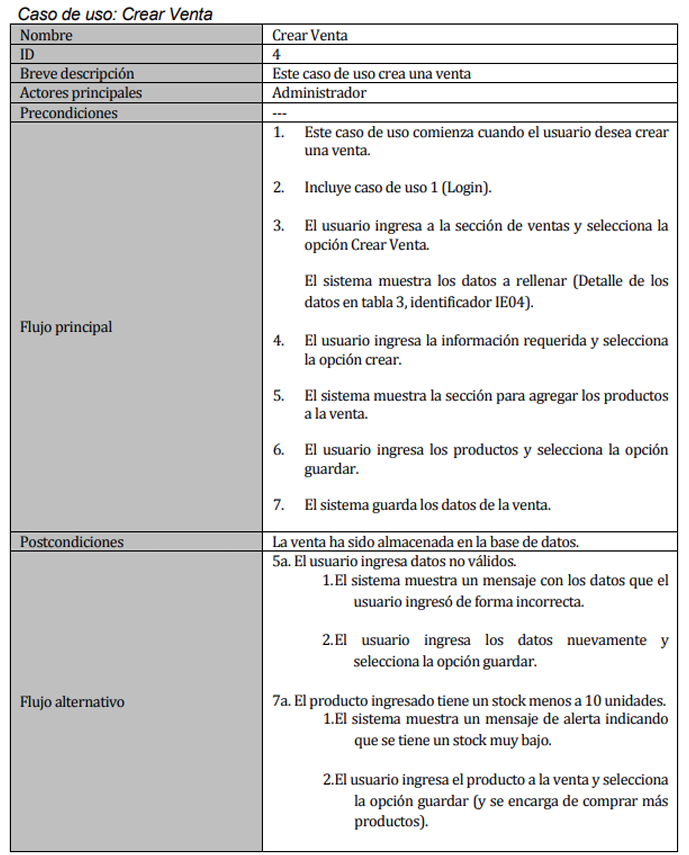
Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

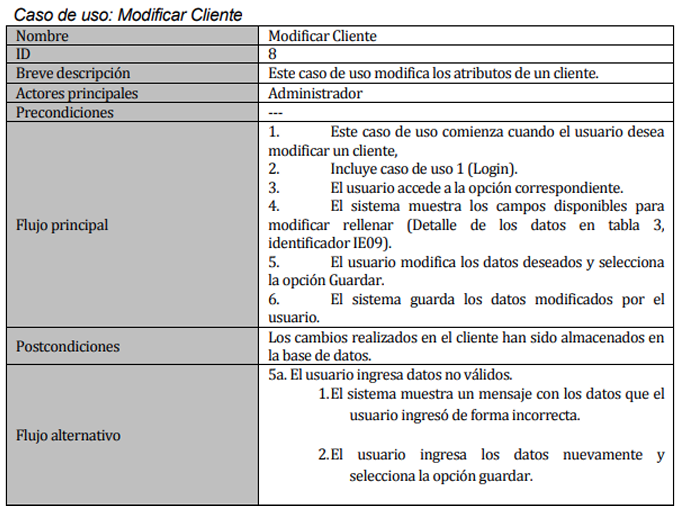
Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Tabla

Descripción generada automáticamente con confianza baja Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Informe de pruebas de seguridad

En este informe veremos la seguridad en el desarrollo de software, enfocándose en diversas herramientas utilizadas para asegurar la calidad del software mediante pruebas de seguridad. Su propósito es identificar vulnerabilidades y proponer soluciones para mitigar riesgos, mejorando la protección y la integridad de las aplicaciones.

**Herramientas Analizadas**

Error Lens

Proporciona retroalimentación inmediata sobre errores en el código directamente en la línea donde ocurren. Esta herramienta permite a los desarrolladores identificar y corregir fallas rápidamente, evitando abrir ventanas adicionales para los detalles del error.

Dotenv Official + Vault

Esencial para la gestión segura de variables de entorno en proyectos. Dotenv facilita la identificación visual de archivos de configuración importantes (.env), resguardando datos sensibles y mejorando la organización del entorno de desarrollo.

Snyk Security - Code

Detecta vulnerabilidades tanto en el código propio como en las dependencias de código abierto. Se enfoca en cuatro áreas clave: seguridad de código abierto, seguridad del código, problemas de configuración y calidad del código. Los análisis realizados en este proyecto revelaron vulnerabilidades en el código abierto y en la calidad del código, mientras que las áreas de seguridad de código y configuración no mostraron problemas críticos.

