**DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE (SRS)**

**Stiven Daniel Robles Marqués**

**MÓDULO: BASE DE DATOS**

**Centro de Comercio y Servicios**

**Soledad, Atlántico. 2025**

**Tabla de contenido**

[**1.** **REVISION HISTÓRICA** 3](#_Toc189323004)

[**2.** **TABLA DE FIGURAS** 4](#_Toc189323005)

[**2.1.** **Diagrama ER: Base de datos.** 4](#_Toc189323006)

[**3.** **MÓDULO DEL SISTEMA** 5](#_Toc189323007)

[**3.1.** **Funciones del Módulo.** 5](#_Toc189323008)

[**3.2.** **Definiciones, Acrónimos y Abreviaciones** 6](#_Toc189323009)

[**4.** **REQUISITOS** 6](#_Toc189323010)

[**4.1.** **Requisitos Funcionales.** 6](#_Toc189323011)

[**4.2.** **Requisitos no Funcionales.** 12](#_Toc189323012)

# **REVISION HISTÓRICA**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Descripción | Autor | Versión |
| 01/02/2025 | Creación del documento. | Stiven Robles | 1.0 |
| 05/02/2025 | Fase de análisis sobre entidades del módulo. | Stiven robles | 1.1 |
| 07/02/2025 | Elaboración de diagrama ER de funcionamiento del módulo respecto a la db. | Stiven robles | 1.3 |
| 10/02/2025 | Especificación de funcionalidades del módulo. | Stiven Robles | 1.4 |
| 11/02/2025 | Especificación de requerimientos y finalización del módulo. | Stiven Robles | 1.6 |

# **TABLA DE FIGURAS**

## **Diagrama ER: Base de datos.**

# **MÓDULO DEL SISTEMA**

Este módulo fue pensado para llevar a cabo, administrar y representar todas las funcionalidades encaminadas a la recolección y almacenamiento de datos en la base de datos, a su vez el módulo es el encargado que representar como se llevan a cabo procesos cotidianos en el software de manera sistemática y práctica, además de gráfica, para su mayor comprensión desde un enfoque más claro y fácil de deducir. Explica cómo se comporta la base de datos ante funciones del sistema para con el usuario.

## **Funciones del Módulo.**

**Almacenamiento de datos:** Esta función es la más importante, ya que sin ella la integridad del sistema no significaría lo que significa en este momento puesto que, una aplicación sin datos almacenados se asemeja a un caparazón vacío. Esta es la fuente que permite que todo funcione, que el sistema recuerde tus credenciales a la hora de iniciar sesión, de recordar el saldo ficticio de cada usuario a la hora de comprar y el que se encarga que almacenar la información que desglosa de una transacción en el sistema.

**Integridad de los datos**: Esta funcionalidad es la encargada de mantener al margen, de asegurar, establecer y sostener un enlace funcional entre las tablas de la base de datos para concretar, conectar y permitir que la información se almacene de manera correcta, además de permitir a su vez que se pueda recuperar la información correcta cuando sea requerida.

**Respaldo y recuperación:** Funcionalidad importante en caso de emergencias o fallos en el sistema. Esta funcionalidad es la encargada de realizar copias de seguridad de cada tabla, información o dato de la base de datos como método de respaldo en caso de accidentes de código, fallos en el sistema o respaldos posteriores.

* 1. **Definiciones, Acrónimos y Abreviaciones.**

|  |  |
| --- | --- |
| Término | Definición |
| SQL | Es un lenguaje de programación de consulta estructurada, utilizado para interactuar con bases de datos relacionales. |
| ORM | Object-Relational-Mapping, es una herramienta que se usa para conectar la capa de la base de datos con la capa de negocio de un sistema por medio del mapeo de las tablas a entidades. |
| Relacion 1:N | Relación o notación que se usa en esquemas DER (Diagrama Entidad-Relación) para definir un enlace de uno a muchos. |
| Relacion N:N | Relación o notación que se usa en esquemas DER (Diagrama Entidad-Relación) para definir un enlace de muchos a muchos. |
| Relación N:1 | Relación o notación que se usa en esquemas DER (Diagrama Entidad-Relación) para definir un enlace de muchos a uno. |
| Backup | Proceso en la base de datos que consiste en la realización de copias de seguridad de todos los datos de un sistema en caso de necesitarlas o de fallos del mismo. |
| DDL | Data Definition Language, Son conocidos como comandos de SQL para definir las estructuras de las tablas en las bases de datos y sus relaciones. |
| DB | Data Base, abreviatura en ingles de “Base de Datos”, que a su vez es una estructura informática diseñada para el almacenamiento de datos de un sistema o programa, etc. |

