

UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA
CICLO II - 2023



CÁTEDRA:
INGENIERIA DE SOFTWARE

TEMA:
SISTEMA DE SERVICIOS AUTOMATRICES "SSA"

DOCENTE:
INGA. LESBIA MARÍA MANCÍA SANDOVAL

INTEGRANTES:

ANGEL IVÁN ROMERO HERNÁNDEZ	RH01121172
RAFAEL ALBINO JOVEL ALFARO	JA01135920
KEVIN OMAR DIAZ PINEDA	DP01135419
ADIEL ENOC PUENTES MONGE	PM01136619

SAN SALVADOR, 25 NOVIEMBRE 2023

Contenido

Contenido.....	2
Introducción.....	4
Objetivos.....	5
Objetivo General:.....	5
Objetivos Específicos:.....	5
Tema del proyecto.....	6
Descripción del proyecto.....	6
Justificación del proyecto.....	6
Listado de requerimientos.....	7
Requisitos Funcionales (RF):.....	7
Requisitos No Funcionales (RNF):.....	8
Desarrollo ágil.....	9
Organigrama.....	9
Planning pocket inicial.....	10
Requerimientos en formato SCRUM.....	10
Sprints de desarrollo.....	12
Tecnologías a ocupar para el desarrollo del proyecto.....	13
Diagrama de casos de uso y descripciones.....	15
Diagrama de clases.....	26
Establecer y justificar las propiedades de confiabilidad que deberá cumplir el software a desarrollar.....	27
Establecer y justificar las propiedades de fiabilidad y disponibilidad que deberá cumplir el software a desarrollar.....	28
Establecer y justificar las propiedades de seguridad que deberá cumplir el software a desarrollar.....	29
Listado de requerimientos que se ajustan para garantizar la confiabilidad y seguridad del software a desarrollar.....	30
Clasificación del riesgo del software a desarrollar.....	30
Listado de requerimientos de seguridad del software a desarrollar.....	31
Diseñar la arquitectura de protección en capas del software a desarrollar.....	31
Identificar las vulnerabilidades asociadas con las opciones de tecnología.....	32
Establecimiento y justificación del tipo de software.....	33
Características:.....	33
Ventajas:.....	33
Desventajas:.....	33

Justificación:	34
Establecer y describir los riesgos del proyecto a desarrollar	34
Establecer y describir estrategias para gestionar los riesgos del proyecto	34
Manual de Usuario.....	40
Fuente del sistema	40
Link de videotutorial de uso del programa.	40
Bibliografía	41
Anexos.....	41
a) Autoevaluaciones.....	41
b) Coevaluaciones	42

Introducción

El proyecto Sistema de Servicios Automotrices (SSA) cumple con el objetivo de la creación de una solución tecnológica integral diseñada específicamente para revolucionar la gestión de talleres automotrices. En un contexto en el que la industria de servicios automotrices es cada vez más competitiva, el SSA es una herramienta esencial para mejorar la eficiencia operativa y la calidad del servicio ofrecido.

Se han seleccionado tecnologías clave para el desarrollo de la solución, tomando en cuenta criterios como robustez, flexibilidad y escalabilidad. Además de una estrategia de gestión de recursos para desarrollo con la metodología ágil Scrum.

El Sistema de Servicios Automotrices (SSA) se encuentra en una posición estratégica para impactar positivamente en la industria de servicios automotrices al brindar herramientas eficaces para una gestión más eficiente y precisa del negocio automotriz.

Objetivos

Objetivo General:

Resolver la problemática de ineficiencia en la gestión procesos empresas de taller automotriz, mejorando así la dinámica operativa y la experiencia del cliente a través del Sistema de Servicios Automotrices (SSA) que opta por un mejor servicio automotriz en todas sus fases.

Objetivos Específicos:

Implementar mejoras específicas en el módulo de gestión de clientes del SSA para optimizar la recopilación, organización y acceso a la información del cliente.

Implementación de funcionalidades para la generación de informes detallados para proporcionar a los talleres automotrices una gestión financiera más eficiente y la capacidad de tomar decisiones informadas basadas en datos concretos.

Tema del proyecto

Sistema de Servicios Automotrices SSA

Descripción del proyecto

El proyecto Sistema de Servicios Automotrices (SSA) se enfoca en la creación de una solución tecnológica integral diseñada específicamente para revolucionar la gestión de un taller automotriz. En un contexto en el que la industria de servicios automotrices es cada vez más competitiva y demandante, SSA se presenta como una herramienta esencial para mejorar la eficiencia operativa y la calidad del servicio ofrecido.

Este proyecto tiene como objetivo central la concepción y desarrollo del SSA, un sistema altamente adaptable que atiende a las necesidades específicas de los talleres automotrices. El SSA permitirá una gestión eficiente de los clientes, la generación de facturas, el cálculo preciso de costos, la organización de citas y la generación de reportes detallados.

El núcleo del SSA se enfoca en la gestión de clientes, facilitando la captura de datos, el seguimiento de historiales de servicio y la programación de citas de manera eficaz. Además, el sistema automatizará la generación de facturas y el cálculo de costos, optimizando la gestión financiera del taller.

La capacidad de generar informes personalizados sobre el desempeño y las tendencias del taller será un elemento clave del SSA, permitiendo una toma de decisiones basada en datos concretos.