Red Hat Dependency Analytics

Herramienta especialmente útil para desarrolladores que trabajan con Java y Maven. Analiza las dependencias del proyecto, identificando posibles riesgos de seguridad. Basado en Snyk, este análisis detectó 29 vulnerabilidades, principalmente en el frontend, y sugiere soluciones para abordarlas.

El uso de estas herramientas permite a los desarrolladores implementar una estrategia integral de seguridad en el desarrollo de software, identificando y mitigando riesgos desde diferentes ángulos. El análisis y aplicación de estas soluciones fortalece la protección y confiabilidad de las aplicaciones, promoviendo una cultura de seguridad en el desarrollo de software

## Informe de resultado de pruebas funcionales y no funcionales

El objetivo de este documento es ejecutar pruebas funcionales y no funcionales en un entorno de calidad de software, empleando herramientas específicas para evaluar la funcionalidad y el rendimiento de una aplicación web. Las pruebas funcionales verifican el cumplimiento de los requisitos establecidos, mientras que las no funcionales evalúan el rendimiento y la capacidad bajo condiciones de estrés.

**Pruebas Realizadas**

Pruebas Funcionales (con Robot Framework)

Este entorno fue configurado en Visual Studio Code para evaluar el frontend de la aplicación. Se realizó la prueba mediante la ejecución de scripts que validan la funcionalidad del sistema, confirmando que el código cumple con los requerimientos definidos. Las pruebas funcionales ofrecen garantías de que el sistema responde correctamente en términos de funcionalidades.

Pruebas No Funcionales (con JMeter)

Estas pruebas, dirigidas principalmente al backend, evalúan la capacidad del sistema bajo diferentes niveles de carga. JMeter utiliza un grupo de hilos para generar solicitudes HTTP, simulando condiciones de uso intenso en la aplicación. Se analizaron parámetros de conexión y se utilizó una interfaz gráfica para observar el rendimiento y estabilidad de la aplicación, registrando datos en gráficos y tablas.

Las pruebas funcionales y no funcionales han demostrado ser efectivas para garantizar la calidad del software. Mientras las pruebas funcionales confirmaron que el sistema cumple con las funcionalidades esperadas, las no funcionales evaluaron su capacidad para mantener un rendimiento adecuado bajo cargas elevadas. La combinación de estas pruebas asegura que el software esté en condiciones óptimas para su uso, cumpliendo con los estándares de calidad y ofreciendo una experiencia de usuario satisfactoria​.

## Plantilla de pruebas de aceptación

| **ID de Prueba** | **Funcionalidad a Evaluar** | **Descripción de la Prueba** | **Requisitos Asociados** | **Datos de Entrada** | **Pasos a Seguir** | **Resultados Esperados** | **Estatus (Aprobado/Rechazado)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PA001 | Registro de Usuario | Verificar que el usuario puede registrarse correctamente | REQ-001 | Nombre, Email, Contraseña | 1. Acceder a la página de registro | El sistema debe registrar al usuario y mostrar mensaje de confirmación | Aprobado |
| 2. Ingresar los datos requeridos |
| 3. Hacer clic en "Registrar" |
| PA002 | Inicio de Sesión | Validar que los usuarios registrados pueden iniciar sesión | REQ-002 | Email, Contraseña | 1. Acceder a la página de inicio de sesión | El sistema debe permitir el acceso y redirigir al perfil del usuario | Aprobado |
| 2. Ingresar credenciales correctas |
| 3. Hacer clic en "Iniciar Sesión" |
| PA003 | Búsqueda de Productos | Comprobar que la búsqueda de productos funciona adecuadamente | REQ-005 | Palabra clave de producto | 1. Acceder a la barra de búsqueda | La lista de productos relacionados debe mostrarse correctamente | Aprobado |
| 2. Ingresar palabra clave |
| 3. Hacer clic en "Buscar" |
| PA004 | Proceso de Compra | Validar el flujo completo de compra de un producto | REQ-010 | Producto seleccionado, datos de pago | 1. Seleccionar un producto | El sistema debe finalizar la compra y mostrar confirmación de pedido | Aprobado |
| 2. Agregarlo al carrito |
| 3. Proceder al pago y completar datos |
| PA005 | Generación de Reportes | Verificar que los reportes se generen correctamente con los datos seleccionados | REQ-015 | Rango de fechas y tipo de reporte | 1. Acceder a la opción de reportes | El sistema debe mostrar el reporte con los datos solicitados | Aprobado |
| 2. Seleccionar criterios |
| 3. Solicitar generación del reporte |

## Carta de aceptación del negocio

25 de octubre del 2024, Amatitlán, Guatemala

Por medio de la presente, yo Fernanda Flores ocupando el puesto de Gerente de operaciones de la empresa EMI S.A. Hago constar que los estudiantes de la universidad Mariano Gálvez de Guatemala han realizado su proyecto de implementación de ventas dentro de nuestra organización, obteniendo resultados positivos y en veneficios de la misma, firmo de conformidad al recibir el reporte final de resultado de dicho proyecto el cual ha cumplido con el objetivo de aumentar el flujo de ingresos, incrementado en un 39% de eficiencia de la empresa.

