|  |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: ByteMart* |
| **30/07/2024** |



**Integrantes:**

Karla Martinez

Michael Marín

Cristian Ojeda

Vicente Laguna

**Profesor:**

Julio Tapia

**Contenido**

[**Ficha del documento 3**](#_heading=h.gjdgxs)

[**Integrantes del Equipo 3**](#_heading=h.30j0zll)

[**1. Introducción 5**](#_heading=h.3znysh7)

[1.1.](#_heading=h.2et92p0) Propósito 5[\_heading=h.2et92p0](#_heading=h.2et92p0)

[1.2.](#_heading=h.tyjcwt) Ámbito del Sistema 5[\_heading=h.tyjcwt](#_heading=h.tyjcwt)

[1.3.](#_heading=h.3dy6vkm) Definiciones, Acrónimos y Abreviaturas 5[\_heading=h.3dy6vkm](#_heading=h.3dy6vkm)

[1.4.](#_heading=h.1t3h5sf) Referencias 5[\_heading=h.1t3h5sf](#_heading=h.1t3h5sf)

[1.5.](#_heading=h.4d34og8) Visión General del Documento 5[\_heading=h.4d34og8](#_heading=h.4d34og8)

[**2.**](#_heading=h.2s8eyo1) **Descripción General 6**[\_heading=h.2s8eyo1](#_heading=h.2s8eyo1)

[2.1.](#_heading=h.17dp8vu) Perspectiva del Producto 6[\_heading=h.17dp8vu](#_heading=h.17dp8vu)

[2.2.](#_heading=h.3rdcrjn) Funciones del Producto 6[\_heading=h.3rdcrjn](#_heading=h.3rdcrjn)

[2.3.](#_heading=h.26in1rg) Características de los Usuarios 6[\_heading=h.26in1rg](#_heading=h.26in1rg)

[2.4.](#_heading=h.lnxbz9) Restricciones 6[\_heading=h.lnxbz9](#_heading=h.lnxbz9)

[2.5.](#_heading=h.35nkun2) Suposiciones y Dependencias 7[\_heading=h.35nkun2](#_heading=h.35nkun2)

[2.6.](#_heading=h.1ksv4uv) Requisitos Futuros 7[\_heading=h.1ksv4uv](#_heading=h.1ksv4uv)

[**3.**](#_heading=h.44sinio) **Requisitos Específicos 8**[\_heading=h.44sinio](#_heading=h.44sinio)

[3.1](#_heading=h.2jxsxqh) Requisitos comunes de las interfaces 9[\_heading=h.2jxsxqh](#_heading=h.2jxsxqh)

[*3.1.1*](#_heading=h.z337ya) *Interfaces de usuario 9*[\_heading=h.z337ya](#_heading=h.z337ya)

[*3.1.2*](#_heading=h.3j2qqm3) *Interfaces de hardware 9*[\_heading=h.3j2qqm3](#_heading=h.3j2qqm3)

[*3.1.3*](#_heading=h.1y810tw) *Interfaces de software 9*[\_heading=h.1y810tw](#_heading=h.1y810tw)

[*3.1.4*](#_heading=h.4i7ojhp) *Interfaces de comunicación 10*[\_heading=h.4i7ojhp](#_heading=h.4i7ojhp)

[3.2](#_heading=h.2xcytpi) Requisitos funcionales 10[\_heading=h.2xcytpi](#_heading=h.2xcytpi)

[3.3](#_heading=h.1ci93xb) Requisitos no funcionales 10[\_heading=h.1ci93xb](#_heading=h.1ci93xb)

[*3.3.1*](#_heading=h.3whwml4) *Requisitos de rendimiento 10*[\_heading=h.3whwml4](#_heading=h.3whwml4)

[*3.3.2*](#_heading=h.2bn6wsx) *Seguridad 10*[\_heading=h.2bn6wsx](#_heading=h.2bn6wsx)

[*3.3.3*](#_heading=h.qsh70q) *Fiabilidad 11*[\_heading=h.qsh70q](#_heading=h.qsh70q)

[*3.3.4*](#_heading=h.3as4poj) *Disponibilidad 11*[\_heading=h.3as4poj](#_heading=h.3as4poj)

[*3.3.5*](#_heading=h.1pxezwc) *Mantenibilidad 11*[\_heading=h.1pxezwc](#_heading=h.1pxezwc)

[*3.3.6*](#_heading=h.49x2ik5) *Portabilidad 11*[\_heading=h.49x2ik5](#_heading=h.49x2ik5)

[3.4](#_heading=h.2p2csry) Otros Requisitos 11[\_heading=h.2p2csry](#_heading=h.2p2csry)

[**4. Propuesta de Planificación 12**](#_heading=h.147n2zr)

[4.1 Descripción general acerca de la Planificación 12](#_heading=h.3o7alnk)

[*4.1.2 Definición del Equipo de Trabajo 12*](#_heading=h.23ckvvd)

[*4.1.3 Definición de Actividades principales del Proyecto 12*](#_heading=h.ihv636)

[*4.1.4 Diagrama EDT 12*](#_heading=h.32hioqz)

[*4.1.5 Carta Gantt 12*](#_heading=h.1hmsyys)

[*4.1.6 Matriz RAM o RACI 12*](#_heading=h.41mghml)

[4.3 Matriz de Riesgos 12](#_heading=h.2grqrue)

[5. Anexos 12](#_heading=h.vx1227)

