Requerimientos de Sistema para Maldo Motos

**Version 1.0.0.1 approved**

**Contenido**

[**1. Introducción 1**](#_3znysh7)

[1.1 Propósito 1](#_2et92p0)

[1.2 Estándares del documento 1](#_tyjcwt)

[1.3 Audiencia 1](#_3dy6vkm)

[1.4 Referencias 1](#_1t3h5sf)

[**2. Ambiente operativo 1**](#_4d34og8)

[2.1 Clases y características de los usuarios 1](#_2s8eyo1)

[2.2 Contexto operativo 2](#_17dp8vu)

[2.3 Restricciones de implementación 2](#_3rdcrjn)

[2.4 Documentación del usuario 2](#_26in1rg)

[2.5 Suposiciones y dependencias 2](#_lnxbz9)

[**3. Interfaces 2**](#_35nkun2)

[3.1 Interfaces de Usuario 2](#_1ksv4uv)

[3.2 Interfaces de Hardware 3](#_44sinio)

[3.3 Interfaces de Software 3](#_2jxsxqh)

[3.4 Interfaces de Comunicaciones 3](#_z337ya)

[**4. Funciones de software 3**](#_3j2qqm3)

[4.1 Función #1 3](#)

[**5. Requerimientos no funcionales 4**](#_1ci93xb)

**Historial de revisiones**

| **Autor** | **Fecha** | **Comentarios** | **Versión** |
| --- | --- | --- | --- |
| Maldonado Hernán | 22/05/23 | Primera Redacción | 1.0.0.1 |
|  |  |  |  |

# Introducción

## Propósito

Este documento especifica los requerimientos de sistema para el software de gestión de taller de motos que se desarrollará. El producto está diseñado para ser utilizado por el personal y administradores del taller de motos en la ciudad de Santa Fe, Argentina. Su objetivo principal es optimizar y mejorar la eficiencia de las operaciones diarias del taller, facilitando la gestión de partes diarios, reparaciones, control de stock, generación de informes y, además, se añadirá la funcionalidad de seguimiento del saldo adeudado de los clientes.

## Estándares del documento

## Audiencia

## Referencias

# Ambiente operativo

## Clases y características de los usuarios

El personal administrativo se encargará de tareas administrativas, como la gestión de partes diarios, el control de stock de piezas, la generación de informes y estadísticas, y el seguimiento de saldos adeudados. Utilizarán el sistema de manera regular, realizando tareas de seguimiento y análisis de datos. Se espera que tengan conocimientos informáticos sólidos y habilidades en la manipulación de herramientas de gestión.

## Contexto operativo

En términos de hardware, se requerirá que los equipos cuenten con capacidad suficiente de procesamiento y memoria para ejecutar el software de manera óptima. Se recomienda utilizar computadoras de escritorio o portátiles con versiones actualizadas de Windows 10 y suficiente capacidad de almacenamiento.

El software estará diseñado y optimizado para funcionar de manera eficiente en el entorno Windows y aprovechar las características y funcionalidades proporcionadas por la plataforma.

## Restricciones de implementación

**Requerimientos de lenguajes de programación:** Si existen requerimientos específicos en cuanto a los lenguajes de programación a utilizar, ya sea por políticas internas o restricciones tecnológicas, se deberán considerar estas limitaciones durante el desarrollo del software.

**Limitaciones de hardware:** El desarrollo del software deberá tener en cuenta las limitaciones de hardware de los equipos en los que se ejecutará. Esto implica considerar el rendimiento de los procesadores, la capacidad de memoria y almacenamiento, así como otros componentes necesarios para un funcionamiento óptimo del software.

# Funciones de software

## Funcion #1

Prioridad: Alta

Estímulo/Respuesta:

* Estímulo: El usuario inicia sesión en el sistema con sus credenciales.
* Respuesta: El sistema valida las credenciales y permite el acceso al usuario.

Funciones:

1. Gestión de Historial:

* Registrar nuevas reparaciones, incluyendo detalles como fecha, descripción del problema, piezas utilizadas, tiempo empleado y cliente .
* Consultar el historial de reparaciones de un cliente en particular, incluyendo,motos, fechas, descripciones, piezas utilizadas y costos asociados.

1. Gestión de clientes:

* Registrar y mantener la información de los clientes, incluyendo datos personales, detalles de contacto y motos en su posesión.

1. Gestión de stock:

* Mantener un control del inventario de piezas y repuestos disponibles en el taller.
* Actualizar automáticamente el stock cuando se realizan reparaciones.

1. Seguimiento de saldo adeudado:

* Llevar un registro del saldo adeudado por los clientes.
* Registrar pagos realizados y fechas de vencimiento.