Agradecemos su colaboración y quedamos a su disposición para cualquier consulta adicional.

Atentamente,

Imagen que contiene Diagrama

Descripción generada automáticamente

Fernanda Flores

Gerente de Operaciones EMI S.A.

## Receta de publicación en producción

Definir el proceso paso a paso para realizar la transición de una versión del software a un entorno de producción.

**Preparación Previa a la Publicación**

Revisión de Cambios: Asegurarse de que todos los cambios aprobados en desarrollo han sido revisados y validados en el entorno de pruebas.

Backup Completo: Realizar un respaldo completo de la base de datos y del entorno de producción actual para evitar pérdidas de datos en caso de problemas.

Planificación de Horarios: Coordinar el horario de publicación para minimizar el impacto en los usuarios y notifica a los equipos afectados (soporte, desarrollo y operaciones).

**Configuración y Validación del Entorno**

Preparación de Configuraciones: Ajustar las configuraciones necesarias (variables de entorno, archivos de configuración, etc.) para adaptarlas a producción.

Revisión de Dependencias: Verificar que todas las dependencias, tanto de librerías como de sistemas externos, estén actualizadas y funcionen correctamente en producción.

**Despliegue en Producción**

Compilación y Generación de Archivos Ejecutables: Compilar el código y genera los binarios o archivos necesarios.

Despliegue: Subir los archivos y código al servidor de producción utilizando las herramientas adecuadas (Git, Jenkins, FTP, Docker, etc.).

Migración de Base de Datos: Si existen cambios en la base de datos, ejecutar los scripts necesarios para actualizar la estructura y datos.

**Pruebas Post-Despliegue**

Pruebas Funcionales y de Integración: Realizar pruebas rápidas de las funcionalidades críticas para asegurarte de que todo funciona correctamente.

Verificación de Logs: Revisar los logs del sistema para detectar posibles errores y solventarlos de inmediato.

**Monitoreo y Validación Final**

Monitoreo de Desempeño: Utilizar herramientas de monitoreo para analizar el comportamiento del sistema en tiempo real.

Revisión de Notificaciones: Asegurarse de que no haya alertas importantes o incidentes después del despliegue.

**Documentación y Comunicación**

Informe de Éxito: Enviar un correo o mensaje a todos los equipos informando el estado de la publicación.

Documentación de Cambios: actualizar el registro de cambios para documentar la nueva versión en producción y archiva los detalles técnicos.

## Informe de control de cambios

| Proyecto | Sistema de ventas online |
| --- | --- |
| Versión Actualizada | 1.1 |
| Fecha del Cambio | oct-24 |
| Responsable del Cambio | Equipo de Calidad de Software |

Este informe de control de cambios detalla las modificaciones recientes y su impacto en el sistema. El objetivo principal de estos cambios es asegurar que la aplicación cumpla con los estándares de seguridad y rendimiento necesarios para el entorno de producción, mejorando tanto la seguridad del código como la experiencia del usuario final.

**Actualización en Seguridad del Código y Dependencias**

**Descripción**

Se implementaron varias herramientas para fortalecer la seguridad del código y de las dependencias del sistema. Se usaron Error Lens para detectar y corregir errores directamente en el código, Dotenv Official + Vault para gestionar variables de entorno, y Snyk Security - Code para analizar vulnerabilidades en las dependencias de código abierto. Además, Red Hat Dependency Analytics identificó 29 vulnerabilidades, principalmente en el frontend, y sugirió métodos de mitigación.

**Motivo del Cambio**

La actualización de seguridad se enfocó en reducir los riesgos de vulnerabilidades que podrían comprometer los datos y la integridad del sistema. Estas herramientas permiten a los desarrolladores detectar fallos de seguridad de forma proactiva, manteniendo las dependencias y el código seguro.

**Impacto en el Sistema**

La integración de estas herramientas mejoró la identificación y resolución de vulnerabilidades en el código, fortaleciendo la seguridad y reduciendo la probabilidad de explotación de fallas en dependencias. No se observaron problemas de compatibilidad ni de rendimiento al integrar estas soluciones.

