**DATOS DEL PROYECTO:**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOMBRE DEL PROYECTO** | **Inventario de objetos** | | |
| **INTEGRANTES DEL PROYECTO:** | **Juan David Cárdenas Pérez**  **Jeiron David Martinez Acevedo** | | |
| **NOMBRE DEL LÍDER DEL PROYECTO:** | | **Juan David Cárdenas Pérez** |

**Propósito**

Este documento especifica los requerimientos y funcionalidades del software de inventario de objetos en bodega, diseñado para facilitar la gestión, control y seguimiento eficiente de los objetos almacenados. La versión 1.0 del software se enfoca en ofrecer una solución ágil y rápida para llevar un inventario detallado y preciso, garantizando la disponibilidad y ubicación de cada objeto en la bodega.

**Alcance del producto / Software**

Alcance del Software de Inventario de Objetos en Bodega

Propósito u objetivo general: Crear una solución que sea eficiente para poder gestionar y controlar el inventario de objetos en bodega.

Beneficios para el área de negocio y la organización: Los beneficios del proyecto son la optimización de procesos, reducción de errores, mayor eficiencia en la gestión de inventarios y mejora en la toma de decisiones basadas en datos precisos.

Objetivos y metas: Los objetivos y metas son la automatización de tareas del inventario, la centralización de información, la generación de reportes y el análisis detallado con la mejora en la eficiencia operaria de la bodega.

**Funcionalidades del producto**

RF01: El sistema debe permitir ingresar al usuario validando los datos correspondientes

RF02: El sistema debe permitir registrar el ingreso de Objetos a la bodega (CRUD)

RF03: Controlar entradas de los objetos en el inventario

RF04: Restar los objetos que se prestan teniendo en cuenta el stock de inventario

RF05: Colocar un límite de tiempo de devolución del objeto que se prestó

RF06: Mostrar el historial del objeto de préstamos

RF07: El sistema debe permitir realizar una búsqueda por medio nombre, código, tipo de objeto de un producto

RF08: El sistema debe permitir hacer devoluciones de un objeto

RF09: Al momento de devolución de un objeto, colocar el estado en que se encuentra el producto

RF10: El sistema debe permitir hacer un formato para poder registrar el manejo de incidentes y daños de los objetos

RF11: El sistema debe permitir registrar préstamos

RF12: El sistema debe permitir llevar un control con fecha, hora, nombre, cédula y demás datos personales requeridos para realizar un préstamo

RF13: El sistema debe generar ventanas emergentes de confirmación

RF14: El sistema debe permitir realizar una búsqueda en los diferentes módulos (Nombre)

RF15: El sistema debe permitir mostrar las especificaciones de cada objeto (Nombre, descripción y Stock)

**Clases y características de usuarios**

1. **Administrador:**

* Frecuencia de Uso: Moderada
* Grupo de Funcionalidades: Gestión y Configuración del Software (incluyendo registro de préstamos, generación de reportes de inventario)
* Privilegios de Seguridad: Acceso completo a todas las funcionalidades del software, capacidad para gestionar usuarios y configuraciones del sistema
* Nivel de Experiencia: Avanzado
* Otros Parámetros: Responsable de la administración y configuración del software, incluyendo la gestión de usuarios y permisos.

**Entorno operativo y arquitectura de software**

* **Plataforma de Hardware**: El software se ejecutará en computadoras y estará disponible como página web.
* **Plataforma de Software**: Se desarrollará utilizando Python con el framework Django para la parte web y se utilizará MySQL como base de datos.
* **Sistema Operativo**: El software será compatible con el sistema operativo Windows.

**Requerimientos funcionales**

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, describiendo el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones.

Los requerimientos se documentan en una matriz de trazabilidad de requerimientos.