1. Generar informes y estadísticas sobre el rendimiento del taller, incluyendo el número de reparaciones realizadas, ingresos generados, motos más frecuentes en reparación, entre otros.

# Requerimientos no funcionales

**Usabilidad:**

El sistema debe ser intuitivo y de fácil aprendizaje para los usuarios, con una interfaz de usuario clara y organizada.

Los tiempos de respuesta del sistema deben ser rápidos, proporcionando una experiencia fluida y sin demoras perceptibles.

**Eficiencia:**

El sistema debe ser eficiente en el uso de los recursos del hardware y minimizar el consumo de memoria y capacidad de procesamiento.

El tiempo de ejecución de las operaciones y consultas en el sistema debe ser óptimo, brindando respuestas rápidas a las acciones realizadas por los usuarios.

**Seguridad:**

El sistema debe contar con mecanismos de autenticación y autorización para garantizar que solo los usuarios autorizados puedan acceder a la información y funcionalidades correspondientes.

Los datos almacenados en el sistema deben estar protegidos y mantener la confidencialidad, integridad y disponibilidad de la información.

**Escalabilidad:**

El sistema debe ser capaz de manejar un aumento en el volumen de datos y la cantidad de usuarios sin comprometer su rendimiento.

Debe ser posible ampliar o mejorar el sistema sin interrumpir su funcionamiento normal.

**Mantenibilidad:**

El sistema debe ser modular y seguir buenas prácticas de programación, lo que facilitará el mantenimiento y la incorporación de futuras mejoras o actualizaciones.

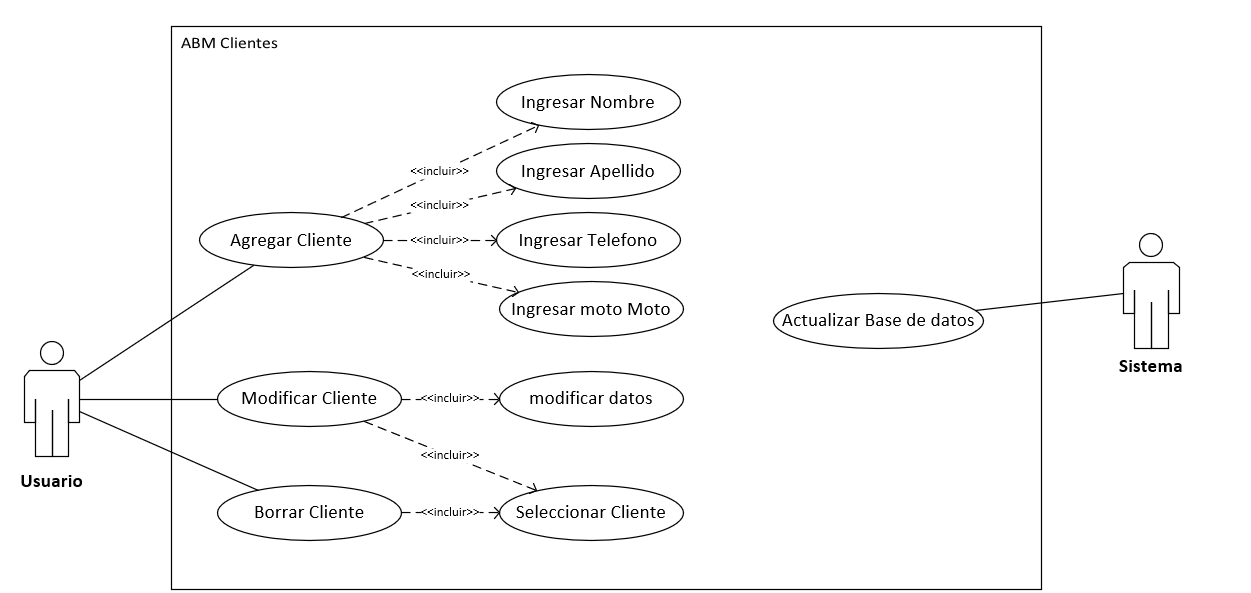
Se deben documentar adecuadamente los componentes del sistema y las funcionalidades implementadas para permitir una fácil comprensión y mantenimiento por parte del personal técnico.

**Criterios de validación:**

* Se realizarán pruebas de usabilidad con usuarios reales para evaluar la facilidad de uso del sistema y su capacidad de respuesta.
* Se medirá el tiempo de respuesta del sistema en diversas operaciones y consultas para verificar su eficiencia.
* Se implementarán medidas de seguridad como autenticación y autorización, y se realizarán pruebas de penetración para garantizar la protección de los datos.
* Se evaluará la capacidad del sistema para manejar un aumento en la carga de trabajo y la escalabilidad mediante pruebas de rendimiento.
* Se verificará la calidad del código y la documentación para asegurar la mantenibilidad del sistema.
* Es importante tener en cuenta que los requerimientos no funcionales pueden variar dependiendo de las necesidades y restricciones específicas del proyecto y el cliente, por lo que estos ejemplos deben adaptarse en consecuencia.