[*5.1 Acta de Proyecto 12*](#_heading=h.3fwokq0)

[*5.2 Matriz Especificación de Requerimientos 13*](#_heading=h.1v1yuxt)

[*5.3 Diagrama de Casos de Uso General 13*](#_heading=h.4f1mdlm)

[*5.4 Planilla Casos de Uso 13*](#_heading=h.2u6wntf)

[*5.5 Prototipado de Software 13*](#_heading=h.19c6y18)

[*5.6 Modelos de Proceso de negocio 13*](#_heading=h.3tbugp1)

[*5.7 Diagrama de actividades de su solución 13*](#_heading=h.28h4qwu)

[*5.8 Modelo Base de Datos 13*](#_heading=h.nmf14n)

[*5.9 Diccionario de datos 13*](#_heading=h.37m2jsg)

[*5.10 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 13*](#_heading=h.1mrcu09)

**Ficha del documento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| *30/07/2024* | *1.0* | *Vicente Laguna* | *Completar informe.* |
| *11/09/2024* | *2.0* | *Karla Martinez* | *Completar informe.* |
| *14/09/2024* | *3.0* | *Cristian Ojeda* | *Completar informe.* |
| *10/10/2024* | *4.0* | *Michael Marín* | *Completar informe.* |

**Integrantes del Equipo**

|  |  |
| --- | --- |
| **Nombre Integrante** | **Rol definido en el Equipo de Proyecto** |
| Karla Martínez | Analista de negocio - Equipo desarrollo - Control de calidad - Responsable de comunicaciones |
| Michael Marín | Analista de negocio - Equipo desarrollo - Control de calidad |
| Cristian Ojeda | Analista de negocio - Equipo desarrollo - Control de calidad |
| Vicente Laguna | Project lead - Analista de negocio - Equipo desarrollo - Control de calidad |

**1. Introducción**

Este documento está orientado a la definición detallada de los requisitos de los distintos sistemas, se definen los propósitos establecidos con el cliente, como es el funcionamiento que se espera lograr con la implementación, el lenguaje técnico importante de manejar para el correcto entendimiento de los conceptos aplicados, que visiones se presentan sobre el objetivo de cada proceso, las limitaciones adjuntas al proyecto, cuál es el ambiente adecuado para el alojamiento de los sistemas, herramientas a utilizar, tipos de lenguaje de código, su escalabilidad. Todo lo anterior para culminar con la interpretación de datos disponibles y resultados generados.

**1.1. Propósito**

El propósito de este documento es tener una guía detallada sobre cómo está diseñada la estructura de cada uno de los softwares solicitados por el cliente y mostrar cómo conseguiremos abarcar cada uno de los puntos detallados en el desglose siguiente.

La audiencia objetiva de este informe está compuesta por el equipo de desarrollo, quien está encargado de generar cada uno de los elementos necesarios para completar los sistemas, esto incluye, los distintos scripts en los distintos lenguajes de programación, las bases de datos y el diseño de las interfaces.

**1.2. Ámbito del Sistema**

El sistema llevará por nombre: ByteMart.

El sistema desarrollado se trata de un e-ecommerce de venta de productos tecnológicos, por lo cual, el sistema está diseñado principalmente para automatizar procesos mediante web scraping, además de realizar extracción e ingesta de datos para permitir la comparativa de precios con las competencias de mercado, además integrar distintas APIS de sistemas de pagos, productos recomendados, consulta de categorías y notificación de stock bajo para el manejo de datos, análisis y publicación de estos. También, el sistema estará diseñado para el usuario, con una interfaz intuitiva y amigable que permite una gran experiencia de usuario.

El sistema no hace principalmente, especificaciones que no estén detalladas en la planilla de requerimientos.

Con lo especificado en este proyecto, se espera agilizar y por sobre todo automatizar procesos, para satisfacer las necesidades de los clientes y lograr una herramienta digitalizada acorde a la demanda de estos nuevos tiempos tecnológicos.

**1.3. Definiciones, Acrónimos y Abreviaturas**

**Python Flask:** Flask es un framework ligero para crear aplicaciones web en Python. Es conocido por su simplicidad y flexibilidad, permitiendo a los desarrolladores crear aplicaciones web y Apis rápidamente. Flask sigue una arquitectura modular que permite usar solo los componentes necesarios, lo que lo hace ideal para proyectos pequeños a medianos.

**MySQL:** MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto. Utiliza SQL (Structured Query Language) como lenguaje para gestionar y consultar bases de datos. Es ampliamente utilizado en aplicaciones web y es conocido por su rendimiento, estabilidad y facilidad de uso. MySQL permite almacenar, modificar y gestionar grandes cantidades de datos de manera estructurada.

**GitHub:** GitHub es una plataforma de desarrollo colaborativo que permite almacenar código fuente utilizando Git como sistema de control de versiones. Ofrece herramientas para el control de versiones, colaboración entre equipos, revisión de código, seguimiento de errores y más. Es una herramienta clave en el desarrollo de software, ya que permite a los desarrolladores trabajar en equipo de manera eficiente y gestionar sus proyectos de forma organizada.

**Apis:** Una API es un conjunto de reglas y definiciones que permite que diferentes aplicaciones o sistemas se comuniquen entre sí. A través de las Apis, los desarrolladores pueden acceder a funcionalidades o datos de otras aplicaciones o servicios sin necesidad de conocer el código interno.