La implementación exitosa del SSA no solo mejorará la productividad y eficiencia del taller, sino que también repercutirá en la satisfacción del cliente al brindar un servicio más rápido y transparente. Los beneficios de este sistema se traducirán en un negocio más competitivo y una posición destacada en la industria de servicios automotrices.

Justificación del proyecto

En un entorno empresarial caracterizado por la creciente competencia y la demanda constante de eficiencia, la implementación de soluciones tecnológicas innovadoras se vuelve esencial para mantener y mejorar la competitividad de los talleres automotrices. La justificación de la creación del "Sistema de Servicios Automotrices (SSA)" se fundamenta en la necesidad crítica de abordar las limitaciones y desafíos actuales en la gestión de estos establecimientos, así como en aprovechar las oportunidades que ofrece la tecnología para optimizar su funcionamiento y calidad de servicio.

1. **Competitividad en la Industria Automotriz:** La industria de servicios automotrices es altamente competitiva y está en constante evolución. Los talleres automotrices enfrentan la presión de ofrecer un servicio eficiente y de alta calidad para retener a los clientes y competir con otros actores del mercado. El SSA se presenta como una solución estratégica para mejorar la eficiencia operativa y diferenciarse en un mercado cada vez más demandante.

2. **Mejora de la Eficiencia Operativa:** La gestión manual de clientes, facturación, cálculo de costos, citas y generación de informes suele ser propensa a errores y consume tiempo. El SSA se justifica al automatizar estas tareas, reduciendo la carga de trabajo administrativo y permitiendo que el personal del taller se concentre en brindar un servicio de mayor calidad y atención al cliente.
3. **Optimización de Recursos Financieros:** La generación automatizada de facturas y el cálculo preciso de costos contribuyen a una gestión financiera más eficiente, evitando pérdidas por errores en los registros financieros y garantizando una facturación precisa.
4. **Toma de Decisiones Basada en Datos:** La generación de informes detallados sobre el desempeño y las tendencias del taller proporciona datos valiosos para la toma de decisiones estratégicas. Esto permite una gestión más informada y la identificación de áreas de mejora que, de otro modo, podrían pasar desapercibidas.
5. **Satisfacción del Cliente:** La implementación del SSA tiene un impacto directo en la satisfacción del cliente al agilizar la programación de citas, reducir tiempos de espera y mejorar la comunicación. Los clientes experimentarán un servicio más rápido y transparente, lo que contribuirá a la retención de la clientela y la generación de recomendaciones.
6. **Posicionamiento Competitivo:** La adopción de tecnologías avanzadas, como el SSA, posicionará al taller automotriz como un negocio moderno y confiable en la mente de los clientes, lo que puede ser un factor clave para ganar cuota de mercado y mantener una ventaja competitiva.

Listado de requerimientos

Requisitos Funcionales (RF):

Gestión de Clientes:

El sistema permitirá el registro de nuevos clientes con información como nombre, dirección y número de contacto.

El sistema podrá buscar y visualizar detalles de clientes existentes.

El sistema proporcionará una opción para actualizar la información del cliente cuando sea necesario.

Creación de Facturas:

El sistema podrá generar una factura al seleccionar un cliente y el vehículo relacionado.

El sistema asignará automáticamente un número de factura único y la fecha de emisión.

El sistema podrá agregar servicios realizados a la factura, incluyendo una descripción detallada y el costo de cada servicio.

El sistema deberá enviar vía correo electrónico las facturas a los clientes y guardar una copia para el taller.

Cálculo de Costos:

El sistema calculará automáticamente el costo total de la factura basado en los servicios seleccionados.

Los impuestos y cargos adicionales se sumarán al total de la factura.

Gestión de Citas:

El sistema mantendrá un registro de servicios disponibles según espacio de citas diarias.

Después de agregar servicios a una factura, el sistema actualizará el calendario de citas diario.

Generación de Reportes:

El sistema generará informes mensuales de facturación que mostrarán el total de ingresos, servicios más solicitados.

El sistema se podrá personalizar los informes según el rango de fechas y otros filtros.

Notificaciones:

El sistema enviará notificaciones a los usuarios en la pantalla principal del mismo sobre las citas pendientes con un día de anticipación.

Requisitos No Funcionales (RNF):

Seguridad:

El sistema implementará autenticación solo para usuarios internos para garantizar un acceso seguro por medio de una encriptación hash MD5 a nivel de base de datos.

Rendimiento:

El sistema debe cargar páginas y generar facturas en menos de 5 segundos, incluso bajo cargas máximas.

Debe ser capaz de manejar al menos 100 facturas generadas por hora.

Usabilidad:

La interfaz de usuario será intuitiva y tendrá un diseño limpio para facilitar la navegación.

Se proporcionarán mensajes de error claros y orientación contextual para ayudar a los usuarios en el uso del sistema.

Escalabilidad:

El sistema debe ser capaz de admitir un aumento del 50% en el volumen de facturación en un período de seis meses sin degradación significativa del rendimiento.

Disponibilidad:

El sistema estará en línea y disponible para su acceso al menos el 99.9% del tiempo.

