Proyecto Sistema de gestion

Maldo Motos

Hernan Maldonado

Tercer cuatrimestre Tecnicatura en Desarrollo Software

**Contenido**

[Introducción 1](#_Toc136382301)

[Propósito 1](#_Toc136382302)

[Requerimientos del negocio 1](#_Toc136382303)

[1. Contexto 1](#_Toc136382304)

[1.1 Situación actual 1](#_Toc136382305)

[1.2 Oportunidad del negocio 1](#_Toc136382306)

[1.3 Objetivos de negocio y Criterios de éxito 2](#_Toc136382307)

[1.4 Riesgos 2](#_Toc136382308)

[2. Visión de la solución 3](#_Toc136382309)

[2.1 Declaración de la visión de la solución 3](#_Toc136382310)

[2.2 Funciones principales 3](#_Toc136382311)

[2.3 Suposiciones y dependencias 4](#_Toc136382312)

[3. Contexto del negocio 4](#_Toc136382313)

[3.1 Perfil de los interesados (Stakeholders) 4](#_Toc136382314)

[3.2 Ambiente de operación 4](#_Toc136382315)

[4. Ambiente operativo 5](#_Toc136382316)

[4.1Clases y características de los usuarios 5](#_Toc136382317)

[4.2 Contexto operativo 5](#_Toc136382318)

[4.3 Restricciones de implementación 5](#_Toc136382319)

[Requerimientos del sistema 6](#_Toc136382320)

[1. Funciones de software 6](#_Toc136382321)

[1.1 Función #1 6](#_Toc136382322)

[2. Requerimientos no funcionales 6](#_Toc136382323)

[3. Casos de uso 7](#_Toc136382324)

[4. Diagrama de clases 16](#_Toc136382325)

**Historial de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Autor** | **Fecha** | **Comentarios** | **Versión** |
| Maldonado Hernán | 22/05/23 | Primera Redacción | 1.0.0.1 |
| Maldonado  Hernán | 30/05/23 | Requerimentos de sistema y de negocio | 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.

# Requerimientos del negocio

## 1. Contexto

### 1.1 Situación actual

La organización cliente, un taller de motos, actualmente trabaja con partes diarios y registros en papel, carece de un historial de reparaciones digitalizado y no cuenta con un sistema de control de stock. Estas limitaciones han generado dificultades en la gestión, pérdida de datos y problemas en el abastecimiento. Por lo tanto, se reconoce la necesidad de desarrollar un software para mejorar la eficiencia y solucionar estos obstáculos.

### 1.2 Oportunidad del negocio

El problema a resolver radica en la falta de un sistema de software adecuado en el mercado actual para el taller de motos. Este taller opera en un entorno altamente competitivo, donde la eficiencia, la calidad y la atención al cliente son factores clave para el éxito.

Actualmente, existen algunas soluciones similares en el mercado, como sistemas genéricos de gestión de talleres o software de gestión de inventarios. Sin embargo, estas soluciones no abordan por completo las necesidades específicas del taller de motos y presentan carencias significativas.

Por ejemplo, los sistemas genéricos de gestión de talleres no están diseñados teniendo en cuenta las peculiaridades y los procesos específicos de reparación de motos. Estas soluciones no proporcionan características específicas para el seguimiento y la documentación de las reparaciones de motos, lo que limita la capacidad del taller para brindar un servicio eficiente y de calidad.

En cuanto al software de gestión de inventarios, generalmente se centra en el control de existencias y el abastecimiento, pero no está adaptado a las necesidades específicas de un taller de motos. No brinda funcionalidades para el registro y seguimiento detallado de las piezas y componentes utilizados en las reparaciones de motos, lo que dificulta la gestión del stock y puede llevar a problemas de abastecimiento y demoras en las reparaciones.

Sin un sistema de software específico para el taller de motos, no se pueden resolver de manera efectiva los problemas relacionados con la gestión diaria de tareas, la documentación de reparaciones, el acceso a un historial completo y la implementación de un sistema eficiente de control de stock. Por lo tanto, es fundamental desarrollar un software personalizado que aborde estas necesidades y proporcione una solución integral al taller de motos.