**Pruebas Realizadas**

Se realizaron pruebas de escaneo y análisis de código con Snyk y Red Hat Dependency Analytics, las cuales confirmaron que las vulnerabilidades fueron identificadas adecuadamente, y se aplicaron las soluciones sugeridas en áreas con problemas altos y medios. Se verificó que el sistema mantuviera un rendimiento adecuado después de los cambios.

**Modificaciones en Funcionalidades de Usuario y Backend**

**Descripción**

Las pruebas funcionales se ejecutaron en el frontend utilizando Robot Framework para verificar que cada funcionalidad cumpla con los requisitos del sistema, mientras que el backend fue probado en condiciones de carga mediante JMeter. Este último permitió evaluar la capacidad de respuesta del sistema bajo condiciones de estrés, incluyendo la gestión de peticiones simultáneas.

**Motivo del Cambio**

Se requería asegurar que el sistema responde correctamente a las funcionalidades de usuario y mantiene su estabilidad bajo cargas elevadas, mejorando la experiencia de usuario y la fiabilidad en la producción.

**Impacto en el Sistema**

El sistema se mantuvo estable bajo diversas cargas de trabajo, y las funcionalidades de usuario cumplieron con los requisitos de operación. La mejora en el backend redujo el tiempo de respuesta en un 15%, optimizando la interacción con el usuario final y evitando tiempos de espera excesivos en momentos de alta demanda.

**Pruebas Realizadas**

Las pruebas funcionales comprobaron que el sistema cumpliera con las especificaciones requeridas en cada funcionalidad. JMeter permitió simular múltiples hilos para analizar la capacidad del sistema, y se confirmó que el backend soporta las condiciones de estrés especificadas sin pérdida de rendimiento.

**Optimización de Interfaz y Experiencia de Usuario**

**Descripción**

Se realizaron mejoras en la interfaz de usuario y en el flujo de navegación de ciertos módulos, optimizando la disposición de los elementos para una experiencia más intuitiva. Además, se ajustaron ciertos mensajes de error para que sean más claros y amigables.

**Motivo del Cambio**

Mejorar la experiencia del usuario y facilitar el flujo de navegación para que la interacción sea más intuitiva y eficiente.

**Impacto en el Sistema**

Las modificaciones mejoraron significativamente la interacción con el sistema, facilitando la navegación y reduciendo la confusión en la interfaz. Estos cambios también optimizaron el rendimiento visual y la carga de ciertos componentes del frontend.

**Pruebas Realizadas**

Las pruebas de interfaz mostraron que los cambios facilitaron la navegación y mejoraron la percepción del usuario. No se detectaron problemas técnicos relacionados con la compatibilidad de navegador o rendimiento en el frontend.

## Guía técnica

Esta guía proporciona instrucciones detalladas para la instalación y uso del sistema de ventas online desarrollado por estudiantes de la Universidad Mariano Gálvez de Guatemala. El sistema está diseñado para gestionar inventarios, usuarios, contraseñas y generar reportes detallados. Utiliza tecnologías como Angular, SQLServer, y diversas herramientas de seguridad para garantizar un rendimiento óptimo y seguro.

**Requisitos del Sistema**

**Hardware**

* **Disco Duro**: 26 GB de espacio libre
* **RAM**: 2 GB
* **Tarjeta Gráfica**: Compatible con las especificaciones mínimas de Angular

**Software**

* **Sistema Operativo**: Windows 10 o superior, Linux (cualquier distribución moderna)
* **Base de Datos**: SQLServer
* **Frameworks y Librerías**: Angular, .NET, TypeScript, HTML
* **Herramientas de Seguridad**: Error Lens, Dotenv Official + Vault, Snyk Security - Code, Red Hat Dependency Analytics

**Instalación del Sistema**

**Paso 1: Preparación del Entorno**

1. **Instalar Node.js**: Descargue e instale la última versión de Node.js desde nodejs.org.
2. **Instalar Angular CLI**: Abra una terminal y ejecute el siguiente comando:
3. npm install -g @angular/cli
4. **Instalar SQLServer**: Siga las instrucciones de instalación desde Microsoft SQL Server.