Compatibilidad:

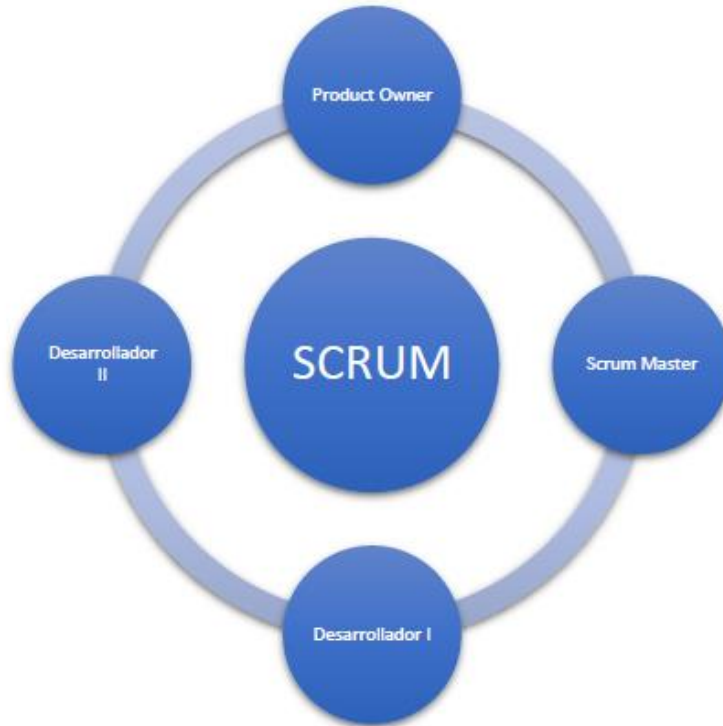
El sistema será compatible con los navegadores web más utilizados, como Chrome, Firefox y Safari.

Mantenibilidad:

El sistema estará diseñado con una arquitectura modular que permita la incorporación de nuevas características y actualizaciones sin afectar la funcionalidad existente.

Desarrollo ágil

Organigrama



PRODUCT OWNER – Adiel Fuentes
SCRUM MASTER – Rafael Jovel
DEVELOPER1 – Angel Romero
DEVELOPER2 – Kevin Diaz

Planning pocket inicial

Actividad	Product Owner	Scrum Master	Developer 1	Developer 2	TOTALES
Desarrollo de la funcionalidad de gestión de clientes	10	10	10	10	40
Implementación de la gestión de citas y actualización del	10	10	10	10	40
Desarrollo de la generación de informes y personalización	7	10	6	9	32
Desarrollo del cálculo de costos de facturas	6	9	6	5	26
Implementación de creación de facturas y asignación de números	8	8	5	5	26
Implementación del sistema de notificaciones	5	8	5	5	26
TOTAL	46	55	42	44	190

Requerimientos en formato SCRUM

Nombre de la tarea	Responsable	Fecha inicio	Fecha final	Días	Estado
HU1: Gestión de Clientes					
Gestión de Clientes:		16/09/2023	30/09/2023	15	
1.1 El sistema permitirá el registro de nuevos clientes con información como nombre, dirección y número de contacto.		16/09/2023	20/09/2023	5	pendiente
1.2 El sistema podrán buscar y visualizar detalles de clientes existentes.		21/09/2023	25/09/2023	5	pendiente
1.3 El sistema proporcionará una opción para actualizar la información del cliente cuando sea necesario.		25/09/2023	30/09/2023	5	pendiente
HU2: Creación de Facturas					
Creación de Facturas:		01/10/2023	15/10/2023	15	
2.1 El usuario podrá generar una factura al seleccionar un cliente y el vehículo relacionado.		1/10/2023	5/10/2023	5	pendiente
2.2 El sistema asignará automáticamente un número de factura único y la fecha de emisión.		6/10/2023	10/10/2023	5	pendiente
2.3 El usuario podrá agregar servicios realizados a la		11/10/2023	15/10/2023	5	pendiente

factura, incluyendo una descripción detallada y el costo de cada servicio.					
HU3: Cálculo de Costos					
Cálculo de Costos:		16/10/2023	28/10/2023	10	
3.1 El sistema calculará automáticamente el costo total de la factura basado en los servicios seleccionados.		16/10/2023	20/10/2023	5	pendiente
3.2 Los impuestos y cargos adicionales se sumarán al total de la factura.		20/10/2023	25/10/2023	5	pendiente
HU4: Gestión de Citas					
Gestión de Citas:		26/10/2023	04/11/2023	10	
4.1 El sistema mantendrá un registro de servicios disponibles según espacio de citas diarias.		26/10/2023	30/10/2023	5	pendiente
4.2 Después de agregar servicios a una factura, el sistema actualizará el calendario de citas diario.		31/10/2023	04/11/2023	5	pendiente
HU5: Generación de Reportes					
Generación de Reportes:		05/11/2023	15/11/2023	10	
5.1 El sistema generará informes mensuales de facturación que mostrarán el total de ingresos, servicios más solicitados y reporte de costos y ganancias		05/11/2023	09/11/2023	5	pendiente
5.2 Los usuarios podrán personalizar los informes según el rango de fechas y otros filtros.		10/11/2023	15/11/2023	5	pendiente
HU6:Notificaciones					
Notificaciones:		16/11/2023	25/11/2023	10	pendiente
6.1 El sistema enviará notificaciones a los usuarios en la pantalla principal sobre las citas pendientes con un día de anticipación.		16/11/2023	25/11/2023	10	pendiente

Sprints de desarrollo

Sprint 1 (Del 16/09/2023 al 30/09/2023)

Objetivo del Sprint: Desarrollar la funcionalidad básica de la gestión de clientes.