### 1.3 Objetivos de negocio y Criterios de éxito

El cliente busca reducir el tiempo dedicado a la realización de partes diarios en papel y agilizar los procesos de documentación de reparaciones. El objetivo es lograr una reducción del 50% en el tiempo empleado en estas tareas en un plazo de 6 meses. Además, el cliente busca tener un control preciso de las piezas y componentes utilizados en las reparaciones, con el fin de evitar problemas de abastecimiento y reducir el tiempo perdido en la búsqueda de piezas. El objetivo es lograr una reducción del 70% en los tiempos de búsqueda de piezas y una disminución del 20% en las pérdidas debido a problemas de stock en un plazo de 8 meses

.

**Criterios de éxito:**

**Mejora en la productividad:** Se medirá el logro de este objetivo mediante la reducción del tiempo dedicado a tareas manuales y repetitivas, así como el aumento en la eficiencia general del taller.

**Calidad de la documentación**: Se evaluará la precisión y la exhaustividad de la documentación de las reparaciones realizadas, así como la facilidad de acceso y recuperación de la información.

**Reducción de problemas de abastecimiento:** Se medirá el éxito en base a la disminución de tiempos perdidos en la búsqueda de piezas y la reducción de pérdidas financieras debido a problemas de stock.

**Satisfacción del cliente:** Se evaluará la satisfacción general de los clientes del taller a través de encuestas y retroalimentación, considerando la calidad del servicio, la rapidez en las reparaciones y la precisión en la facturación.

### 1.4 Riesgos

Aparición de otros productos mejores: Existe el riesgo de que aparezcan en el mercado soluciones de software más avanzadas y completas que superen las funcionalidades ofrecidas por el producto desarrollado. Esto podría disminuir la demanda y la competitividad del software del cliente. Severidad: Moderada. Impacto en el proyecto: Medio.

Restricciones de tiempo: La falta de tiempo suficiente para desarrollar, probar e implementar el software podría afectar la calidad del producto final y provocar retrasos en la entrega. Severidad: Alta. Impacto en el proyecto: Alto.

Resistencia al cambio por parte de los usuarios: Es posible que los empleados del taller muestren resistencia al cambio al abandonar los métodos tradicionales basados en papel y adaptarse al nuevo sistema de software. Esto podría dificultar la adopción y el uso efectivo del software. Severidad: Moderada. Impacto en el proyecto: Medio.

Problemas para la implementación: Dificultades técnicas o logísticas durante la implementación del software podrían generar retrasos, interrupciones o incluso la incapacidad de implementar el sistema de manera adecuada. Severidad: Alta. Impacto en el proyecto: Alto.

## 2. Visión de la solución

### 2.1 Declaración de la visión de la solución

Nuestra visión es proporcionar al taller de motos una solución de software integral y personalizada que revolucione su forma de trabajo, optimice sus operaciones y mejore la experiencia tanto para el personal como para los clientes. Al implementar nuestro producto, el cliente logrará una transformación completa en su gestión diaria, alcanzando un nivel de eficiencia y competitividad sin precedentes en el mercado.

### 2.2 Funciones principales

* **Gestión de partes diarios:** El producto permitirá al cliente generar y administrar partes diarios de manera digital, agilizando y simplificando el proceso de registro de actividades diarias.
* **Registro y seguimiento de reparaciones:** El sistema proporcionará un módulo especializado para documentar y realizar un seguimiento detallado de las reparaciones realizadas en las motos, incluyendo información sobre fechas, tipo de reparación y piezas utilizadas.
* **Historial completo de reparaciones:** El software mantendrá un historial completo y accesible de todas las reparaciones realizadas en el taller, permitiendo una rápida búsqueda y recuperación de información para fines de seguimiento, evaluación y garantía.
* **Control de stock de piezas y componentes:** El producto contará con un módulo de control de stock que permitirá al cliente gestionar de manera eficiente las existencias de piezas y componentes utilizados en las reparaciones, evitando problemas de abastecimiento y pérdidas financieras.
* **Programación y asignación de tareas:** El sistema ofrecerá la capacidad de programar y asignar tareas a los empleados del taller, optimizando la distribución de trabajo y mejorando la eficiencia operativa.
* **Registro y seguimiento de pagos:** El sistema ofrecerá un seguimiento sobre el saldo adeudado de cada cliente, así como también un registro de los pagos realizados con sus datos correspondientes.