**HTTP:** HTTP es el protocolo base de la web, utilizado para la transmisión de datos entre un cliente (como un navegador) y un servidor web. Es un protocolo basado en solicitudes y respuestas: el cliente envía una solicitud HTTP al servidor, y este responde con los datos solicitados (como una página web, un archivo, etc.).

**Web Scraping:** El web scraping es la técnica de extraer información de sitios web de manera automatizada. Utilizando scripts o herramientas especializadas, se accede a páginas web y se recopilan datos estructurados (como texto, imágenes o tablas). Aunque puede ser útil para análisis de datos, vigilancia de precios, entre otros, es importante respetar las políticas de los sitios web y los aspectos legales del scraping.

**RPA:** Es la tecnología que permite automatizar tareas repetitivas mediante la creación de robots de software o scripts que imitan las acciones humanas en interfaces digitales. Se utiliza para realizar tareas como el procesamiento de datos, la generación de reportes, la extracción de información y otras actividades que consumen tiempo. Los robots de RPA pueden interactuar con aplicaciones y sistemas de la misma manera que lo haría una persona, mejorando la eficiencia y reduciendo el error humano.

**1.4. Referencias**

* Documento APT.
* Matriz de Requerimientos.
* Matriz de Riesgos.
* Roles y Responsabilidades
* Matriz EDT.
* Carta Gantt.

**1.5. Visión General del Documento**

El documento se divide exactamente en 5 puntos:

1. Explica una visión detallada del sistema ByteMart. El propósito es facilitar el desarrollo del sistema, considerando que la estructura y funcionamiento establecidos cumplan con la expectativa del cliente.
2. Describir las funciones generales del sistema, es decir, los tipos de usuarios, necesidades y futuras mejoras del producto.
3. Establece los requisitos específicos relacionados con el sistema, este debe ser una guía de cómo será el producto final.
4. Planificación del proyecto, cuenta con el cronograma de actividades e hitos del proyecto, permitiendo que el desarrollo del sistema sea organizado y eficiente.
5. Documentos anexados que sirven de utilidad para respaldar y complementar el documento presente.

**2. Descripción General**

Debido a la creciente demanda de herramientas digitales para el manejo de información crucial para la toma de decisiones. Se optimizan los procesos operativos, para reducir errores al automatizar tareas repetitivas, con el fin de mejorar los procesos mediante esta herramienta digitalizada.

El objetivo del proyecto APT es ofrecer una solución integral, esta solución se centra en el uso de RPA (Automatización Robótica de Proceso), se debe incluir una plataforma web creada en Python, utilizando tecnologías como Flask y como gestión de base de datos MySQL, plataforma de escritorio desarrollada en Java y NetBeans, Integración de Apis para sistema de pago y manejo de información, análisis y publicación de datos mediante Power BI.

**2.1. Perspectiva del Producto**

El proyecto ByteMart se dirige a organizaciones que desean modernizar los procesos, con el fin de reducir costos operacionales a través del uso de herramientas digitalizadas. Esta solución es flexible y altamente escalable, permitiendo la implementación en diversos mercados, sin importar el tamaño de la empresa o rubro, aportando un manejo amplio de la información para una toma de decisiones acertada en el tiempo.

**2.2. Funciones del Producto**

Una de las principales funciones del proyecto ByteMart es la automatización para realizar un web scraping, se realiza una extracción e ingesta de datos con el objetivo de estudiar la competencia y analizar el estado del mercado, esto optimiza la toma de decisiones. Además de integrar API para el sistema de pago y desarrollar Apis con distintos propósitos, para la gestión de inventario, productos recomendados, notificación de stock bajo y consultar categorías. Nuestra plataforma web ofrece una interfaz amigable para el usuario, que permita la visualización de productos y realizar compras. Por otro lado, la plataforma de escritorio permite asegurar continuidad del trabajo sin necesidad de tener conectividad a internet. Finalmente, permite gestionar información empresarial de forma efectiva, estructurando los datos para facilitar el análisis.

**2.3. Características de los Usuarios**

* Analista de negocio.
* Equipo de desarrollo.
* Responsable de comunicaciones.
* Project lead.
* Administrador.
* Jefe de tienda.
* Vendedor.
* Cliente.

**2.4. Restricciones**

Dentro de ByteMart existen distintas políticas sobre la privacidad y protección de sus datos, donde hay una fuente de información que relaciona a clientes, usuarios y con qué fines se utiliza.

Esta misma política exige que los usuarios den su consentimiento explícito para el uso de datos personales. También, el cumplimiento de normativas para la protección de la información debe ser considerado al momento de concebir los accesos.

Existen distintos tipos de roles que pueden tener accesos a distintos tipos de datos, es importante definir qué tipo de rol tendrán los desarrolladores de los distintos aplicativos diseñados para la organización.

Para el correcto funcionamiento de los aplicativos, es necesario contar con un ambiente. Este mismo, debe contar con un servidor local, sus especificaciones deben ser: Sistema operativo Windows 10 o posterior, este debe ser un equipo con 16 gb mínimos de RAM, un procesador Intel i7 de quinta generación o posterior, conectividad a internet para la configuración inicial y levantamiento de herramientas.

Se utiliza dentro del ambiente Python con las librerías de Flask para el levantamiento de la plataforma web ecommerce, la contiene los mantenedores de los productos y usuarios, los cuales pueden ser: administrador, jefe de tienda, vendedor, cliente o comprador.