**6. Casos de uso**

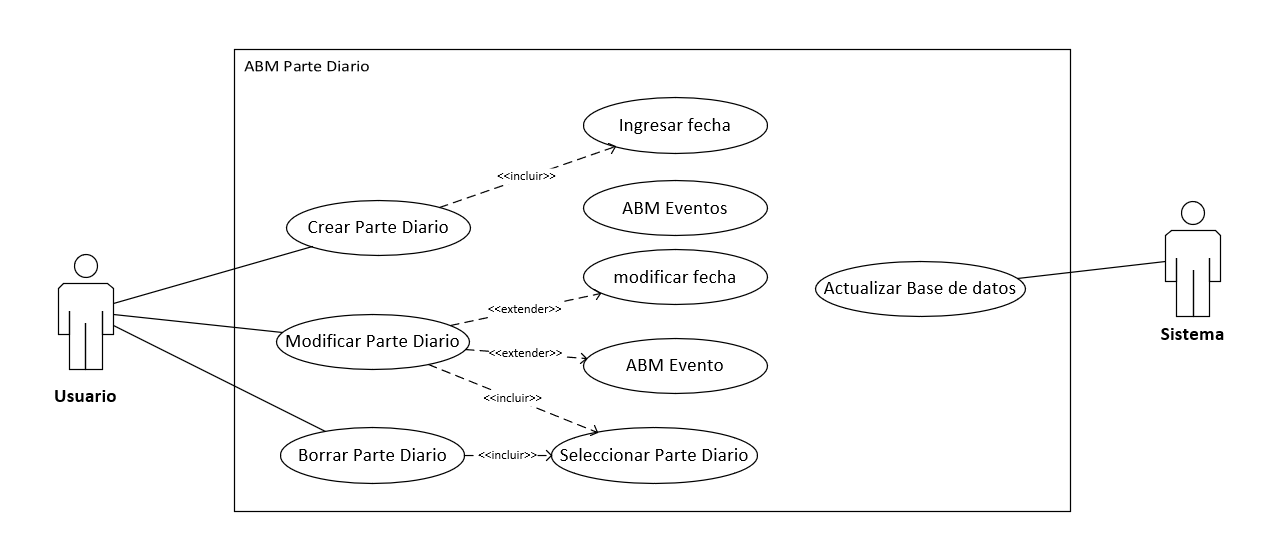
**6.1 ABM Clientes**

****

| **ID** | 1 | | |
| --- | --- | --- | --- |
| **Nombre**: | ABM Clientes | | |
| **Autor**: | Hernán Maldondo | **Modificó:** |  |
| **Fecha de creación:** | 22/05/23 | **Fecha de última modificación:** | 22/05/22 |

| **Actor**: | Usuario |
| --- | --- |
| **Descripción**: | Este caso de uso permite al usuario administrar la información de los clientes, incluyendo la posibilidad de dar de alta nuevos clientes, modificar los datos existentes y eliminar clientes del sistema. |
| **Trigger:** | * El usuario abre el portal * El colaborador ha hecho click Clientes |
| **Precondiciones:** | 1. El usuario debe iniciar sesión |
| **Postcondiciones:** | 1. Se ha actualizado la base de datos clientes |
| **Flujo Normal:**  **(Happy Path)** | 1. El Sistema Muestra la lista de clientes 2. El sistema muestra la lista de clientes registrados. 3. El usuario selecciona la opción de "Agregar cliente". 4. El sistema presenta un formulario en blanco para ingresar los datos del nuevo cliente, como nombre, dirección y detalles de contacto. 5. El usuario completa los campos del formulario con la información del nuevo cliente y selecciona la opción de "Guardar". 6. El sistema valida los datos ingresados y registra al nuevo cliente en la base de datos. 7. El usuario puede seleccionar un cliente existente de la lista y optar por "Modificar cliente". 8. El sistema muestra los datos actuales del cliente seleccionado en un formulario editable. 9. El usuario realiza las modificaciones necesarias en los campos deseados y selecciona la opción de "Guardar cambios". 10. El sistema valida los cambios realizados y actualiza los datos del cliente en la base de datos. 11. El usuario puede seleccionar un cliente existente de la lista y optar por "Eliminar cliente". 12. El sistema muestra una confirmación de eliminación y solicita la confirmación del usuario. 13. El usuario confirma la eliminación del cliente. 14. El sistema elimina al cliente seleccionado de la base de datos. |
| **Flujos alternativos:** | 1. En el paso 6, si los datos ingresados no cumplen con las validaciones requeridas, el sistema muestra mensajes de error y solicita al usuario que corrija los campos correspondientes. 2. En el paso 10, si los cambios realizados no pasan las validaciones requeridas, el sistema muestra mensajes de error y solicita al usuario que realice las correcciones necesarias. |
| **Excepciones:** | 1. Datos incorrectos o incompletos: Si el usuario ingresa datos incorrectos o incompletos al dar de alta o modificar un cliente, el sistema debe mostrar mensajes de error y solicitar al usuario que corrija los campos correspondientes. Por ejemplo, si el usuario omite un campo obligatorio o ingresa un formato incorrecto en un campo, se generará una excepción de datos incorrectos o incompletos. 2. Cliente duplicado: Si el usuario intenta agregar un cliente que ya está registrado en el sistema, se generará una excepción de cliente duplicado. En este caso, el sistema debe mostrar un mensaje de error indicando que el cliente ya existe y ofrecer opciones alternativas, como la búsqueda del cliente existente o la actualización de sus datos. |
| **CU Incluidos:** | Agregar Cliente, Modificar Cliente, Borrar Cliente |
| **Prioridad:** | Alta |
| **Frecuencia de uso:** | Diario |