Tareas:

- 1.1 El sistema permitirá el registro de nuevos clientes con información como nombre, dirección y número de contacto.

Sprint 2 (Del 01/10/2023 al 15/10/2023)

Objetivo del Sprint: Implementar la creación de facturas y la asignación automática de números de factura.

Tareas:

- 2.1 El usuario podrá generar una factura al seleccionar un cliente y el vehículo relacionado.
- 2.2 El sistema asignará automáticamente un número de factura único y la fecha de emisión.

Sprint 3 (Del 16/10/2023 al 28/10/2023)

Objetivo del Sprint: Desarrollar el cálculo de costos de las facturas.

Tareas:

- 3.1 El sistema calculará automáticamente el costo total de la factura basado en los servicios seleccionados.
- 3.2 Los impuestos y cargos adicionales se sumarán al total de la factura.

Sprint 4 (Del 26/10/2023 al 04/11/2023)

Objetivo del Sprint: Implementar la gestión de citas y la actualización del calendario.

Tareas:

- 4.1 El sistema mantendrá un registro de servicios disponibles según espacio de citas diarias.
- 4.2 Después de agregar servicios a una factura, el sistema actualizará el calendario de citas diario.

Sprint 5 (Del 05/11/2023 al 15/11/2023)

Objetivo del Sprint: Desarrollar la generación de informes y la personalización de los mismos.

Tareas:

- 5.1 El sistema generará informes mensuales de facturación que mostrarán el total de ingresos, servicios más solicitados y reporte de costos y ganancias.
- 5.2 Los usuarios podrán personalizar los informes según el rango de fechas y otros filtros.

Sprint 6 (Del 16/11/2023 al 25/11/2023)

Objetivo del Sprint: Implementar el sistema de notificaciones.

Tareas:

6.1 El sistema enviará notificaciones a los usuarios en la pantalla principal sobre las citas pendientes con un día de anticipación.

Tecnologías a ocupar para el desarrollo del proyecto



PHP es un lenguaje de programación ampliamente utilizado en el desarrollo de aplicaciones web y sistemas de software debido a su versatilidad y facilidad de uso. Algunas de las razones por las que PHP es una elección común para el desarrollo de proyectos incluyen:

Facilidad de lectura y escritura: PHP destaca por su sintaxis clara y su facilidad de aprendizaje. Esto permite a los desarrolladores escribir código de manera rápida y colaborar eficientemente.

Gran comunidad y abundancia de bibliotecas: PHP cuenta con una comunidad activa de desarrolladores que contribuyen con bibliotecas y frameworks que simplifican tareas comunes, acelerando el desarrollo de aplicaciones.

Soporte multiplataforma: PHP es compatible con diversas plataformas, permitiendo la creación de aplicaciones que pueden ejecutarse en diferentes sistemas operativos.

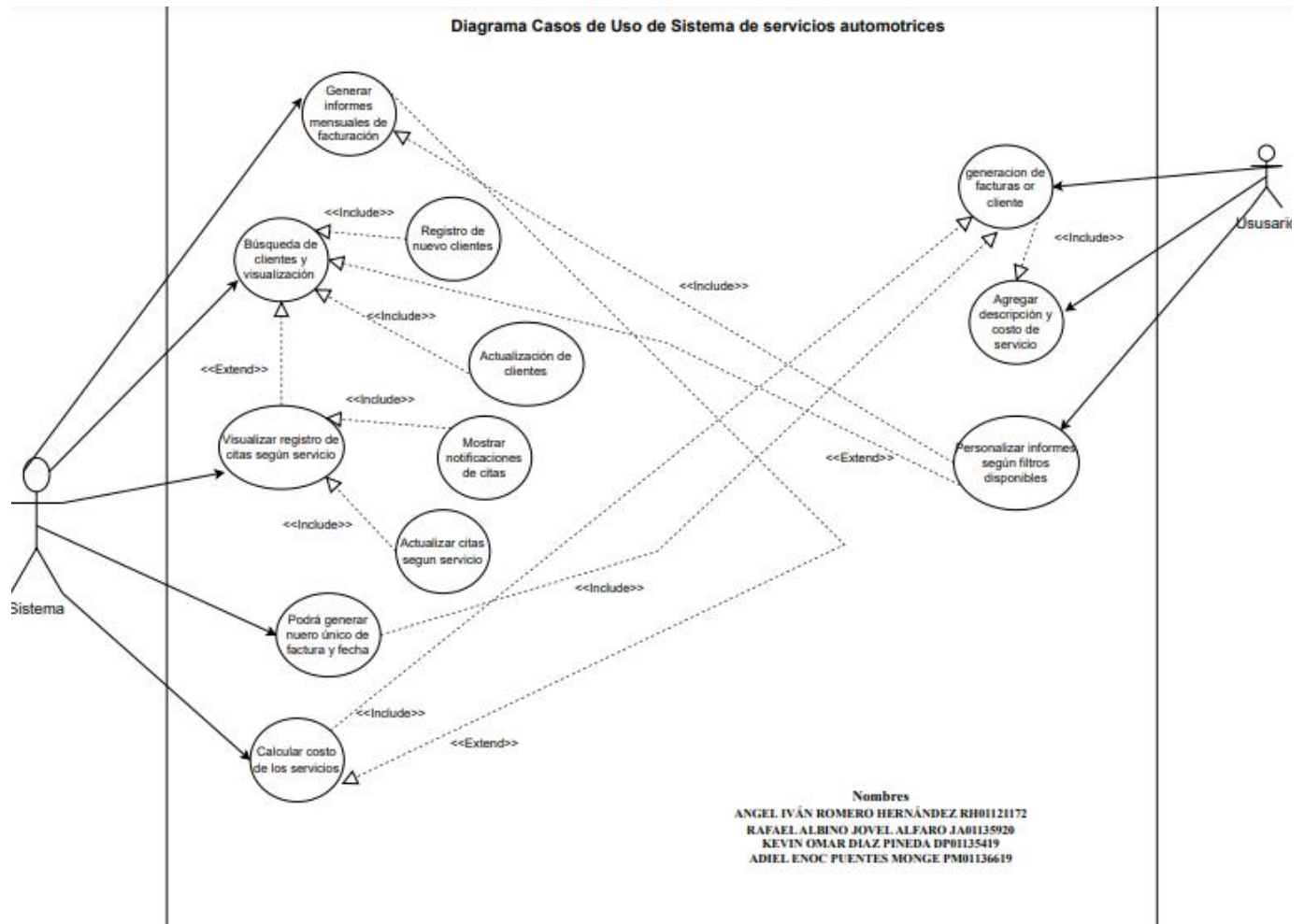
Enfoque en la eficiencia: A pesar de su simplicidad, PHP es un lenguaje eficiente y puede ser utilizado en una amplia gama de aplicaciones, incluyendo aquellas relacionadas con bases de datos como MySQL.