* Código de identificación de mayor nivel definido para el requisito. Puede definirse con números, por ejemplo 001, 002, 003, y así sucesivamente.
* Descripción del requisito: Se proporciona una descripción de que comprende o en qué consiste el requisito. La descripción del requisito depende del tipo que sea, por ejemplo: requisitos del negocio, requisitos de los interesados, requisitos funcionales, requisitos no funcionales, requisitos del proyecto o requisitos del producto (solución).
* Estado actual: Puede ser solicitado, aprobado, asignado, completado, cancelado, diferido, aceptado, entre otros.
* Criterios de aceptación: Lista los criterios de aceptación, una lista de puntos o condiciones específicas que deben cumplirse para poder registrar que el requisito ha sido satisfecho.
* Nivel de complejidad: Puede definirse una complejidad de forma cualitativa, por ejemplo: baja, moderada o alta. Esto dependerá del criterio del evaluador.
* Necesidades, oportunidades u objetivos del negocio: Vínculo del requisito con la estrategia de la organización, listando necesidades específicas que tenga el área de negocio, objetivos de la planificación estratégica que busca lograr u oportunidades de negocio o del mercado.
* Objetivo del proyecto: Vínculo del requisito con los objetivos del proyecto. Aquí se establece la trazabilidad entre el requisito y los objetivos específicos del proyecto definidos en su alcance.
* Entregables: De la estructura desagregada de tarea (EDT) en los cuales está inmerso el requisito. Puede especificarse tanto el nombre del elemento de la EDT como su código EDT.
* Diseño del producto: Entregables de la estructura desagregada de tarea (EDT) en los cuales está inmerso el requisito. Puede especificarse tanto el nombre del elemento de la EDT como su código EDT.
* Desarrollo del proyecto: Implicaciones del requisito en el desarrollo del producto. Describe como los procedimientos de trabajo, metodología o estándares usados incorporan el requisito. Esto aplica principalmente para requisitos que definen la forma de trabajar, estándares a cumplir, entre otros.
* Estrategia y escenarios de pruebas: Listado de las estrategias y escenarios de pruebas que se contemplarán para validar la aceptación del requisito. Estos se definen a partir de los criterios de aceptación.
* Interesado (Stakeholders) dueño del requisito: código de identificación de mayor nivel definido para el requisito. Puede definirse con números, por ejemplo 001, 002, 003, y así sucesivamente.
* Nivel de prioridad: Según la evaluación de la importancia del requisito para el logro de los objetivos del proyecto, se asigna un nivel de prioridad. Este nivel también puede depender del grado de influencia del interesado y estrategias que se estén empleando para gestionar la participación de los interesados.

**RF001: El sistema debe permitir ingresar al usuario validando los datos correspondientes**

|  |  |
| --- | --- |
| **Funcionalidad** | El sistema debe permitir ingresar al usuario validando datos Descripción: Permitir registrar usuarios al sistema validando los datos correspondientes |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | El sistema permitira registrar usuarios nuevos (Administradores) validando siempre sus respectivos datos |
| **Información de contacto** | El usuario ingresará los detalles del objeto a través de un formulario en la interfaz del sistema |
| **Requerimientos funcionales** | 1. El sistema debe permitir ingresar datos personales del usuario 2. El sistema debe registrar la fecha y hora de ingreso del usuario 3. El sistema debe permitir ingresar cualquier otra información relevante del usuario, como su descripción o características especiales en el espacio de (otros)   El sistema debe generar un código único para cada usuario ingresado a la bodega |
| **Software de pruebas** | Se deben probar casos donde se ingresen objetos con información incompleta o incorrecta para garantizar que el sistema si maneje correctamente las situaciones dentro del sistema |

**RF002: El sistema debe permitir registrar el ingreso de objetos a la bodega (CRUD)**

|  |  |
| --- | --- |
| **Funcionalidad** | El sistema debe permitir ingresar objetos que dan los proveedores  por medio de un formulario |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | El sistema permitirá registrar objetos a la base de datos del sistema y almacenarlos allí con su respectiva funcionalidad |
| **Información de contacto** | El usuario ingresará los detalles del objeto a través de un formulario en la interfaz del sistema |
| **Requerimientos funcionales** | 1. El sistema debe permitir ingresar objetos a la bodega y quedar   almacenados   1. El sistema debe permitir ingresar cualquier otra información relevante del objeto, como su descripción o características especiales en el espacio de (otros)   El sistema debe generar un código único para cada objeto ingresado a la bodega |
| **Software de pruebas** | Se deben probar casos donde se ingresen objetos con información incompleta o incorrecta para garantizar que el sistema si maneje correctamente las situaciones dentro del sistema |