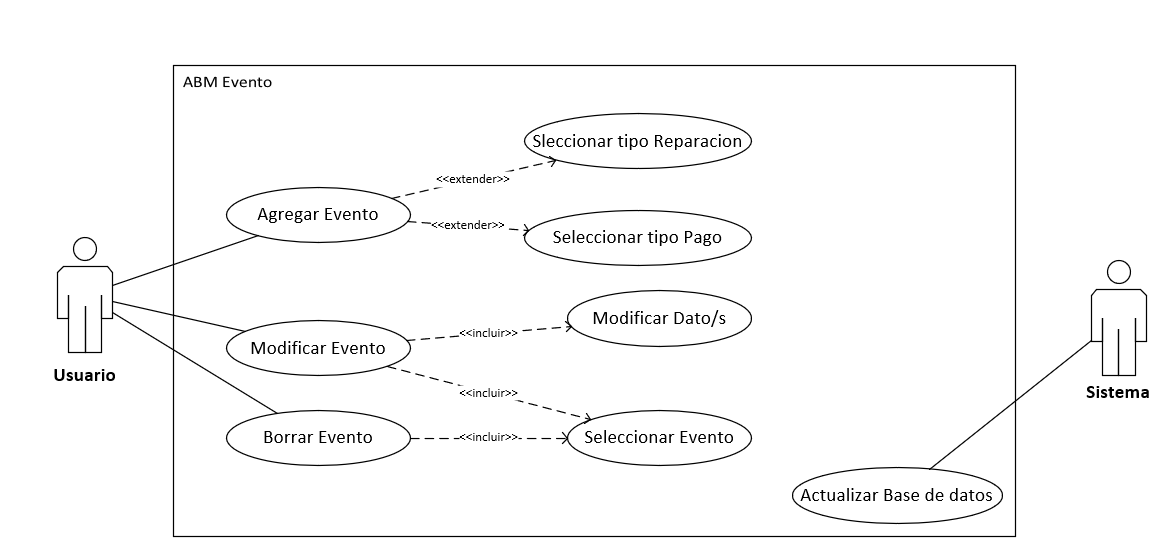
**6.2 ABM Parte Diario**

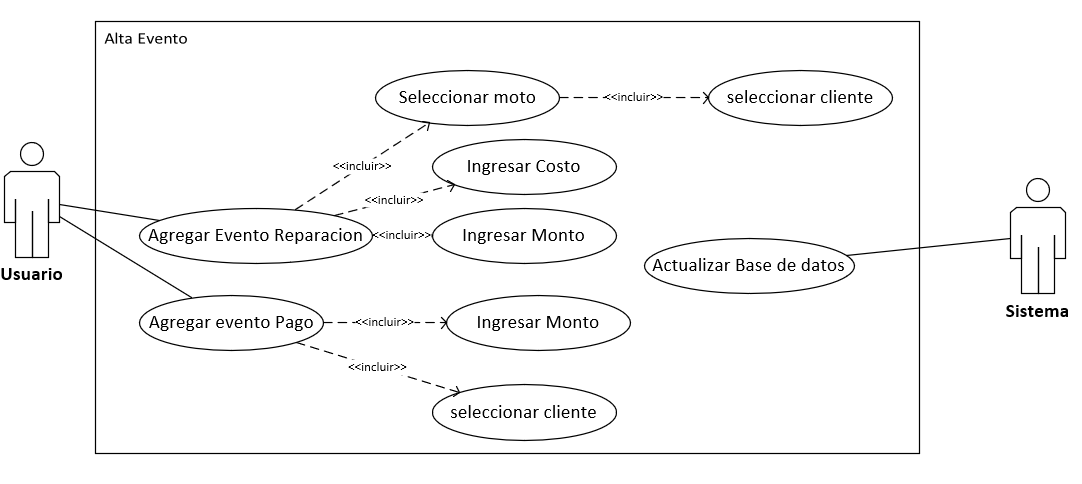
****

| **ID** | 2 | | |
| --- | --- | --- | --- |
| **Nombre**: | ABM Parte Diario | | |
| **Autor**: | Hernán Maldondo | **Modificó:** |  |
| **Fecha de creación:** | 22/05/23 | **Fecha de última modificación:** | 22/05/22 |

| **Actor**: | Usuario |
| --- | --- |
| **Descripción**: | Este caso de uso permite a los mecánicos y al administrador del sistema realizar las operaciones de Alta, Baja y Modificación de los Partes Diarios. Los Partes Diarios son registros que contienen eventos relacionados a reparaciones y pagos. |
| **Trigger:** | * El usuario abre el portal * El usuario ha hecho click Parte Diario |
| **Precondiciones:** | 1. El usuario debe iniciar sesión |
| **Postcondiciones:** | 1. El sistema debe actualizar la base de datos. |
| **Flujo Normal:**  **(Happy Path)** | 1. El sistema muestra la lista de Partes Diarios existentes. 2. El usuario selecciona la opción de "Crear Parte Diario". 3. El sistema muestra un formulario vacío para completar la información del Parte Diario. 4. El usuario ingresa los detalles del Parte Diario, como la fecha, el mecánico responsable, y cualquier información adicional relevante. 5. El sistema muestra una lista de eventos vacia para que el usuario la rellene durante el dia 6. El usuario guarda el Parte Diario al final de la jornada. 7. El sistema valida los datos ingresados y registra el Parte Diario. 8. El sistema muestra la confirmación de que el Parte Diario ha sido creado exitosamente. 9. El usuario puede optar por realizar alguna de las siguientes acciones:   9.1Editar un Parte Diario existente:  a. El usuario selecciona el Parte Diario que desea editar de la lista.  b. El sistema muestra el formulario con los datos actuales del Parte Diario.  c. El usuario modifica los detalles necesarios del Parte Diario.  d. El usuario guarda los cambios.  e. El sistema valida los datos modificados y actualiza el Parte Diario.  f. El sistema muestra la confirmación de que los cambios han sido guardados exitosamente.  9.2Eliminar un Parte Diario existente:  a. El usuario selecciona el Parte Diario que desea eliminar de la lista.  b. El sistema muestra una confirmación de eliminación para solicitar la confirmación del usuario.  c. El usuario confirma la eliminación del Parte Diario.  d. El sistema elimina el Parte Diario.  e. El sistema muestra la confirmación de que el Parte Diario ha sido eliminado exitosamente.   1. El usuario puede elegir salir del sistema o realizar otras acciones relacionadas con la gestión de los Partes Diarios. |