También, se hace uso de Apache NetBeans para el desarrollo del aplicativo de escritorio el cual cuenta con los mismos roles del aplicativo web, pero sin considerar el cliente o comprador, este sistema ayuda y trabaja de manera simultánea con el sistema web, estos se apoyan de distintas APIS, de esta manera distribuir los flujos de trabajo.

**2.5. Suposiciones y Dependencias**

* Se cuenta con infraestructura básica, que permite el rendimiento y disponibilidad para la estación de trabajo esta permite optimizar el desarrollo y pruebas de los sistemas.
* Utilizar herramientas de código abierto, que nos permita desarrollar sin depender de un presupuesto adicional.
* Se utiliza la plataforma GitHub para seguir un formato de trabajo en equipo más efectivo, con un control de versiones y acceso libre al proyecto.

**2.6. Requisitos Futuros**

Proyecto ByteMart expande su capacidad de automatización, permitiendo realizar flujo con más complejidad, cubriendo diversas áreas dentro de la empresa.

**3. Requisitos Específicos**

Para suplir las necesidades de ByteMart se realiza la implementación de distintos sistemas, los cuales cuentan con distintos roles y usuarios.

Estos generan un registro de usuario, iniciar sesión, realizar administración de sus cuentas, realizar compras con medios de pago, realizar administración de productos. Todo esto considerando la personalización y experiencia del cliente.

**3.1 Requisitos comunes de las interfaces**

**3.1.1 Interfaces de usuario**

Para el sistema que se ha compuesto, se realiza una interfaz gráfica la cual permite realizar distintas interacciones de usuarios finales y a los administradores. Esta misma es intuitiva, atractiva visualmente y que todas sus funcionalidades cuentan con un fácil acceso.

Como entradas se consideran las siguientes:

1. Formulario de registro de usuarios: Los usuarios completan distintos campos como correo electrónico, nombre de usuario, contraseña y dirección.
2. Formulario de Login: Los usuarios ingresan mediante los campos de nombre de usuario y contraseña.
3. Formulario para la recuperación de la contraseña: El usuario registrado ingresa el correo electrónico, para hacer el envío de código de recuperación.
4. Búsqueda de productos: El sistema debe operar con un campo de texto el cual permite ingresar distintos términos de búsqueda.
5. Formulario para agregar productos al inventario: Los usuarios completan los campos requeridos incluyendo fotografías o imágenes de los productos.
6. Eliminar productos del inventario: Los usuarios autorizados pueden seleccionar los productos a eliminar del inventario.
7. Formulario para ingresar categorías de productos: Los usuarios autorizados ingresan nombre de la categoría a ingresar.
8. Eliminar órdenes de compra: Los usuarios autorizados seleccionan la orden de compra a eliminar.

Como salidas se consideran las siguientes:

1. Confirmación de registro de usuarios: El sistema valida los datos entregados por el usuario, genera un mensaje si la cuenta se ha registrado exitosamente
2. Respuesta del Login: El sistema al validar los datos entregados por el usuario muestra un mensaje cuando las credenciales son correctas o incorrectas.
3. Respuesta de recuperación de contraseña: El sistema solicita al usuario que ingrese el código que se envía por correo electrónico, para solicitar que el usuario ingrese una contraseña nueva.
4. Resultado búsqueda de productos: El sistema muestra mediante un listado, estos resultados coinciden con las palabras claves que se ingresan en el campo de búsqueda.
5. Resultados del ingreso de producto: El sistema valida los datos e imagen ingresada por el usuario.
6. Confirmación para eliminar el producto: El sistema muestra un mensaje confirmando que el producto fue eliminado.
7. Resultado del ingreso de categoría de productos: El sistema muestra un mensaje cuando la categoría se registró exitosamente o un mensaje de error si ya existe la categoría en la base de datos.
8. Confirmación para eliminar orden de compra: El sistema debe mostrar un mensaje de que la orden de compra se eliminó correctamente.

**3.1.2 Interfaces de hardware**

Se señalan los requisitos de las características lógicas de las diferentes interfaces del producto y cada componente asociado, para entregar la trazabilidad entre los distintos sistemas.

El sistema web utiliza la tecnología de Flask levantada en Python, para la cual se utiliza un entorno local para poder realizar conexión con el aplicativo, con la tecnología de Flask podremos realizar la creación directa de clases para interactuar con la base de datos, como también manejar las distintas peticiones que pueden existir en el portal, ya sean formularios de creación de distintos tipos de datos, como pueden ser usuarios, productos, categorías, entre otros.

Para el manejo y almacenamiento de los datos generados, se utiliza el motor de base de datos de MySQL, dónde hacemos uso de distintas herramientas como secuencias para la auto creación de ID en algunas clases, creación de tablas y la conexión entre ellas para entrelazar el modelo de negocio dentro de sus procesos.

Para el sistema de escritorio, al igual que el web, se hace uso de un entorno local, levantado en NetBeans y construido en lenguaje java, en el cual se crean distintas clases para la interacción con la base de datos que en este caso compartirá con el sistema web en MySQL, este sistema contará con las funcionalidades más importantes para la administración del modelo de negocio.

Para el resguardo y seguridad de los datos se hará uso de una base de datos espejo la cual periódicamente se va cargando con la información generada por el portal, el proceso de carga de datos de respaldo será por lotes después de un tiempo determinado por el cliente, de esta manera minimizamos los riesgos ante incidentes perjudiciales para la organización y/o el cliente.