**RF003: Controlar entradas de los objetos en el inventario**

|  |  |
| --- | --- |
| **Funcionalidad** | Restar los objetos prestados del stock de la bodega Descripción: Esto permite actualizar el stock de la bodega al realizar el préstamo de objetos, donde se reduzca la cantidad disponible dentro de la función de los objetos prestados |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Cuando un objeto es prestado, se espera que el sistema pueda actualizar automáticamente la cantidad disponible del objeto en la bodega |
| **Información de contacto** | La reducción del stock se activará automáticamente al registrar un préstamo en el sistema |
| **Requerimientos funcionales** | 1. El sistema debe restar la cantidad de objetos prestados del stock disponible en la bodega 2. El sistema debe registrar la cantidad de objetos prestados en una transacción de préstamo 3. El sistema debe mantener un registro de los objetos prestados y su estado actual 4. El sistema debe permitir la devolución de objetos prestados, lo que aumentará la cantidad disponible en el stock 5. El sistema debe generar alertas cuando un objeto prestado no es devuelto dentro del plazo establecido |
| **Software de pruebas** | Se deben probar casos donde se registren préstamos y devoluciones de objetos para asegurar que el stock se actualice correctamente |

**RF004: Restar los objetos que se prestan teniendo en cuenta el stock de inventario**

|  |  |
| --- | --- |
| **Funcionalidad** | Restar los objetos prestados del stock de la bodega Descripción: Esto permite actualizar el stock de la bodega al realizar el préstamo de objetos, donde se reduzca la cantidad disponible dentro de la función de los objetos prestados |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Cuando un objeto es prestado, se espera que el sistema pueda actualizar automáticamente la cantidad disponible del objeto en la bodega |
| **Información de contacto** | La reducción del stock se activará automáticamente al registrar un préstamo en el sistema |
| **Requerimientos funcionales** | 1. El sistema debe restar la cantidad de objetos prestados del stock disponible en la bodega 2. El sistema debe registrar la cantidad de objetos prestados en una transacción de préstamo 3. El sistema debe mantener un registro de los objetos prestados y su estado actual 4. El sistema debe permitir la devolución de objetos prestados, lo que aumentará la cantidad disponible en el stock 5. El sistema debe generar alertas cuando un objeto prestado no es devuelto dentro del plazo establecido |
| **Software de pruebas** | Se deben probar casos donde se registren préstamos y devoluciones de objetos para asegurar que el stock se actualice correctamente |

**RF005: Colocar un límite de tiempo de devolución del objeto que se prestó**

|  |  |
| --- | --- |
| **Funcionalidad** | Establecer un límite de tiempo para la devolución de objetos prestados cuando son solicitados por otro funcionario y son de tipo devolutivo Descripción: Esto permite definir un plazo específico en que un objeto prestado debe ser devuelto si es solicitado por otro funcionario y es de tipo devolutivo |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Cuando un objeto prestado de tipo devolutivo es solicitado por otro funcionario, se espera que el sistema pueda establecer automáticamente un límite de tiempo para su devolución |
| **Información de contacto** | Los usuarios recibirán notificaciones sobre el plazo de devolución a través del sistema u otro medio de comunicación como email |
| **Requerimientos funcionales** | 1. El sistema debe identificar si el objeto prestado es de tipo devolutivo 2. El sistema debe establecer automáticamente un límite de tiempo para la devolución del objeto si es solicitado por otro funcionario 3. El sistema debe enviar notificaciones al usuario responsable del objeto prestado sobre el plazo de devolución establecido 4. El sistema debe generar alertas cuando el plazo de devolución se aproxime o expire sin que el objeto haya sido devuelto 5. El sistema debe permitir ajustar manualmente el plazo de devolución si es necesario, dependiendo de las circunstancias específicas del préstamo |
| **Software de pruebas** | Se deben probar casos donde se soliciten objetos prestados de tipo devolutivo para poder asegurar que el sistema pueda establecer y gestionar correctamente los límites de tiempo de devolución |

**RF006: Mostrar el historial del objeto de préstamos**

|  |  |
| --- | --- |
| **Funcionalidad** | Mantener un historial de los objetos prestados para facilitar su seguimiento y ubicación Descripción: Esto permite acceder a un registro histórico de los objetos que han sido prestados, incluyendo información sobre quién lo solicitó y cuándo |
| **Prioridad** | Media |
| **Acciones iniciadoras y comportamiento esperado** | Cuando se necesite ubicar un objeto prestado, los usuarios podrán consultar el historial del objeto para obtener información sobre quién lo solicitó y en qué fecha |
| **Información de contacto** | Los usuarios podrán acceder al historial del objeto a través de una función de búsqueda en el sistema |
| **Requerimientos funcionales** | 1. El sistema debe mantener un registro histórico de los objetos prestados, incluyendo información sobre quién lo solicitó, fecha de préstamo y fecha de devolución 2. El sistema debe permitir buscar objetos prestados por nombre o código único 3. El sistema debe mostrar de manera clara y organizada la información del historial del objeto, facilitando su comprensión y análisis 4. El sistema debe permitir acceder al historial completo de cada objeto prestado, incluyendo todos los préstamos realizados desde su ingreso a la bodega |
| **Software de pruebas** | Se deben probar casos donde se consulte el historial de objetos prestados para asegurar que el sistema muestre correctamente la información, sea fácil de usar y revisar |