| **Flujos alternativos:** | Si en el paso 7 del flujo básico se detecta que algún campo obligatorio no ha sido completado o que los datos ingresados no cumplen con las validaciones requeridas, el sistema muestra mensajes de error específicos y solicita al usuario que corrija los datos.  Si en el paso 2 del flujo básico el usuario decide cancelar la creación del Parte Diario, el sistema vuelve a la lista de Partes Diarios sin realizar cambios.  Si en el paso 9.1 del flujo básico se selecciona la opción de editar un Parte Diario existente y no se realizan cambios en los datos, el sistema muestra un mensaje indicando que no se han realizado modificaciones.  Si en el paso 9.2 del flujo básico se selecciona la opción de eliminar un Parte Diario existente y el Parte Diario tiene eventos asociados, el sistema muestra un mensaje de advertencia indicando que los eventos también serán eliminados y solicita la confirmación del usuario antes de proceder con la eliminación.  Si en el paso 9.2 del flujo básico se selecciona la opción de eliminar un Parte Diario existente y el usuario cancela la eliminación en la confirmación, el sistema vuelve a la lista de Partes Diarios sin realizar cambios. |
| **Excepciones:** | Datos incorrectos o incompletos: Si el usuario ingresa datos incorrectos o incompletos al dar de alta o modificar un Parte Diario, el sistema debe mostrar mensajes de error y solicitar al usuario que corrija los campos correspondientes. Por ejemplo, si el usuario omite un campo obligatorio o ingresa un formato incorrecto en un campo, se generará una excepción de datos incorrectos o incompletos.  Parte Duplicado: Si el usuario intenta agregar un Parte Diario que ya está registrado en el sistema, se generará una excepción de Parte Diario duplicado. En este caso, el sistema debe mostrar un mensaje de error indicando que el cliente ya existe y ofrecer opciones alternativas, como la búsqueda del cliente existente o la actualización de sus datos. |
| **CU Incluidos:** | Crear Parte Diario, Eliminar Parte diario, Modificar Parte Diario,Agregar Evento, Modificar Evento, Borrar Evento |
| **Prioridad:** | Alta |
| **Frecuencia de uso:** | Diario |