En cuanto a MySQL, es un sistema de gestión de bases de datos relacionales ampliamente utilizado en aplicaciones web y empresariales. Algunas razones para elegir MySQL como base de datos en el proyecto incluyen:

- **Fiabilidad:** MySQL es conocido por su confiabilidad y rendimiento en entornos de producción, lo que lo hace adecuado para sistemas críticos.
- **Facilidad de uso:** MySQL ofrece una administración sencilla y una sintaxis SQL estándar que facilita el trabajo con bases de datos.
- **Escalabilidad:** MySQL es escalable, lo que significa que puede manejar grandes cantidades de datos y crecer con el proyecto a medida que sea necesario.
- **Soporte de la comunidad:** Al igual que php, MySQL tiene una comunidad activa que proporciona soporte, documentación y recursos valiosos.

Diagrama de casos de uso y descripciones.



Caso de uso	Búsqueda de clientes y visualización
Objetivo	El sistema se encargará de la búsqueda de los clientes registrados y los mostrará en sistema
Actor principal	sistema.
Personal involucrado	usuario.
Precondición	Ingresa un usuario y contraseña.
Garantías de éxito	Podrá imprimir su factura o ticket
Escenario principal	<ol style="list-style-type: none"> 1. Logearse 2. Ingresar al menú de opciones de servicios. 3. Seleccionar un servicio 4. Generar y mostrar cita.
Flujos alternativos	<ol style="list-style-type: none"> 1. Que no le permita registrarse. 2. Que no le genere cita al usuario 3. No le permita generar cita.
Requisitos especiales	Que la cita generada no excede límites de tiempo.
Frecuencia	Al generar la cita la visualización regrese a la pantalla principal.

Caso de uso	Visualizar registro de citas según servicios
Objetivo	Permita al usuario visualizar sus citas según opciones o tipos de servicios.
Actor principal	Cliente o usuario
Personal involucrado	Usuario y empleado.
Precondición	Dar clic en la cita para ver fecha y hora de cita.
Garantías de éxito	Imprimir la cita si el usuario lo desea y poder realizar una segunda cita.
Escenario principal	<ol style="list-style-type: none"> 1 Opciones de tipos de servicio. 2 Opciones de fechas de citas 3 Opciones de hora de cita 4 Confirmar programaciones de citas
Flujos alternativos	<ol style="list-style-type: none"> 1 la cita no se registra 2 No aparece hora o fecha 3 Muestra fecha errónea. 4 que no permita visualizar si se generó la cita o no
Requisitos especiales	El tiempo que tarda en visualizar la cita generada o la confirmación de la cita.
Frecuencia	Cada vez que el usuario o empleado desee ver las citas que hay registradas.

Caso de uso	Calcular costo de los servicios
Objetivo	Realizar un cálculo del costo del servicio seleccionado y mostrarlo en pantalla
Actor principal	Sistema.