**RF007: El sistema debe permitir realizar una búsqueda por medio nombre, código, tipo de objeto de un producto**

|  |  |
| --- | --- |
| **Funcionalidad** | Implementar una barra de búsqueda para encontrar productos en stock de manera rápida y sencilla Descripción: Esto permite a los usuarios que puedan buscar productos en el stock de la bodega donde utilicen una barra de búsqueda útil y de fácil acceso |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Los usuarios podrán ingresar el nombre o código del producto en la barra de búsqueda y obtener resultados relevantes de manera inmediata |
| **Información de contacto** | Los usuarios interactuarán con la barra de búsqueda en la interfaz del sistema |
| **Requerimientos funcionales** | 1. El sistema debe proporcionar una barra de búsqueda visible y accesible desde cualquier pantalla relacionada con el stock de la bodega 2. La barra de búsqueda debe permitir a los usuarios ingresar términos de búsqueda, como el nombre o código del producto 3. El sistema debe realizar búsquedas en tiempo real y mostrar resultados relevantes a medida que se ingresan los términos de búsqueda 4. El sistema debe permitir a los usuarios filtrar los resultados de búsqueda por diferentes criterios, como tipo de producto, ubicación en la bodega, entre otras 5. La barra de búsqueda debe ser fácil de usar y entender, incluso para usuarios con poca experiencia en el sistema |
| **Software de pruebas** | Se deben probar casos donde se realicen búsquedas de productos en stock utilizando diferentes términos, palabras y/o signos para poder confirmar que el sistema muestre resultados correctos y que sean precisos de manera rápida y fácil |

**RF008: El sistema debe permitir hacer devoluciones**

|  |  |
| --- | --- |
| **Funcionalidad** | Registro de devolución de un objeto Descripción: Esto permite registrar una devolucion de un objeto al sistema |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Al devolver un objeto prestado, se espera que el usuario complete un formulario o espacio designado donde se detalle el estado del producto |
| **Información de contacto** | Los usuarios completarán el formulario o espacio entregado para registrar el estado del objeto devuelto |
| **Requerimientos funcionales** | 1. El sistema debe proporcionar un formulario o espacio designado para registrar la devolución correspondiente 2. El formulario debe permitir al usuario agregar comentarios adicionales si es necesario para describir con más detalle el estado del objeto 3. El sistema debe registrar el estado del objeto devuelto junto con la información del préstamo en la base de datos |
| **Software de pruebas** | Se deben probar casos donde se realicen devoluciones de objetos en diferentes estados para asegurar que el sistema maneje correctamente la información del estado del producto |

**RF009: Al momento de devolución de un objeto, colocar el estado en que se encuentra el producto**

|  |  |
| --- | --- |
| **Funcionalidad** | Registro del estado del producto al momento de devolución Descripción: Esto permite registrar el estado del objeto devuelto al momento de su devolución, ya sea en buen estado o con daños |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Al devolver un objeto prestado, se espera que el usuario complete un formulario o espacio designado donde se detalle el estado del producto |
| **Información de contacto** | Los usuarios completarán el formulario o espacio entregado para registrar el estado del objeto devuelto |
| **Requerimientos funcionales** | 1. El sistema debe proporcionar un formulario o espacio designado para registrar el estado del objeto devuelto 2. El formulario debe permitir al usuario seleccionar entre opciones predefinidas que describan el estado del objeto (por ejemplo, "buen estado", "dañado", "irreparable") 3. El formulario debe permitir al usuario agregar comentarios adicionales si es necesario para describir con más detalle el estado del objeto 4. El sistema debe registrar el estado del objeto devuelto junto con la información del préstamo en la base de datos 5. El sistema debe generar alertas o notificaciones en caso de que se reporten daños en el objeto devuelto |
| **Software de pruebas** | Se deben probar casos donde se realicen devoluciones de objetos en diferentes estados para asegurar que el sistema maneje correctamente la información del estado del producto |