**6.3 ABM Eventos**

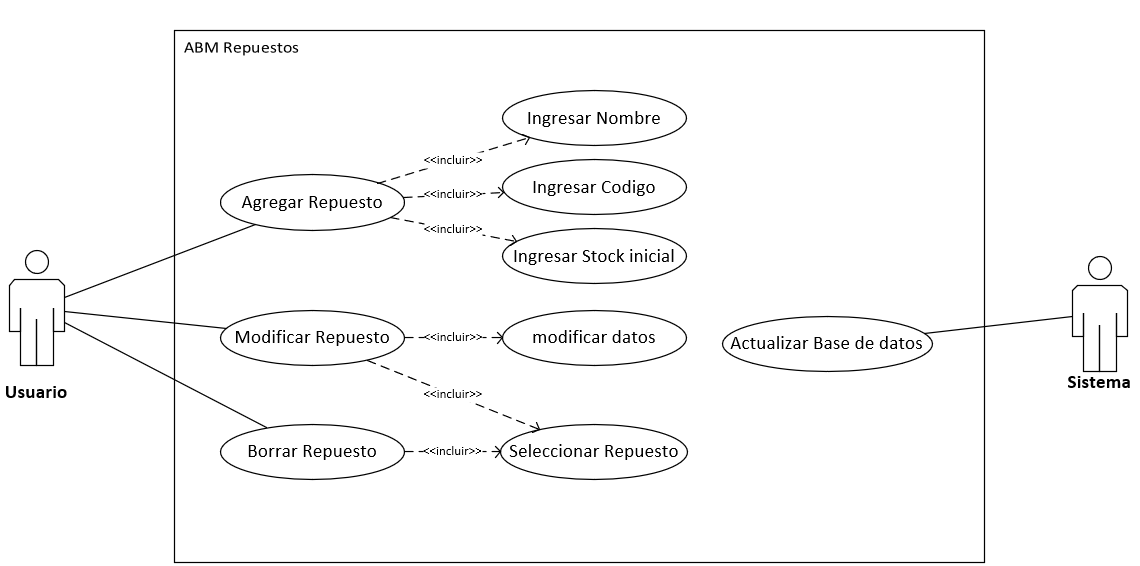
****

****

| **ID** | 3 | | |
| --- | --- | --- | --- |
| **Nombre**: | ABM Evento | | |
| **Autor**: | Hernán Maldondo | **Modificó:** |  |
| **Fecha de creación:** | 22/05/23 | **Fecha de última modificación:** | 22/05/22 |