Personal involucrado	El cliente o usuario.
Precondición	<ol style="list-style-type: none"> 1 Seleccionar tipo de servicio. 2 Introducir datos del usuario.
Garantías de éxito	<ol style="list-style-type: none"> 1 Mostrar en pantalla costo del servicio seleccionado. 2 Que imprima un ticket con el costo del servicio.
Escenario principal	<ol style="list-style-type: none"> 1 Que el usuario navegue en opciones de servicios en oferta con costos más bajos 2 Seleccionar el servicio. 3 Opciones de pago del costo del servicio.
Flujos alternativos	<ol style="list-style-type: none"> 1 Si no le parece el costo del servicio
Requisitos especiales	El usuario tiene la opción de salirse de las opciones si el costo no le parece justo
Frecuencia	En cuanto el cliente desee ver los costos de los servicios disponibles

Caso de uso	Podrá generar número único de factura y fecha
Objetivo	Generar un numero único en cada factura y que lleve fecha en la que se creó la factura
Actor principal	Sistema.
Personal involucrado	El cliente o usuario a quien se le brinda un servicio
Precondición	ber cancelado el costo del servicio.
Garantías de éxito	<ol style="list-style-type: none"> 1. Que muestre en pantalla o visualice la factura generada 2. Imprimir y entregar la factura al usuario
Escenario principal	usuario tiene a la vista la factura generada
Flujos alternativos	Mostrar un mensaje con Error si no se pudiera generar la factura
Requisitos especiales	El usuario no desee que se le imprima factura
Frecuencia	Cada vez que se pague un servicio

Caso de uso	Generar Informes Mensuales de Facturación
Objetivo	Llevar un registro o control y estadística de facturación mensual
Actor principal	Sistema.
Personal involucrado	Personal del taller o Cliente
Precondición	ar fecha de cada mes que se quiera el

	informe, por ejemplo, puede ser cada ultimo día de mes o cada 15 de mes
Garantías de éxito	<ol style="list-style-type: none"> 1. Tener el registro de toda la facturación cada mes. 2. llevar el control de todas las facturaciones por mes
Escenario principal	personal o empleado encargado del control visualiza los informes mensuales de facturación
Flujos alternativos	el personal o empleados no ha estado generando o registrando las facturas en ese mes
Requisitos especiales	Que no se hayan generado facturas en ese mes
Frecuencia	Cada mes

Caso de uso	Registro de nuevo cliente
Objetivo	Registrar la información de un nuevo cliente en el sistema de servicios automotrices para futuras interacciones y seguimiento.
Actor principal	Sistema y usuario autorizado
Personal involucrado	Personal empleado y Cliente
Precondición	no haya un registro anterior del cliente nuevo.
Garantías de éxito	<ol style="list-style-type: none"> 1. Se ha creado un nuevo registro de cliente en la base de datos del sistema con la información proporcionada. 2. El nuevo cliente ahora está registrado y disponible en el sistema para futuras interacciones y seguimiento.
Escenario principal	sistema mostrando al empleado autorizado todas las opciones o campos a validar la información del nuevo cliente.
Flujos alternativos	el sistema detecta información duplicada o conflicto con registros existentes, muestra un mensaje de error y solicita al empleado revisar y corregir la información.
Requisitos especiales	Que el usuario no desee ser agregado o no quiera dar su información
Frecuencia	Cada que hay un cliente nuevo

Caso de uso	Actualización de cliente
Objetivo	Actualizar la información de un cliente existente en el sistema de servicios automotrices para reflejar cambios en los

	datos del cliente.
Actor principal	Empleado o usuario autorizado
Personal involucrado	Sistema y Cliente
Precondición	El cliente ya este registrado en el sistema y solo se desee hacer una actualización de algún dato o inflo del cliente
Garantías de éxito	<ol style="list-style-type: none"> 1. Que muestre en pantalla que se ha realizado la actualización de nueva información del cliente 2. La información del cliente se actualiza en la base de datos del sistema, reflejando los cambios realizados. 3. La información actualizada del cliente está disponible en el sistema para futuras interacciones y seguimiento.
Escenario principal	cliente o personal autorizado interactúa con el sistema colocando la nueva información en los campos.
Flujos alternativos	el sistema detecta algún conflicto o error en los datos actualizados, muestra un mensaje de error y solicita al empleado revisar y corregir la información.
Requisitos especiales	La actualización de la información del cliente puede ser necesaria cuando el cliente cambia de dirección, número de teléfono o realiza modificaciones en su vehículo. También es útil para mantener actualizados los datos de contacto para enviar recordatorios de citas y promociones.
Frecuencia	Cada vez que un cliente desee actualizar sus datos en el sistema porque haya cambiado o modificado alguna información personal como numero de teléfono, por ejemplo.

Caso de uso	Mostrar Notificaciones de citas
Objetivo	Proporcionar a los empleados o usuarios información relevante sobre las citas programadas para garantizar una gestión eficiente de las agendas y servicios.
Actor principal	sistema.
Personal involucrado	Empleado o usuario autorizado.
Precondición	Haber realizado una cita previa.
Garantías de éxito	Que tanto el usuario como empleado pueda visualizar las citas existentes con su fecha, hora y a nombre de quien esta la cita.
Escenario principal	El empleado inicia sesión en el sistema de servicios automotrices con las credenciales adecuadas.