**RF010: El sistema debe permitir hacer un formato para poder registrar el manejo de incidentes y daños de los objetos**

|  |  |
| --- | --- |
| **Funcionalidad** | Gestión de incidentes y daños relacionados con los objetos prestados Descripción: Esto permite registrar y gestionar cualquier incidente o daño reportado en los objetos prestados durante su uso |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Cuando se reporta un incidente o daño en un objeto prestado, se espera que el sistema registre los detalles del incidente y tome las acciones apropiadas, como la reparación o reposición del objeto |
| **Información de contacto** | Los usuarios reportarán incidentes o daños a través de un formulario o espacio designado en el sistema |
| **Requerimientos funcionales** | 1. El sistema debe proporcionar un formulario o espacio designado para reportar incidentes o daños en los objetos prestados 2. El formulario debe permitir al usuario describir el incidente o daño de manera detallada, incluyendo la fecha y hora en que ocurrió 3. El sistema debe asociar el incidente o daño reportado con el objeto prestado y el préstamo correspondiente 4. El sistema debe permitir tomar acciones correctivas, como la reparación o reposición del objeto, según la gravedad del incidente o daño 5. El sistema debe generar alertas o notificaciones para el personal responsable en caso de reporte de incidentes o daños |
| **Software de pruebas** | Se deben probar casos donde se registren incidentes o daños en objetos prestados para poder confirmar que el sistema si maneje correctamente la información y tome las acciones adecuadas respecto al estado del objeto |

**RF011: El sistema debe permitir registrar prestamos**

|  |  |
| --- | --- |
| **Funcionalidad** | REGISTRAR INSTRUCTORES (CRUD) Descripción: Esto permite registrar, ver, eliminar y actualizar mediante métodos y poder ser usadas en el sistema |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Los usuarios podrán registrar instructores y tener acceso a las funcionalidades de una CRUD para poder hacer un uso más óptimo del sistema |
| **Información de contacto** | Los usuarios podrán acceder a cualquier funcionalidad de la CRUD según necesidad |
| **Requerimientos funcionales** | 1. El sistema debe permitir al usuario CREAR, EDITAR, VER Y ELIMINAR un instructor 2. El sistema debe tener una facilidad de opciones para poder acceder a esta funcionalidad 3. El sistema deberá ejecutar cualquier cambio en la base de datos según la funcionalidad que se elija 4. El sistema debe permitir guardar y acceder a informes previamente generados para referencia futura 5. El sistema debe permitir programar la generación automática de informes en intervalos de tiempo predefinidos, si es necesario |
| **Software de pruebas** | Se deben probar casos donde se generen informes con diferentes combinaciones de normas y parámetros para poder confirmar que el sistema pueda producir informes completos y correctos según las necesidades del usuario |

**RF012: Llevar control con fecha, hora, nombre, cédula y demás datos personales requeridos para realizar un préstamo**

|  |  |
| --- | --- |
| **Funcionalidad** | Control de información detallada de préstamos Descripción: Esto permite registrar información completa de los préstamos realizados, incluyendo fecha, hora, nombre, cédula y otros datos personales necesarios |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Al realizar un préstamo de un objeto, se espera que el sistema pueda llevar un control de los datos que se requieren para un prestamo |
| **Información de contacto** | Los datos del solicitante se ingresarán en un formulario al registrar un préstamo en el sistema |
| **Requerimientos funcionales** | El sistema debe almacenar todos los datos personales requeridos para un préstamo  El sistema deberá mostrar los datos personales de la persona que requiere un préstamos  El sistema debe asociar la información del préstamo con el solicitante |
| **Software de pruebas** | Se deben probar casos donde se registren préstamos con diferentes combinaciones de datos del solicitante para asegurar que el sistema maneje correctamente la información |

**RF013: El sistema debe generar ventanas emergentes de confirmación**

|  |  |
| --- | --- |
| **Funcionalidad** | El sistema debe generar ventanas emergentes de confirmación Descripción: Esto permite que en cada proceso que lo necesite se despliegue una ventana emergente de confirmación |
| **Prioridad** | Media |
| **Acciones iniciadoras y comportamiento esperado** | Los usuarios verán una ventana emergente que confirmara el proceso que están realizando los usuarios en los procesos que lo requieran |
| **Información de contacto** | El usuario confirma el proceso que está realizando |
| **Requerimientos funcionales** | 1. El sistema debe mostrar una ventana de emergencia con opciones booleanas ej: (¿CONFIRMA ELIMINAR XXX?   “SI” o “NO”).   1. El sistema adquiere esta funcionalidad por seguridad de los usuarios |
| **Software de pruebas** | Se deben probar casos donde las ventanas emergentes se desplieguen y confirmen el caso de uso deseado |