### 2.3 Suposiciones y dependencias

El proyecto puede depender de relaciones existentes o futuras del negocio, como acuerdos con talleres de motos asociados, distribuidores de piezas o socios estratégicos. Estas relaciones son importantes para garantizar la integración adecuada del software en el contexto operativo del cliente y para el éxito general del proyecto.

## 3. Contexto del negocio

### 3.1 Perfil de los interesados (Stakeholders)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Stakeholder*** | ***Beneficio y valor percibido*** | ***Actitudes*** | ***Funciones de interés mayor*** | ***Restricciones*** |
| *Gerente* | *Aumentar la eficiencia de sus empleados, obtener estadísticas* | *Muy interesado en apoyar el producto, puede presionar por calendarios ajustados* | *Control de stock* | *ninguna identificada* |
| *Administrador* | *Mantener la integridad de la información* | *Altamente receptivo; espera buena usabilidad* | *Ingreso de la información concisa y estructurada* | *Necesitará capacitación* |
| *Mecánico* | *Obtener rápidamente a la información sobre el historial de una moto* | *Muy interesado en que el producto sea implementado* | *Obtener la información de las reparaciones pasadas* | *Ninguna identificada* |

### 3.2 Ambiente de operación

El sistema será utilizado por usuarios que se encuentran concentrados en un único taller ubicado en la ciudad de Santa Fe, Argentina, provincia de Santa Fe. Los usuarios accederán al sistema en diferentes momentos del día, como cuando llegan proveedores para actualizar el stock o cuando los clientes retiran sus motos reparadas.

Los datos se generan en el taller, donde los mecánicos proporcionan la información sobre las reparaciones realizadas, mientras que el personal administrativo suministra los precios de las reparaciones. Actualmente, solo existen las páginas de los proveedores como referencia para los precios de los repuestos.

En cuanto a los controles de seguridad, no se considera necesario implementar medidas adicionales, ya que no hay requerimientos específicos de seguridad en este contexto.

El producto será utilizado de manera continua, aunque habrá momentos de interrupción debido a que quien utilice el sistema también estará atendiendo a los clientes.

## 4. Ambiente operativo

### 4.1Clases 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.

### 4.2 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.

### 4.3 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.

# Requerimientos del sistema

## Funciones de software

## Función #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.

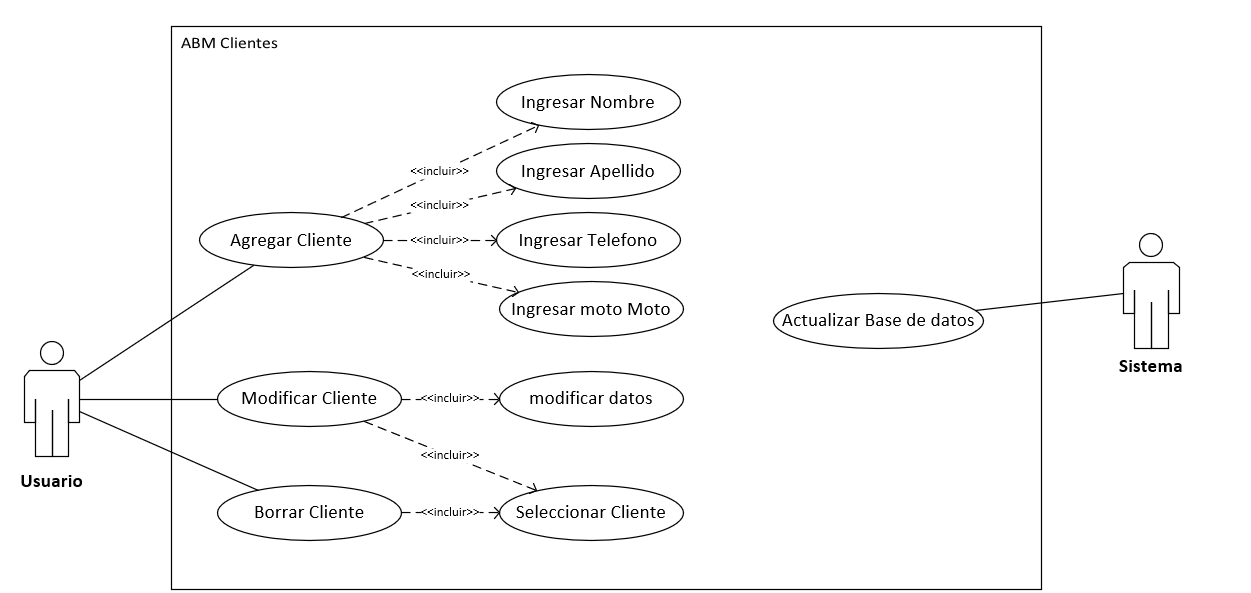
Se deben documentar adecuadamente los componentes del sistema y las funcionalidades implementadas para permitir una fácil comprensión y mantenimiento por parte del personal técnico.