	<ol style="list-style-type: none"> 1. El sistema verifica la fecha actual y busca citas programadas para el día o período de tiempo relevante. 2. El sistema muestra una lista de notificaciones de citas pendientes en la interfaz principal del empleado. 3. Cada notificación de cita puede incluir información como: <ul style="list-style-type: none"> • Nombre del cliente. • Fecha y hora de la cita. • Tipo de servicio programado. • Detalles adicionales de la cita (si los hay). • Nombre y número de contacto del cliente. 4. El empleado revisa las notificaciones de citas pendientes y toma las acciones apropiadas según sea necesario, como preparar el equipo y materiales necesarios, confirmar la cita con el cliente o asignarla a un técnico específico. 5. El sistema marca las notificaciones de citas como leídas o procesadas una vez que el empleado las ha revisado.
Flujos alternativos	no hay citas programadas para el día o período de tiempo relevante, el sistema muestra un mensaje indicando que no hay notificaciones de citas pendientes.
Requisitos especiales	Que la cita generada no excede límites de tiempo.
Frecuencia	Cada vez que haya citas deben ser notificadas

Caso de uso	Actualizar citas según servicios
Objetivo	Modificar o ajustar las citas programadas para acomodar los servicios específicos solicitados por los clientes y garantizar una gestión eficiente de los recursos y el tiempo.
Actor principal	Empleado o usuario autorizado.
Personal involucrado	cliente.
Precondición	Tener ya una cita y que al actualizarla esta no coincida en horario de otra cita
Garantías de éxito	Las citas se han actualizado de acuerdo con los servicios solicitados por los clientes.

	<ul style="list-style-type: none"> • La información actualizada de las citas se refleja en el sistema y está disponible para su posterior seguimiento y ejecución
Escenario principal	<p>El empleado inicia sesión en el sistema de servicios automotrices con las credenciales adecuadas.</p> <ul style="list-style-type: none"> • El empleado accede a la lista de citas programadas para el día o período de tiempo relevante. • El sistema muestra las citas programadas, incluyendo detalles como: <ul style="list-style-type: none"> • Nombre del cliente. • Fecha y hora de la cita. • Tipo de servicio originalmente programado. • Detalles adicionales de la cita. • El empleado identifica una cita que requiere cambios en los servicios solicitados por el cliente. • El empleado selecciona la cita específica para su modificación. • El sistema permite al empleado realizar las siguientes acciones: <ul style="list-style-type: none"> • Cambiar el tipo de servicio programado. • Modificar la fecha y hora de la cita (si es necesario). • Asignar el servicio a un técnico o especialista adecuado. • Actualizar los detalles de la cita según las nuevas necesidades del cliente. • El empleado guarda los cambios realizados en la cita. • El sistema verifica que los cambios se hayan realizado con éxito y actualiza la información de la cita en la base de datos
Flujos alternativos	<p>el empleado intenta realizar cambios que generen conflictos de horario o recursos</p>

	insuficientes, el sistema mostrará un mensaje de error y solicitará que se realicen ajustes adecuados
Requisitos especiales	Este caso de uso es fundamental para asegurar que las citas se ajusten adecuadamente a las necesidades de los clientes y que los recursos estén disponibles para realizar los servicios programados.
Frecuencia	Cada vez que el cliente tenga la necesidad de actualizar la cita ya sea para quitar o agregar un servicio o simplemente cambiar fecha u hora

Caso de uso	Agregar descripción y costo de servicio
Objetivo	Ingresar o actualizar la información relacionada con la descripción y el costo de un servicio específico ofrecido por el taller.
Actor principal	Administrador del sistema
Personal involucrado	Usuario y empleado.
Precondición	El administrador del sistema debe haber iniciado sesión en el sistema. Debe existir al menos un servicio registrado en la lista actual de servicios ofrecidos.
Garantías de éxito	El nuevo servicio con su descripción y costo ha sido agrado correctamente a la lista de servicios ofrecidos. La descripción y el costo del servicio se han agregado o actualizado en la base de datos del sistema. <ul style="list-style-type: none"> • La información actualizada del servicio está disponible en el sistema y se utilizará para cotizaciones y comunicación con los clientes.
Escenario principal	El empleado inicia sesión en el sistema de servicios automotrices con las credenciales adecuadas. <ol style="list-style-type: none"> 1. El empleado accede a la sección de administración o gestión de servicios dentro del sistema. 2. El sistema muestra una lista de servicios existentes o una opción para agregar un nuevo servicio. 3. Si el empleado desea agregar un nuevo servicio: <ul style="list-style-type: none"> • El empleado selecciona la opción "Agregar nuevo servicio".

	<ul style="list-style-type: none"> • El sistema presenta un formulario vacío para ingresar la información del nuevo servicio. • El empleado completa los campos requeridos, que pueden incluir: <ul style="list-style-type: none"> • Nombre del servicio. • Descripción detallada del servicio. • Duración estimada del servicio. • Costo base del servicio. <p>4. Si el empleado desea actualizar un servicio existente:</p> <ul style="list-style-type: none"> • El empleado selecciona el servicio específico que desea modificar de la lista existente. • El sistema muestra la información actual del servicio en un formulario editable. • El empleado actualiza los campos necesarios, como la descripción o el costo del servicio. <p>5. El empleado verifica que la información ingresada o actualizada sea precisa.</p> <p>6. Una vez que se completa el formulario de ingreso o actualización, el empleado selecciona la opción "Guardar" o "Actualizar" para almacenar los cambios en la base de datos del sistema.</p> <p>7. El sistema verifica que los cambios se hayan realizado con éxito y actualiza la información del servicio.</p>
Flujos alternativos	Si el sistema detecta errores o información faltante en el formulario, mostrará un mensaje de error y solicitará al empleado que revise y corrija los campos correspondientes.
Requisitos especiales	Puede ser necesario que los cambios en la