Las interacciones con las distintas Apis, como las de pago, gestión de inventario, productos recomendados y consultar categorías, se usa el formato JSON para el intercambio de información, de esta manera administramos efectivamente el detalle de cada concepto relevante para el cliente.

**3.1.3 Interfaces de software**

Utilizamos Flask, una herramienta basada en Python para el aplicativo web en conjunto con el motor de base de datos MySQL para la gestión de la data generada. Por otro lado, utilizamos Java para el desarrollo del aplicativo de escritorio, el cual cuenta con el mismo motor de base de datos.

Para el manejo del contenido y el formato, utilizamos el formato json para el intercambio de datos entre los sistemas y las APIS, también, el uso de consultas básicas en el portal ecommerce para la creación, eliminación, actualización y listado de datos. Finalmente, el sistema de escritorio a través del driver jdbc realizará la conexión a la base de datos.

**3.1.4 Interfaces de comunicación**

El sistema diseñado para ByteMart se comunica e interactúa a través de diversos componentes, tanto web como escritorio, utilizando un entorno local para una gestión de inventarios, productos, simulaciones de pago, e intercambio de datos compartidos entre los sistemas.

El protocolo definido para la comunicación, en este caso es local, HTTP para el aplicativo web en el localhost 8080, para el cliente del sistema de escritorio, este funciona directamente con la gestión y administración de data con una base de datos.

**3.2 Requisitos funcionales**

**Definición de requerimientos específicos:**

3.2.1 Login de usuario: El sistema permite a los usuarios iniciar sesión mediante credenciales válidas.

3.2.2 Registro de usuarios: El sistema permite crear nuevas cuentas a usuarios con su información.

3.2.3 Eliminar usuario: El administrador puede eliminar usuarios del sistema

3.2.4 Editar usuario: El administrador y los usuarios pueden modificar y visualizar la información de sus cuentas.

3.2.5 Recuperación de contraseña: El sistema permite a los usuarios restablecer la contraseña de sus cuentas mediante un formulario de recuperación.

**Requerimientos para la gestión de inventario y carrito de compras:**

3.2.6 Agregar productos al carrito de compras: Los usuarios y clientes pueden añadir productos al carrito de compras.

3.2.7 Eliminación de productos del carrito de compras: Los usuarios y clientes pueden eliminar productos agregados al carrito de compras.

3.2.8 Añadir elementos al inventario: El administrador, los jefes de tienda y los vendedores pueden añadir elementos al inventario.

3.2.9 Agregar productos al inventario: Los usuarios autorizados pueden ingresar productos dentro del inventario.

3.2.10 Editar productos en el inventario: Los usuarios autorizados pueden editar los productos que se encuentran en el inventario.

3.2.11 Eliminar productos del inventario: Los usuarios autorizados podrán editar los productos que se encuentran en el inventario.

3.2.12 Eliminación de elementos añadidos al inventario: El administrador, los jefes de tienda y los vendedores pueden eliminar elementos del inventario.

3.2.13 Filtrado de elementos del inventario: El administrador, los jefes de tienda y los vendedores pueden filtrar los elementos del inventario.

3.2.14 Visualización de elementos del inventario: El administrador, los jefes de tienda y los vendedores pueden visualizar los elementos del inventario.

3.2.15 Visualización de stock de los artículos: Los usuarios podrán visualizar los stocks de los artículos que se encuentran en el inventario.

3.2.16 Entregar precio de productos: Los usuarios podrán visualizar los stocks de los artículos que se encuentran en el inventario.

**Requerimientos para la gestión de categorías en el portal web:**

3.2.17 Agregar categorías de productos: Los usuarios autorizados pueden agregar categorías al sistema.

3.2.18 Editar categorías de productos: Los usuarios autorizados pueden editar categorías en el sistema.

3.2.19 Eliminar categoría de productos: Los usuarios autorizados, pueden eliminar categorías en el sistema.

3.2.20 Visualización de categorías por productos: Los usuarios pueden visualizar las categorías que se encuentran en el inventario.

3.2.21 Buscador de productos por categoría: Los usuarios pueden buscar productos según su categoría por palabras claves.

**Requerimientos para la gestión de facturas y ordenes de compras:**

3.2.22 Los usuarios autorizados pueden eliminar órdenes de compra registradas en el sistema

3.2.23 Visualizar facturas de venta: En el sistema permite a los usuarios acceder y visualizar las facturas de ventas correspondientes.

**Requerimientos para la gestión de usuarios en el portal de escritorio:**

3.2.24 Login de usuario: El sistema permite a los usuarios iniciar sesión mediante credenciales válidas.

3.2.25 Registro de usuarios: El sistema permite crear nuevas cuentas a usuarios con su información.

3.2.26 Editar usuario: El administrador y los usuarios pueden modificar y visualizar la información de sus cuentas.

3.2.27 Eliminar usuario: El administrador puede eliminar usuarios del sistema

**Requerimientos para la gestión de inventario en el portal de escritorio:**

3.2.28 Agregar productos al inventario: Los usuarios autorizados pueden ingresar productos dentro del inventario.

3.2.29 Editar productos en el inventario: Los usuarios autorizados pueden editar los productos que se encuentren en el inventario.

3.2.30 Eliminar productos del inventario: Los usuarios autorizados pueden editar los productos que se encuentren en el inventario.

3.2.31 Visualización de artículos: Usuarios pueden visualizar los artículos que se encuentran en el inventario.

3.2.32 Visualización de stock de productos: Usuarios pueden visualizar el stock de los productos registrados en el inventario.