**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.

## Casos de uso

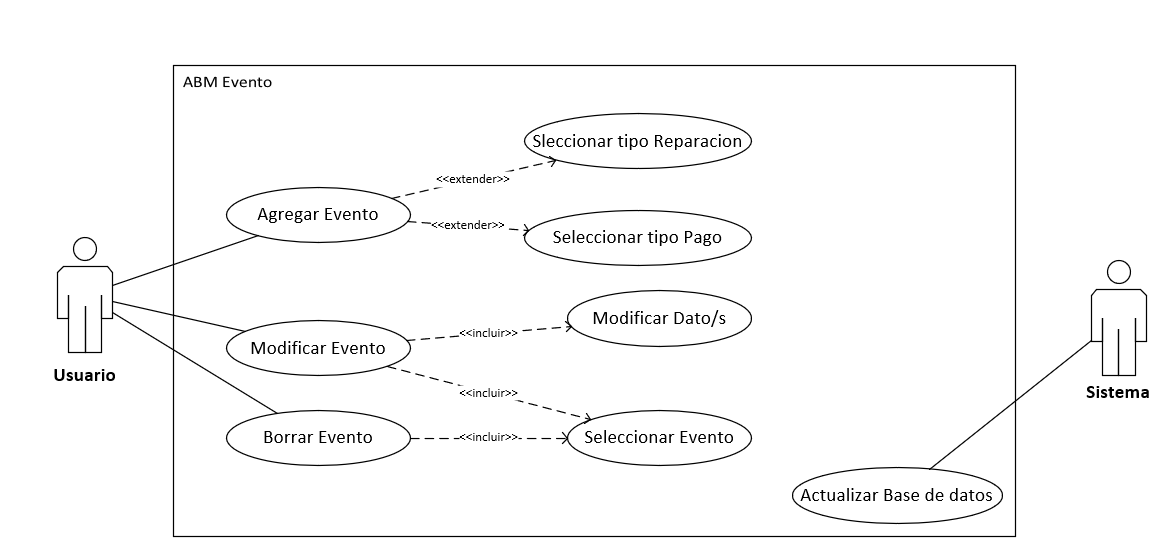
**3.1 ABM Clientes**

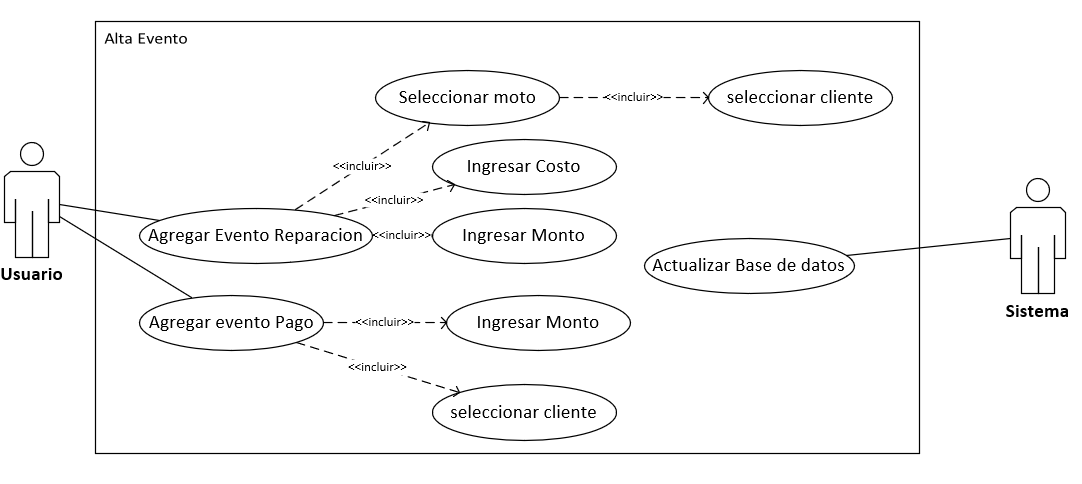
****

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 1 | | |
| **Nombre**: | ABM Clientes | | |