| **Actor**: | Usuario |
| --- | --- |
| **Descripción**: | Este caso de uso permite a los mecánicos y al administrador del sistema realizar las operaciones de Alta, Baja y Modificación de los Eventos que se registrarán en un Parte Diario. Los eventos pueden ser reparaciones o pagos asociados a las reparaciones. |
| **Trigger:** | * El usuario abre el portal * El usuario ha hecho click Parte Diario |
| **Precondiciones:** | 1. El usuario debe iniciar sesión |
| **Postcondiciones:** | 1. El sistema debe actualizar la base de datos. |
| **Flujo Normal:**  **(Happy Path)** | 1. El usuario selecciona la opción de Nuevo Parte Diario. 2. El sistema muestra una lista de eventos vacía. 3. El usuario selecciona la opción “Ingresar Evento” 4. El sistema muestra un formulario vacío para completar la información del Evento. 5. El usuario ingresa los detalles del Evento, como la descripción, la fecha, el tipo de evento (reparación o pago) , monto ,repuestos utilizados,y la información adicional relevante(Descripción).   5.1. Si el Evento es una reparación:  a. El usuario registra los detalles específicos de la reparación, como la moto involucrada, las partes utilizadas, el costo, etc.  5.2 Si el Evento es un pago asociado a una reparación:  a. El usuario selecciona la reparación correspondiente.  b. El usuario ingresa los detalles del pago, como el monto, el método de pago utilizado, etc.   1. El usuario guarda el Evento. 2. El sistema valida los datos ingresados y registra el Evento en el Parte Diario correspondiente. 3. El sistema muestra la confirmación de que el Evento ha sido creado exitosamente. 4. El usuario puede optar por realizar alguna de las siguientes acciones:   9.1 Editar un Evento existente:  a. El usuario selecciona el Evento que desea editar de la lista.  b. El sistema muestra el formulario con los datos actuales del Evento.  c. El usuario modifica los detalles necesarios del Evento.  d. El usuario guarda los cambios.  e. El sistema valida los datos modificados y actualiza el Evento en el Parte Diario correspondiente.  f. El sistema muestra la confirmación de que los cambios han sido guardados exitosamente.  9.2 Eliminar un Evento existente:  a. El usuario selecciona el Evento que desea eliminar de la lista.  b. El sistema muestra una confirmación de eliminación para solicitar la confirmación del usuario.  c. El usuario confirma la eliminación del Evento.  d. El sistema elimina el Evento del Parte Diario correspondiente.  e. El sistema muestra la confirmación de que el Evento ha sido eliminado exitosamente.   1. El usuario puede elegir salir del sistema o realizar otras acciones relacionadas con la gestión de los Eventos en el Parte Diario. |
| **Flujos alternativos:** | Si en el paso 9 del flujo básico se detecta que algún campo obligatorio no ha sido completado o que los datos ingresados no cumplen con las validaciones requeridas, el sistema muestra mensajes alertando sobre el error.  Si en el paso 8 del flujo básico se selecciona que el Evento es un pago asociado a una reparación, pero no se ha seleccionado ninguna reparación previamente, el sistema muestra un mensaje de error y solicita al usuario que seleccione una reparación antes de continuar. |
| **Excepciones:** | Datos incorrectos o incompletos: Si el usuario ingresa datos incorrectos o incompletos al dar de alta o modificar un cliente, el sistema debe mostrar mensajes de error y solicitar al usuario que corrija los campos correspondientes. Por ejemplo, si el usuario omite un campo obligatorio o ingresa un formato incorrecto en un campo, se generará una excepción de datos incorrectos o incompletos. |
| **CU Incluidos:** | Agregar Evento, Modificar Evento, Borrar Evento, Agregar Parte |
| **Prioridad:** | Alta |
| **Frecuencia de uso:** | Diario |

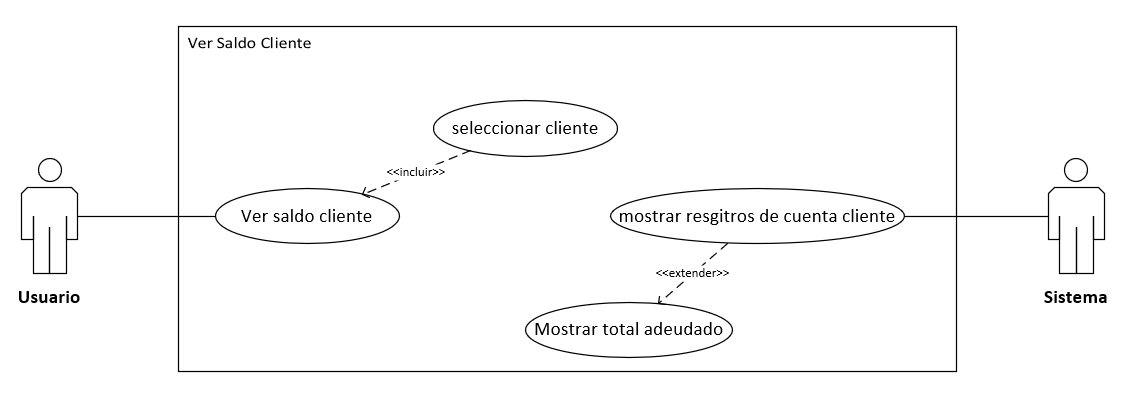
**6.4 ABM Repuestos**

****

| **ID** | 4 | | |
| --- | --- | --- | --- |
| **Nombre**: | ABM Repuestos | | |
| **Autor**: | Hernán Maldondo | **Modificó:** |  |
| **Fecha de creación:** | 23/05/23 | **Fecha de última modificación:** | 23/05/22 |