# **REQUISITOS**

## **Requisitos Funcionales.**

* + 1. **Requisito Funcional RF01.**

Esta funcionalidad le permite al administrador de la información, al técnico, o cualquier actor que tenga acceso a la información sensible del sistema manipular la información relacionada con la gestión de los usuarios del software. El objetivo de esta funcionalidad es tener las herramientas necesarias para manipular a disposición a un(os) usuario(s) del sistema.

|  |  |
| --- | --- |
| Requisito | Gestión de usuarios (RF01). |
| Dependencias | Node.Js, Sequelize, Mysql |
| Actores: | Administradores. |
| Precondición: | Usuario autenticado como Administrador. |
| Prioridad: | Alta |

|  |  |
| --- | --- |
| Curso normal | Observaciones |
| El sistema debe realizar los siguientes procesos:   * Debe permitirle al Administrador encargado de realizar funciones pertinentes para la correcta gestión de los usuarios como lo son la creación, lectura, actualización y eliminación de datos asociados a los usuarios y sus funcionalidades en el sistema. |  |
| Alternativas | Observaciones |
| Una alternativa para el registro del usuario es la siguiente: |  |
| Excepciones | Observaciones |
| Conjunto de errores y excepciones si el usuario:   * Presenta error si los datos se encuentran incompletos. * Si la información eliminada, agregada o modificada no cumple con los requisitos del sistema o altera directa o indirectamente algún método o funcionalidad. |  |

* + 1. **Requisito Funcional RF02.**

Esta funcionalidad le permite el correcto almacenamiento de la información proporcionada de los vendedores al sistema y propiciar al correcto uso del sistema. Los productos según el común estándar manejado por el sistema son guardados siguiendo un esquema necesario en el que se recopilan los atributos necesarios para la identificación del artículo, indexando la tabla y haciendo mucho más fácil el acceso a esta información por medio de peticiones HTTP. Esta relación en la base de datos esta representada por 3 tablas: La tabla de Usuarios que tiene la posibilidad de realizar muchas transacciones (R1:N) la tabla de Transacciones que tiene una relación de N:1 con la tabla Usuarios y de N:N con la tabla de productos (Funciona como tabla intermediaria) y al final la tabla de Productos que tiene como relación 1:N con la tabla de Transacciones.

|  |  |
| --- | --- |
| Requisito | Almacenamiento de productos (RF02) |
| Dependencias | Node.Js, Sequelize, MySQL |
| Actores: | Usuarios y Vendedores |
| Precondición: | El producto debe tener los datos válidos y los requerimientos y restricciones que impone el sistema. |
| Prioridad: | Alta |

|  |  |
| --- | --- |
| Curso normal | Observaciones |
| El sistema debe realizar los siguientes procesos:   * Registrar los datos proporcionados por el usuario. * Recibir y analizar los datos enviados. * Gestionar, validar y filtrar los datos para luego ser almacenados. * Asociar un ID único y representativo a cada producto almacenado. * Asociar el Id\_producto al vendedor que lo ha creado desde el software. |  |
| Alternativas | Observaciones |
| Una alternativa para el registro del usuario es la siguiente: |  |
| Excepciones | Observaciones |
| Conjunto de errores y excepciones si el usuario:   * No proporciona los datos necesarios. * No cumple con los requerimientos que exige el sistema. * Fallo en la actualización de la db. * Fallo en la conexión del sistema con la db. |  |

* + 1. **Requisito Funcional RF03.**

Esta funcionalidad es la que les permite a los usuarios del sistema tener una constancia de los movimientos y transacciones que se realizan en el sistema de manera constante y repetida. Es la encargada de almacenar toda la información asociada a las transacciones, movimientos y procesos de cada usuario con su respectivo Id de transacción, el estado de la misma y la descripción que devela brevemente en que consistió el movimiento y el id de los usuarios que están relacionados. En la base de datos es representada como una tabla intermediaria que tiene relación tanto con la tabla de Usuarios como la tabla de Productos, además esta tiene un relación de N:1 con el estado que los productos comprados o vendidos puedan tener.