| **Autor**: | Hernán Maldonado | **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 |

**3.2 ABM Eventos**

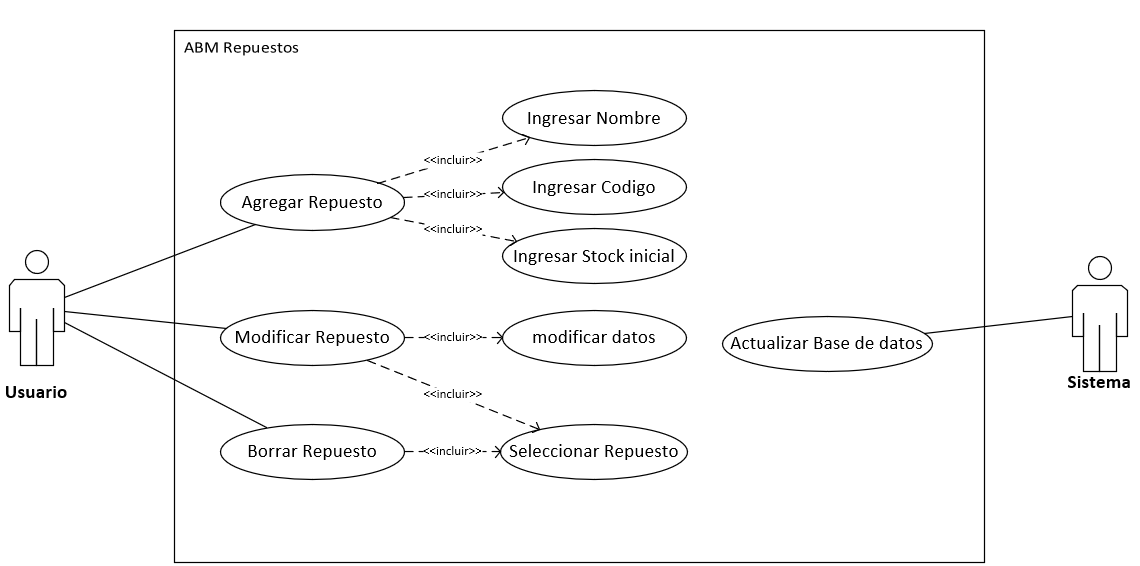
****

****

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 2 | | |
| **Nombre**: | ABM Evento | | |
| **Autor**: | Hernán Maldonado | **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 |