3.2.33 Buscador de productos: Los usuarios pueden buscar productos por palabras claves.

**Requerimientos para la gestión de categorías en el portal de escritorio:**

3.2.34 Agregar categorías de productos: Los usuarios autorizados pueden agregar categorías al sistema.

3.2.35 Editar categorías de productos: Los usuarios autorizados pueden editar categorías en el sistema.

3.2.36 Eliminar categoría de productos: Los usuarios autorizados, pueden eliminar categorías en el sistema.

3.2.37 Buscador de productos por categoría: Los usuarios pueden buscar productos según su categoría por palabras claves.

**Requerimientos para la gestión de facturas y ordenes de compras:**

3.2.38 Eliminar órdenes de compra: Los usuarios autorizados pueden eliminar órdenes de compra registradas en el sistema

3.2.39 Visualizar facturas de venta: En el sistema permite a los usuarios acceder y visualizar las facturas de ventas correspondientes.

**3.3 Requisitos no funcionales**

**3.3.1 Requisitos de rendimiento**

El tiempo de respuesta que tiene el sistema al momento de realizar las transacciones es de un 85% de las transacciones en menos de 5 segundos.

El sistema de ByteMart soporta un máximo de 200 usuarios conectados simultáneamente, también se considera que el tiempo de respuesta de las diversas operaciones no excede los 5 segundos.

**Requerimientos para la integración de Apis en el portal web:**

3.2.40 Integración de API de pago: La plataforma web está conectada con la API de pago desarrollada, permitiendo que el proceso de pago sea efectivo, el tiempo de espera para validar el proceso de pago es de al menos 5 segundos para el 85% de las transacciones.

3.2.41 Integración de API de productos recomendados: La plataforma web está conectada con la API para productos recomendados, donde se visualizan sólo artículos recomendados por temporada.

3.2.42 Integración de API DPA: La plataforma web está conectada con una API para extraer información sobre provincias, comunas y regiones de chile, con el fin de tener una mayor cobertura con respecto al envió de nuestros productos.

3.2.43 Integración de API para Notificación de stock bajo: La plataforma web está conectada con una API para la notificación de stock bajo, la cual mediante correo electrónico envía una notificación que señala el producto y el total de stock disponible en inventario.

**Requerimientos para las Interfaces de usuarios:**

3.2.43 Sistema debe ser responsivo: La interfaz de la plataforma web se adapta automáticamente a las distintas resoluciones, desde dispositivos móviles hasta computadores, sin afectar el rendimiento, continuidad y usabilidad de la página web

3.2.44 UI web: El aplicativo web cuenta con una interfaz web que es intuitiva para los usuarios y de fácil uso, el 90% de las funcionalidades principales, se puede acceder en menos de 3 clics

3.2.45 UI escritorio: El aplicativo de escritorio cuenta con una interfaz de escritorio que permite realizar la gestión de inventario, categorías, órdenes y usuarios mediante una interfaz intuitiva y de fácil uso, el tiempo de respuesta promedio para realizar las distintas tareas de carácter administrativo no supera los 15 segundos.

**Requerimientos para la sincronización de datos entre los sistemas Web y Escritorio:**

3.2.46 Sistema debe ser confiable: El Sistema de ByteMart cuenta con una disponibilidad del sistema de un mínimo del 80% asegurando una operación continua durante las horas de más demanda.

3.2.47 Sistema debe ser eficiente: El sistema garantiza que las operaciones para la gestión de productos, categorías y usuarios debe realizarse en menos de 5 segundos, esto garantiza que la respuesta sea rápida y fluida.

3.2.48 Sistema debe ser escalable: El sistema puede manejar un aumento del 30% en los usuarios, además se considera un aumento del 50% del inventario sin realizar una reestructuración de la infraestructura

3.2.49 Sistema debe ser tolerable a los fallos: El sistema funciona parcialmente en caso de fallos, minimizando el impacto en los usuarios.

3.2.50 Sincronización de datos: El sistema web y el sistema de escritorio mantiene sus datos sincronizados con el fin de tener un respaldo de los datos en ambos aplicativos.

3.2.51 Respaldo de información: Ambos sistemas cuentan con una base de datos espejo la cual es almacenada y realiza un respaldo de la misma información generada en los aplicativos de producción.

**3.3.2 Seguridad**

Considerando que el sistema de ByteMart trata sobre un ecommerce, que maneje datos sensibles de los clientes, gestión de inventario y transacciones financieras. Tenemos que contar con un control para proteger al sistema de accesos no autorizados o modificaciones accidentales de productos o usuarios.

Se incorpora una base de datos espejo, que sirve como respaldo para la información, en caso de pérdidas o fallos en el sistema. También se adopta una política para garantizar la protección contra acceso no autorizados al sistema, un cambio periódico de contraseñas cada 90 días.

**3.3.3 Fiabilidad**

Los factores de fiabilidad para este sistema cuentan con un 85% bajo condiciones normales de funcionamiento. Además, hay que considerar que, ante fallos críticos, el sistema tiene un tiempo de recuperación no mayor a dos horas. Los incidentes permisibles dependen de su gravedad y serán considerados semanalmente para un mayor monitoreo y control de estos. Lo más importante de este punto y del punto de Seguridad, son los datos de los usuarios, los cuales deben estar protegidos ante posibles pérdidas, en caso de pérdidas, el sistema es capaz de restaurar la información en el menor tiempo posible.