| **Actor**: | Usuario |
| --- | --- |
| **Descripción**: | Este caso de uso permite al usuario administrar el stock de los repuestos, incluyendo la posibilidad de dar de alta nuevos repuestos, modificar los datos existentes y eliminar repuestos del sistema. |
| **Trigger:** | * El usuario abre el portal * El colaborador ha hecho click Stock |
| **Precondiciones:** | 1. El usuario debe iniciar sesión |
| **Postcondiciones:** | 1. Se ha actualizado la base de datos Stock |
| **Flujo Normal:**  **(Happy Path)** | 1. El Sistema Muestra la lista de Repuestos 2. El usuario selecciona la opción de "Agregar repuesto". 3. El sistema presenta un formulario en blanco para ingresar los datos del nuevo repuesto, como nombre, código, proveedor, stock inicial, etc. 4. El usuario completa los campos del formulario con la información del nuevo repuesto y selecciona la opción de "Guardar". 5. El sistema valida los datos ingresados y registra al nuevo repuesto en la base de datos. 6. El usuario puede seleccionar un repuesto existente de la lista y optar por "Modificar repuesto". 7. El sistema muestra los datos actuales del repuesto seleccionado en un formulario editable. 8. El usuario realiza las modificaciones necesarias en los campos deseados y selecciona la opción de "Guardar cambios". 9. El sistema valida los cambios realizados y actualiza los datos del repuesto en la base de datos. 10. El usuario puede seleccionar un cliente existente de la lista y optar por "Eliminar repuesto". 11. El sistema muestra una confirmación de eliminación y solicita la confirmación del usuario. 12. El usuario confirma la eliminación del repuesto. 13. El sistema elimina al repuesto seleccionado de la base de datos. 14. El usuario puede seleccionar en “Ingresar Pedido”. 15. El sistema presenta un formulario en blanco para seleccionar un repuesto existente (de no encontrarlo deberá ingresarlo volviendo hacia atrás), e ingresar datos como las unidades que han ingresado. |
| **Flujos alternativos:** | 1. En el paso 6, si los datos ingresados no cumplen con las validaciones requeridas, el sistema muestra mensajes de error y solicita al usuario que corrija los campos correspondientes. 2. En el paso 10, si los cambios realizados no pasan las validaciones requeridas, el sistema muestra mensajes de error y solicita al usuario que realice las correcciones necesarias. |
| **Excepciones:** | 1. Datos incorrectos o incompletos: Si el usuario ingresa datos incorrectos o incompletos al dar de alta o modificar un cliente, el sistema debe mostrar mensajes de error y solicitar al usuario que corrija los campos correspondientes. Por ejemplo, si el usuario omite un campo obligatorio o ingresa un formato incorrecto en un campo, se generará una excepción de datos incorrectos o incompletos. 2. Cliente duplicado: Si el usuario intenta agregar un cliente que ya está registrado en el sistema, se generará una excepción de cliente duplicado. En este caso, el sistema debe mostrar un mensaje de error indicando que el cliente ya existe y ofrecer opciones alternativas, como la búsqueda del cliente existente o la actualización de sus datos. |
| **CU Incluidos:** | Agregar Cliente, Modificar Cliente, Borrar Cliente |
| **Prioridad:** | Alta |
| **Frecuencia de uso:** | Diario |

**6.5 Ver Estado de cuenta**

****

| **ID** | 5 | | |
| --- | --- | --- | --- |
| **Nombre**: | Seguimiento saldo clientes | | |
| **Autor**: | Hernán Maldondo | **Modificó:** |  |
| **Fecha de creación:** | 22/05/23 | **Fecha de última modificación:** | 22/05/22 |

| **Actor**: | Usuario |
| --- | --- |
| **Descripción**: | Este caso de uso permite al usuario administrar la información de los clientes, exclusivamente la de su estado de cuenta actual. |
| **Trigger:** | * El usuario abre el portal * El colaborador ha hecho click Clientes * El usuario ha hecho click en Estado de Cuenta |
| **Precondiciones:** | 1. El usuario debe iniciar sesión |
| **Postcondiciones:** | 1. No se han generado cambios en la base de datos |
| **Flujo Normal:**  **(Happy Path)** | 1. El Sistema Muestra un formulario en blanco en el que el usuario podrá ingresar datos del cliente y presiona “buscar” 2. El sistema muestra una lista con los registros de cuenta del cliente. 3. Si el cliente presenta deuda el sistema muestra un mensaje con el monto adeudado. |
| **Flujos alternativos:** | 1. En el paso 2, si los datos ingresados no coinciden con ningún cliente registrado el sistema muestra un mensaje alusivo. |
| **Excepciones:** | Datos incorrectos o incompletos: Si el usuario ingresa datos incorrectos o inexistentes el sistema generará una excepción |
| **CU Incluidos:** | Ver estado de cuenta, Buscar Cliente |
| **Prioridad:** | Alta |
| **Frecuencia de uso:** | Diario |