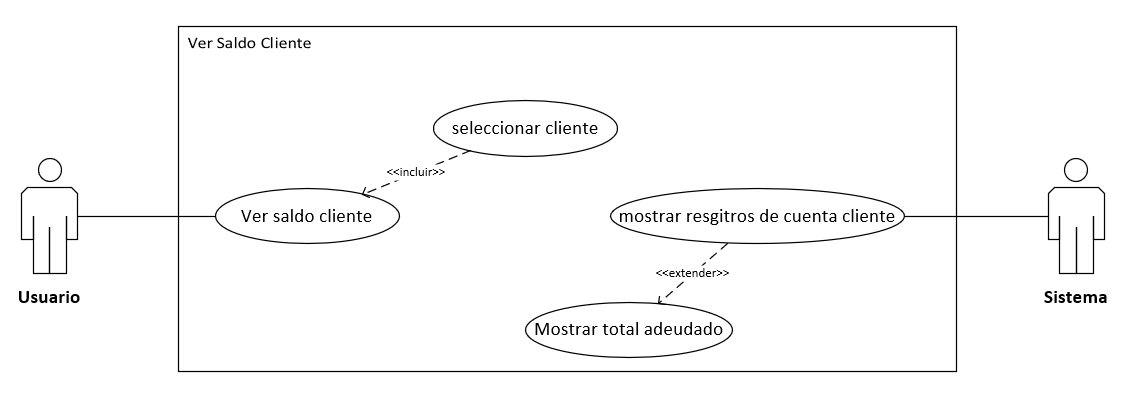
**3.3 ABM Repuestos**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 3 | | |
| **Nombre**: | ABM Repuestos | | |
| **Autor**: | Hernán Maldonado | **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 |

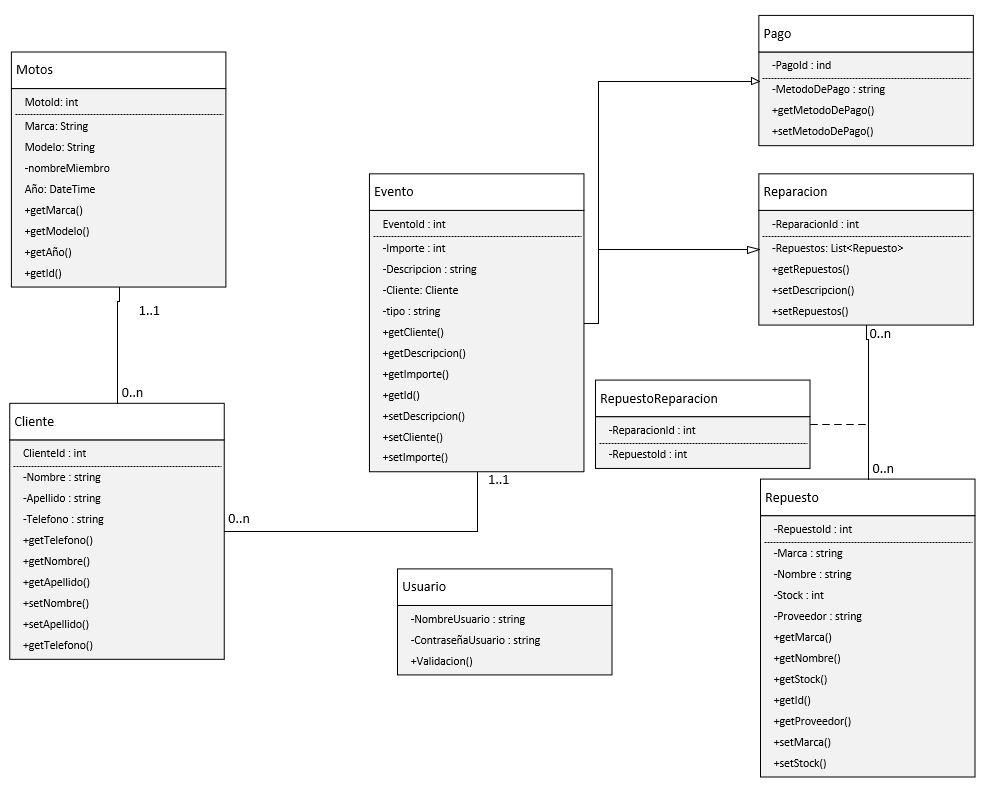
**3.4 Ver Estado de cuenta**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | 4 | | |
| **Nombre**: | Seguimiento saldo clientes | | |
| **Autor**: | Hernán Maldonado | **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 |

## Diagrama de clases

****