**3.3.4 Disponibilidad**

Se garantiza la disponibilidad total del sistema en un 85%, lo que conlleva a tener el sistema operativo en casi la totalidad de días en el año, el porcentaje restante corresponde a días en los que el sistema se pueda ver sometido a mantención, lo cual ocurrirá fuera de los horarios críticos y en caso de inactividad no planificada del sistema, se debe resolver en un tiempo máximo de dos horas.

El sistema es monitoreado constantemente para prevenir y mitigar caídas, además de tener la capacidad de soportar una alta demanda.

**3.3.5 Mantenibilidad**

La mantenibilidad del sistema ocurre por las siguientes opciones: errores, mejoras o prevención ante problemas del sistema.

* Errores: Errores que puedan surgir en la plataforma web, base de datos o aplicación de escritorio, estos son detectados por usuarios o por sistemas de monitoreo.
* Mejoras: Implementación de nuevas tecnologías, mejoras de los procesos o interfaz del usuario. Esto depende de las necesidades del negocio o solicitudes de los usuarios.
* Prevención: Dependiendo de los estándares de rendimiento del sistema, se debe realizar el proceso de mantenibilidad de la infraestructura, arquitectura, equipos, para que no surja un problema mayor.

Todos estos puntos, deben ir con su correspondiente reporte y se realizarán dependiendo de su tipo en momentos y horarios de bajo flujo de usuarios.

Los encargados de realizar el proceso de mantenibilidad son principalmente:

* Usuarios finales: Principales notificadores de cualquier inconveniente que pueda presentar el sistema.
* Equipo de Desarrollo: Encargados de realizar este proceso, principalmente en la codificación general del sistema, ajustes de la base de datos y aplicación de escritorio.
* Administrador del Sistema: Responsable de mantenimiento de la infraestructura de funcionamiento del sistema, así como también de los equipos, actualizaciones, manejo de respaldos, etc.

El proceso de mantenimiento en cuanto a frecuencia y tiempo:

Errores: tan pronto como se notifique o detecten errores, se debe aplicar las correcciones necesarias en un plazo máximo de **48 horas.**

Mejoras y prevención: Cada **3 meses**, el equipo de desarrollo realizará una revisión del sistema para identificar posibles problemas futuros o desarrollar e implementar nuevos requerimientos solicitados por el cliente.

Reportes: Los reportes de funcionamiento y rendimiento del sistema se realizan **semanalmente** para ser revisados por el equipo técnico.

**3.3.6 Portabilidad**

El sistema ByteMart no presenta especificaciones de portabilidad, por lo cual el software no contará con atributos para facilitar el traslado a otras plataformas y entornos.

**3.4 Otros Requisitos**

De momento, no se especifican otros requerimientos fuera de los mencionados anteriormente.

**4. Propuesta de Planificación**

**4.1 Descripción general acerca de la Planificación**

El proyecto se desarrolla en un período total de cuatro meses, desde su inicio hasta su finalización. Cuenta con la participación de **cuatro** personas, cada una con roles y responsabilidades claramente asignados, para asegurar que el proyecto avance de manera exitosa y se complete satisfactoriamente.

Para que el proyecto pueda finalizar de buena manera, se ha determinado trabajar bajo una metodología tradicional, lo cual se enfoca principalmente en una estructura de desarrollo claramente establecida y lineal, lo cual nos permite enfocarnos en entregables en específico y además de una mejor organización y distribución de tareas.