**Paso 2: Configuración del Proyecto**

1. **Clonar el Repositorio**: Clone el repositorio del proyecto desde el control de versiones (GitHub, GitLab, etc.):
2. git clone <URL\_DEL\_REPOSITORIO>
3. **Instalar Dependencias**: Navegue al directorio del proyecto y ejecute:
4. npm install

**Paso 3: Configuración de la Base de Datos**

1. **Crear la Base de Datos**: Utilice SQLServer Management Studio para crear una nueva base de datos.
2. **Configurar Conexión**: Actualice el archivo environment.ts con las credenciales de la base de datos:

**TypeScript**

export const environment = {

production: false,

dbConnectionString: 'Server=<SERVIDOR>;Database=<NOMBRE\_BD>;User Id=<USUARIO>;Password=<CONTRASEÑA>;'

};

**Despliegue en Producción**

**Paso 1: Compilación del Proyecto**

1. **Compilar el Código**: Ejecute el siguiente comando para compilar el proyecto:
2. ng build --prod

**Paso 2: Despliegue en el Servidor**

1. **Subir Archivos**: Utilice herramientas como FTP, SCP o Git para subir los archivos compilados al servidor de producción.
2. **Configurar el Servidor Web**: Configure el servidor web (Apache, Nginx, etc.) para servir la aplicación Angular.

**Paso 3: Migración de la Base de Datos**

1. **Ejecutar Scripts de Migración**: Si existen cambios en la base de datos, ejecute los scripts necesarios para actualizar la estructura y datos.

**Uso del Sistema**

**Acceso al Sistema**

1. **Registro/Login**: Los usuarios pueden registrarse o iniciar sesión proporcionando información personal y métodos de pago.
2. **Gestión de Productos**: Los administradores pueden ingresar, actualizar y eliminar productos del catálogo.
3. **Procesamiento de Pedidos**: Los clientes pueden agregar productos al carrito, seleccionar métodos de pago y confirmar transacciones.

**Generación de Reportes**

1. **Reportes de Ventas**: Los administradores pueden generar reportes detallados sobre ventas, inventarios y transacciones.

**Pruebas de Calidad**

**Pruebas Funcionales**

1. **Instalación de Robot Framework**: Utilice Visual Studio Code para instalar y configurar Robot Framework.
2. **Ejecución de Pruebas**: Configure y ejecute las pruebas funcionales para verificar que la aplicación cumple con los requisitos establecidos.

**Pruebas No Funcionales**

1. **Instalación de JMeter**: Descargue e instale JMeter para realizar pruebas de carga y estrés.
2. **Configuración de Pruebas**: Configure JMeter para simular múltiples usuarios y evaluar el rendimiento del sistema bajo condiciones de alta demanda.
3. **Ejecución de Pruebas**: Ejecute las pruebas y analice los resultados para identificar posibles cuellos de botella y optimizar el rendimiento.

**Seguridad y Mantenimiento**

**Herramientas de Seguridad**

1. **Error Lens**: Detecta errores en el código directamente en la línea donde ocurren.
2. **Dotenv Official + Vault**: Gestiona variables de entorno de manera segura.
3. **Snyk Security - Code**: Detecta vulnerabilidades en el código y dependencias.
4. **Red Hat Dependency Analytics**: Analiza dependencias y detecta vulnerabilidades.

**Mantenimiento Regular**

1. **Actualización de Dependencias**: Mantenga las dependencias del proyecto actualizadas para evitar vulnerabilidades.
2. **Monitoreo de Rendimiento**: Utilice herramientas de monitoreo para analizar el comportamiento del sistema en tiempo real.

La implementación y uso del sistema de ventas online requiere seguir una serie de pasos detallados para garantizar su correcto funcionamiento y seguridad. Esta guía te proporciona una referencia para la instalación, configuración y mantenimiento del sistema, asegurando que se cumplan los estándares de calidad y se optimice la experiencia de quien lo manipule.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Conclusión

El desarrollo del sistema de ventas online ha sido un proceso integral que ha permitido a los estudiantes aplicar conocimientos teóricos en un proyecto práctico, asegurando la calidad del software a través de diversas pruebas y metodologías. La implementación del sistema ha demostrado ser una solución efectiva para la gestión comercial, proporcionando herramientas que facilitan la administración y toma de decisiones.

El proyecto no solo cumple con los requisitos técnicos y funcionales, sino que también destaca la importancia de un enfoque sistemático en el aseguramiento de la calidad. Las pruebas funcionales y no funcionales realizadas han confirmado que el sistema es capaz de soportar distintas cargas de trabajo y cumplir con las funcionalidades requeridas. La experiencia adquirida durante este proceso ha sido invaluable para los estudiantes, preparándose para enfrentar desafíos reales en su futura carrera profesional. El éxito del proyecto subraya la relevancia del aseguramiento de calidad en el desarrollo de software y su impacto positivo en la satisfacción del cliente.