**RF014: El sistema debe permitir realizar una búsqueda en los diferentes módulos**

|  |  |
| --- | --- |
| **Funcionalidad** | El sistema debe permitir realizar una búsqueda en los diferentes módulos Descripción: Esto permite que en cada proceso que lo necesite se puedan realizar búsquedas de objetos, instructores entre otras cosas mas segun la necesidad del usuario |
| **Prioridad** | Media |
| **Acciones iniciadoras y comportamiento esperado** | Los usuarios desplegaran una barra de busqueda donde se puede encontrar objetos, instructores entre otras cosas más |
| **Información de contacto** | El usuario realiza una búsqueda en específico |
| **Requerimientos funcionales** | 1. El sistema debe mostrar una ventana de búsqueda donde se pueda filtrar los objetos deseados 2. El sistema necesitará de una barra de búsqueda para facilitar su navegabilidad en el sistema |
| **Software de pruebas** | Se deben probar casos donde se busquen objetos con sus respectivos nombres e igualmente nombres errados |

**RF015: El sistema debe permitir mostrar las especificaciones de cada objeto (stock)**

|  |  |
| --- | --- |
| **Funcionalidad** | Controlar la existencia de objetos en la bodega (stock) Descripción: Esto permite poder mantener actualizado el inventario de objetos disponibles en la bodega, donde se muestre la cantidad disponible de cada objeto |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Los usuarios dentro del área logística o de almacenamiento podrán consultar el stock disponible de objetos desde cualquier ubicación dentro del centro de formación. Se espera que el sistema muestre de manera precisa la cantidad de cada objeto en stock |
| **Información de contacto** | El usuario podrá acceder a la información de stock a través de una interfaz en el sistema |
| **Requerimientos funcionales** | 1. El sistema debe mostrar la cantidad disponible de cada objeto en la bodega 2. El sistema debe permitir realizar búsquedas de objetos por nombre 3. El sistema debe actualizar automáticamente la cantidad disponible de un objeto cuando se realiza una entrada o salida de la bodega 4. El sistema debe mostrar alertas cuando la cantidad de un objeto esté por debajo de un umbral mínimo establecido 5. El sistema debe permitir realizar un inventario físico para conciliar el stock registrado con el stock real |
| **Software de pruebas** | Se deben probar casos donde se realicen entradas y salidas de objetos para asegurar que el stock se actualice correctamente |

**Requerimientos no funcionales**

**RNF001: Hardware**

|  |  |
| --- | --- |
| **Funcionalidad** | Requisitos de hardware para el sistema Descripción: Especificación de los componentes de hardware que son necesarios para ejecutar el sistema de manera óptima |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Los usuarios interesados en la instalación del sistema desde diferentes ubicaciones geográficas |
| **Información de contacto** | Los usuarios deben verificar que sus sistemas cumplan con los requisitos mínimos de hardware antes de proceder con la instalación |
| **Requerimientos funcionales** | * Procesador: Intel Core i3, AMD SERIES 3000 * Memoria RAM: 8GB * Tarjeta de video: Nvidia GT 710, AMD RX 550 * Conexión a Internet: Velocidad mínima de +10Mbps * Software de pruebas: Se deben realizar pruebas de compatibilidad del sistema con diferentes configuraciones de hardware para garantizar un rendimiento óptimo |
| **Software de pruebas** | Se deben realizar pruebas de compatibilidad del sistema con diferentes configuraciones de hardware para poder garantizar un rendimiento óptimo |

**RNF002: Lenguaje de Programación**

|  |  |
| --- | --- |
| **Funcionalidad** | Especificación del lenguaje de programación utilizado para el desarrollo del sistema Descripción: Definición de los lenguajes de programación que serán utilizados en el desarrollo del sistema |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Desarrolladores de software encargados de la implementación del sistema |
| **Información de contacto** | Los desarrolladores deben asegurarse de utilizar los lenguajes de programación especificados durante el desarrollo del sistema |
| **Requerimientos funcionales** | * Python 3.11.4 * HTML5 * MySQL |
| **Software de pruebas** | Se deben realizar pruebas exhaustivas para garantizar la compatibilidad y funcionalidad adecuada del sistema con los lenguajes de programación especificados |