**4.1.2 Definición del Equipo de Trabajo**

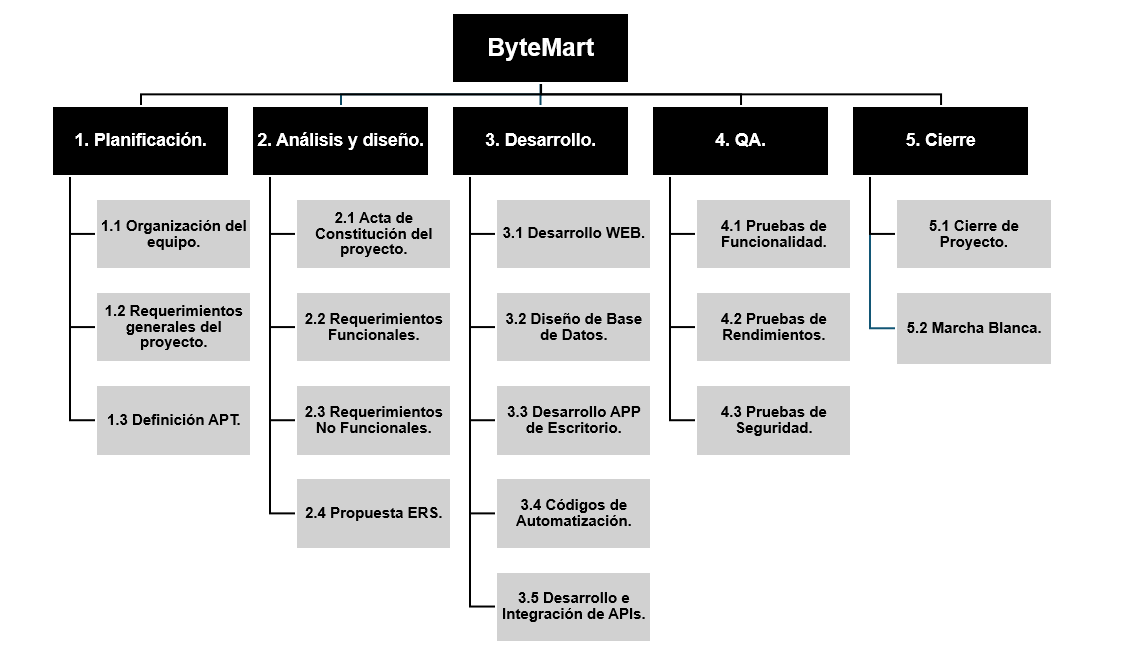
|  |  |  |
| --- | --- | --- |
| **Nombre Integrante** | **Rol definido en el Equipo de Proyecto** | **Funciones** |
| Karla Martínez | Analista de negocio - Equipo desarrollo - Control de calidad - Responsable de comunicaciones | -Define los requisitos, mediará entre el equipo y los stakeholders, supervisará el cumplimiento de los objetivos.  -Diseñar, programar e integrar la solución. -Ejecutar pruebas, garantizar la calidad y documentar resultados.  -Gestionar las comunicaciones internas y externas, mantener informados a los stakeholders y compartir las actualizaciones del proyecto. |
| Michael Marín | Analista de negocio - Equipo desarrollo - Control de calidad | -Define los requisitos, mediará entre el equipo y los stakeholders, y supervisará el cumplimiento de los objetivos.  -Diseñar, programar e integrar la solución. -Ejecutar pruebas, garantizar la calidad y documentar resultados. |
| Cristian Ojeda | Analista de negocio - Equipo desarrollo - Control de calidad | -Define los requisitos, mediará entre el equipo y los stakeholders, y supervisará el cumplimiento de los objetivos.  -Diseñar, programar e integrar la solución. -Ejecutar pruebas, garantizar la calidad y documentar resultados. |
| Vicente Laguna | Project lead - Analista de negocio - Equipo desarrollo - Control de calidad | -Coordinar al equipo, gestionar los recursos y asegurar el cumplimiento de plazos y objetivos generales.  -Define los requisitos, mediará entre el equipo y los stakeholders, y supervisará el cumplimiento de los objetivos.  -Diseñar, programar e integrar la solución. -Ejecutar pruebas, garantizar la calidad y documentar resultados. |

**4.1.3 Definición de Actividades principales del Proyecto**

Las principales actividades del proyecto son las siguientes y se dividen dentro de 5 importantes puntos:

1. **Planificación:** Punto donde se definen los objetivos del proyecto, los requisitos del sistema y alcance de este. Además, se detallan los requisitos más generales del proyecto y para entender de mejor manera el proyecto y ordenar ideas se realiza un informe de proyecto APT. Realización del cronograma para la distribución de las tareas, consideración de los recursos disponibles para el desarrollo y prueba de los sistemas. La fase se guía de las buenas prácticas del PMI para asegurar una planificación efectiva.
2. **Análisis y diseño:** Se comienzan a definir requisitos más específicos del proyecto, tanto funcionales como no funcionales, También se define la propuesta ERS, documento el cual engloba todos los detalles necesarios para el proyecto. Se definen las tecnologías a implementar en los sistemas de escritorio y web, desarrollo RPA, creación e integración de Apis y motor de base de datos.
3. **Desarrollo:** Definición e integración de funcionalidad según los requisitos establecidos, consideración de las tecnologías como Python para el desarrollo de web y creación de bots para la automatización, Flask como herramienta backend para la plataforma web, MySQL como motor de base de datos, Java y NetBeans para el desarrollo la plataforma de escritorio. También la integración de una API de pago online, desarrollo de Apis para la gestión de inventario, visualización de productos destacados y API DPA para el . Además del desarrollo de bots en RPA para la realización de web scraping, extraer y realizar la ingesta de datos con el objetivo de analizar y estudiar las competencias y ver el estado del mercado.
4. **QA:** Se realizan pruebas de funcionalidad, rendimiento y de integración para asegurar la calidad del producto.
5. **Cierre del Proyecto:** Nos aseguramos de que los objetivos y entregables se hayan completado y aprobado de acuerdo con los requisitos establecidos al comienzo del proyecto. Se debe elaborar un informe final de proyecto, además, se debe asegurar que todos los documentos estén completos adecuadamente. Transición del sistema a equipos de operaciones, proporcionando la correspondiente capacitación y estableciendo un plan de soporte.

**4.1.4 Diagrama EDT**



**4.1.5 Carta Gantt**

**4.1.6 Matriz RAM o RACI**



**4.3 Matriz de Riesgos**

En la siguiente matriz, detallamos lo que serían los riesgos identificados dentro de nuestro proyecto ByteMart, dichos riesgos están relacionados con aspectos técnicos, organizacionales y operativos.

En la matriz identificamos el riesgo, los clasificamos con una categoría, además cada uno será catalogado con un determinado Impacto y Probabilidad, también, se define su correspondiente acción de mitigación con el responsable de ello.



**5. Anexos**

**5.1 Acta de Proyecto**



**5.2 Matriz Especificación de Requerimientos**

****

**5.3 Diagrama de Casos de Uso General**

**Se considera en modelo 4+1**

****

**5.4 Planilla Casos de Uso**

****

**5.5 Prototipado de Software**

**5.6 Modelos de Proceso de negocio**

**5.7 Diagrama de actividades de su solución**

**Se considera en modelo 4+1**

****

**5.8 Modelo Base de Datos**

****

**5.9 Diccionario de datos**

**5.10 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo**