|  |  |
| --- | --- |
| Requisito | Almacenamiento de transacciones (RFn3) |
| Dependencias | Node.Js, Sequelize, MySQL. |
| Actores: | Usuarios del sistema |
| Precondición: | La realización de transacciones dentro del entorno cerrado del sistema |
| Prioridad: | Media |

|  |  |
| --- | --- |
| Curso normal | Observaciones |
| El sistema debe realizar los siguientes procesos:   * La capa de negocio recibe la información de la capa visual y esta es procesada para luego ser analizada y gestionada por la capa de datos del sistema. * La información recolectada de la transacción del sistema es almacenada con una llave primaria única para la recuperación eficaz de los datos en caso de ser requeridos. * Los datos enviados se almacenan en la tabla “Transacciones” la cual es la encargada de registrar tanto las compras/ventas de los productos que ocurren en el entorno. |  |
| Alternativas | Observaciones |
| Una alternativa para el registro del usuario es la siguiente: |  |
| Excepciones | Observaciones |
| Conjunto de errores y excepciones si el usuario:   * Fallos en la recolección de la información * Fallos durante la transacción. * Fallos de conexión con la db. * Fallos del sistema en general. * Mala conexión a internet. * Mensaje general: “Está ocurriendo un problema, la transacción no ha sido posible”. |  |

**Requisito Funcional RF04.**

Esta funcionalidad es la encargada de almacenar el capital ficticio de los usuarios y la encargada de recibir cualquier notificación de adición o sustracción de montos en las billeteras virtuales de cada usuario. La información es almacenada por medio de un id que representa el usuario en cuestión y luego la cantidad de capital (ByCoins) que presenta virtualmente en el sistema para realizar cualquier transacción o movimiento. Esta relación es representada como 1:1 ya que una tabla de Saldo únicamente puede estar asociado a Un usuario.

|  |  |
| --- | --- |
| Requisito | Almacenamiento de ByCoins (RF04) |
| Dependencias | Node.Js, Sequelize, MySQL |
| Actores: | Usuarios y compradores del sistema. |
| Precondición: | Tener sesión iniciada y con credenciales válidos. |
| Prioridad: | Alta |

|  |  |
| --- | --- |
| Curso normal | Observaciones |
| El sistema debe realizar los siguientes procesos:   * El sistema la crear una sesión asigna un valor capital fijo a la cuenta del usuario que le permitirá adquirir productos de manera ficticia. Dichos productos luego de ser comprados serán almacenados en la base de datos con un id asociado al usuario que adquirió el producto, este hará parte del stock del nuevo usuario permitiéndole entonces la facilidad de volverse vendedor y revender el producto anteriormente vendido para multiplicar su capital. * Las transacciones de compra o venta que ocurren en el sistema son monitoreadas y de estas depende la integridad del saldo actual del usuario. * En un caso de compra el saldo será descontado de la cuenta del usuario comprador automáticamente y el saldo fijo nuevo será actualizado y procesado. * En caso de una venta el saldo será adicionado a la cuenta asociada del vendedor automáticamente y el sado fijo nuevo será actualizado y procesado. |  |
| Alternativas | Observaciones |
| Una alternativa para el registro del usuario es la siguiente: |  |
| Excepciones | Observaciones |
| Conjunto de errores y excepciones si el usuario:   * Fallos en la conexión a la base de datos. * Fallo en la utilización de métodos o funcionalidades * Eliminación de la tabla que almacena los ByCoins de cada usuario. * Fallos o errores en la asignación de los id que representa el saldo de cada usuario en el sistema. * Error en el sistema de proporcionar saldo al iniciar sesión por vez primera. |  |

## **Requisitos no Funcionales.**

**RNF01: Fiabilidad:** El sistema debe mantener una disponibilidad del 99.5% para evitar interrupciones en el acceso a la base de datos y evitar interrupciones en las funcionalidades.

**RNF02: Seguridad:** Los respaldos deben hacerse cada 3(tres) días y deben almacenarse en ubicaciones seguras y confiables.

**RNF03: Rendimiento:** Los tiempos de consulta y peticiones del sistema no deben propasar los 2 segundos en espera.

**RNF04: Mantenibilidad:** La base de datos debe cumplir con el estándar establecido por la ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).

**RNF05: Portabilidad**: Las funcionalidades y peticiones que se pueden realizar deben ser compatibles con el estándar de los navegadores mas usados actualmente.

**RNF06: Escalabilidad:** El sistema deben soportar un número de hasta 1000 usuarios concurrentes o realizando peticiones sin degradar o disminuir su rendimiento.