**RNF003: Conectividad**

|  |  |
| --- | --- |
| **Funcionalidad** | Especificación de los requisitos de conectividad del sistema Descripción: Definición de los requisitos relacionados con la conectividad del sistema, donde se incluye el hosting y la conexión a Internet |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Administradores de sistemas encargados de configurar la infraestructura de conectividad |
| **Información de contacto** | Los administradores deben asegurarse de contar con un hosting adecuado para que se permita cargar la base de datos y poder establecer una conexión HTTP para el sistema |
| **Requerimientos funcionales** | * Hosting: Compartido o privado, que sea capaz de cargar la base de datos y permitir una conexión HTTP * Conexión a Internet: El sistema requiere una conexión a Internet para que funcione correctamente |
| **Software de pruebas** | Se deben realizar pruebas de conectividad para verificar que el sistema pueda acceder al hosting y a Internet de manera efectiva |

**RNF004: Marketing o Publicidad**

|  |  |
| --- | --- |
| **Funcionalidad** | Definición de las actividades de marketing o publicidad asociadas al sistema Descripción: Especificación de las actividades de marketing o publicidad que se llevarán a cabo en relación con el sistema de información |
| **Prioridad** | Baja |
| **Acciones iniciadoras y comportamiento esperado** | Equipo de marketing encargado de la promoción del sistema |
| **Información de contacto** | El equipo de marketing debe coordinar las actividades de promoción del sistema con el cliente |
| **Requerimientos funcionales** | * Las actividades de marketing no dependerán del sistema de información en sí mismo |
| **Software de pruebas** | N/A |

**RNF005: Inicios de Sesión**

|  |  |
| --- | --- |
| **Funcionalidad** | Especificación de los requisitos de inicio de sesión del sistema Descripción: Definición de los requisitos relacionados con el inicio de sesión de los usuarios en el sistema |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Usuarios que intentan acceder al sistema |
| **Información de contacto** | Los usuarios deben utilizar credenciales únicas y personales para acceder al sistema |
| **Requerimientos funcionales** | * Los usuarios deben tener un usuario y contraseña únicos e intransferibles * La seguridad de las credenciales es responsabilidad del usuario |
| **Software de pruebas** | Se deben realizar pruebas de seguridad para verificar la robustez del sistema de inicio de sesión |

**RNF006: Mantenimiento**

|  |  |
| --- | --- |
| **Funcionalidad** | Definición de los requisitos de mantenimiento del sistema  Descripción: Especificación de los términos de mantenimiento acordados en el contrato del sistema |
| **Prioridad** | Media |
| **Acciones iniciadoras y comportamiento esperado** | Personal de mantenimiento encargado de mantener el sistema operativo |
| **Información de contacto** | Los instructores deben recibir garantía sobre los términos de mantenimiento estipulados en el contrato |
| **Requerimientos funcionales** | * El sistema debe recibir mantenimiento conforme a los términos estipulados en el contrato. * El cliente debe recibir garantía sobre los términos de mantenimiento acordados. |
| **Software de pruebas** | Se deben realizar pruebas de mantenimiento para verificar que el sistema pueda ser actualizado y reparado según lo acordado en el contrato |

**RNF007: Seguridad**

|  |  |
| --- | --- |
| **Funcionalidad** | Especificación de los requisitos de seguridad del sistema  Descripción: Definición de los requisitos relacionados con la protección del sistema contra accesos no autorizados |
| **Prioridad** | Alta |
| **Acciones iniciadoras y comportamiento esperado** | Personal de seguridad encargado de proteger el sistema contra posibles amenazas |
| **Información de contacto** | Se deben implementar medidas de seguridad adecuadas para proteger el sistema contra accesos no autorizados |
| **Requerimientos funcionales** | * El sistema debe estar protegido contra accesos no autorizados |
| **Software de pruebas** | Se deben realizar pruebas de seguridad para verificar la efectividad de las medidas implementadas para proteger el sistema contra los accesos no autorizados |

**PUNTAJE TOTAL OBTENIDO: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ APROBÓ: SI ( ) NO ( ) PRESENTA PDM : SI ( ) NO ( )**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 **VoBo FIRMA DEL LÍDER DEL PROYECTO**