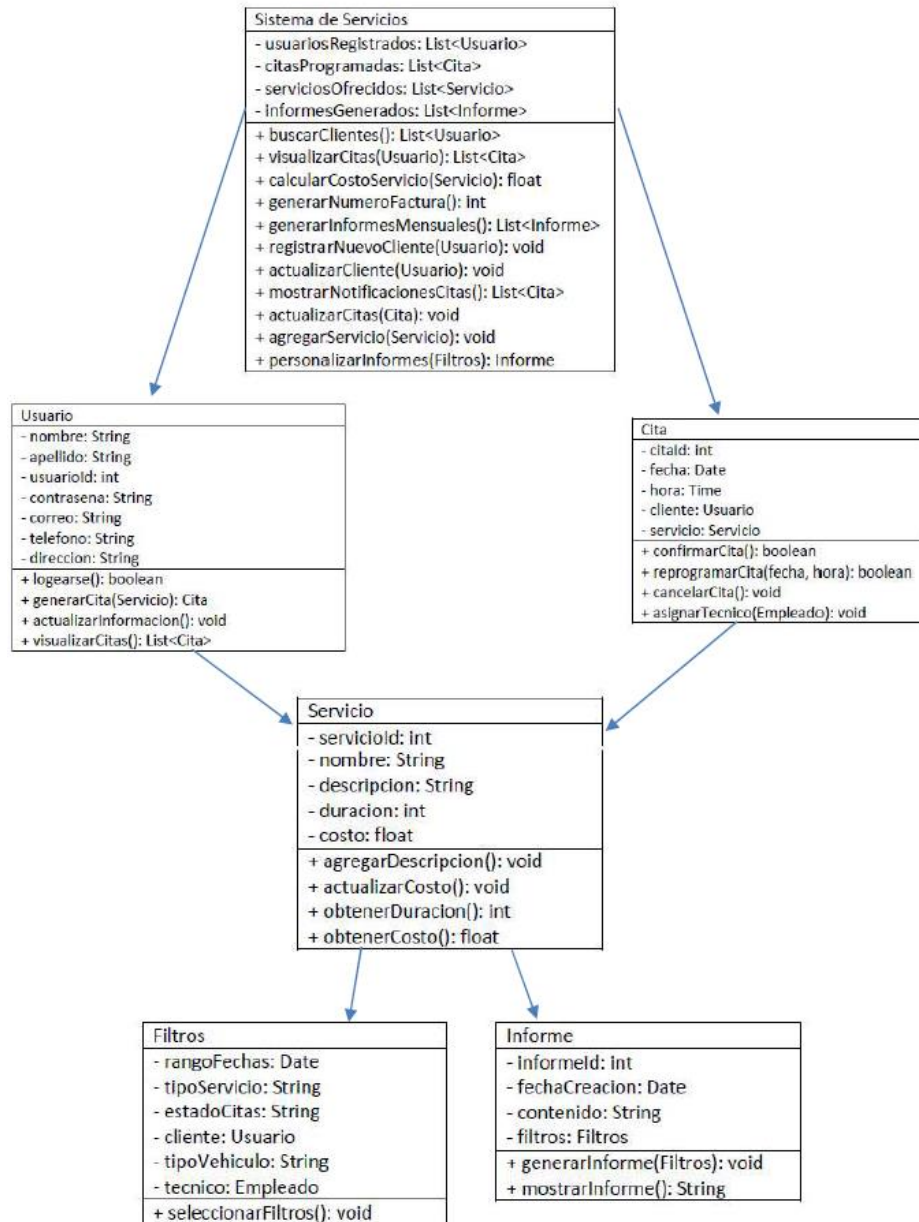
	descripción o el costo del servicio sean aprobados por un supervisor o gerente antes de que se reflejen en el sistema.
Frecuencia	Cada vez que se agrega a la lista un nuevo servicio

Caso de uso	Personalizar Informes según filtros disponibles
Objetivo	Permitir al usuario crear informes personalizados utilizando filtros disponibles para obtener información específica de la base de datos del sistema
Actor principal	usuario autorizado (por ejemplo, gerente, supervisor o empleado con permisos de generación de informes).
Personal involucrado	Personal del taller o Cliente
Precondición	ben existir informes disponibles en la lista de informes del sistema.
Garantías de éxito	usuario obtiene un informe personalizado basado en los filtros seleccionados y puede utilizarlo para el análisis de datos, la toma de decisiones o cualquier otro propósito específico.
Escenario principal	<p>El usuario inicia sesión en el sistema de servicios automotrices con las credenciales adecuadas.</p> <ol style="list-style-type: none"> 1. El usuario accede a la sección de generación de informes o informes personalizados dentro del sistema. 2. El sistema presenta una interfaz que permite al usuario seleccionar y aplicar filtros para definir los parámetros del informe. 3. Los filtros disponibles pueden incluir opciones como: <ul style="list-style-type: none"> • Rango de fechas. • Tipo de servicio. • Estado de las citas (pendientes, completadas, canceladas, etc.). • Cliente específico. • Tipo de vehículo. • Técnico asignado. • Otros criterios relevantes. 4. El usuario selecciona los filtros que desea aplicar y configura sus valores según sus necesidades específicas.

	<ol style="list-style-type: none"> 5. Una vez que se han seleccionado y configurado los filtros, el usuario solicita la generación del informe seleccionando la opción "Generar informe" o similar. 6. El sistema utiliza los filtros seleccionados para consultar la base de datos y recopilar la información relevante. 7. El sistema genera el informe personalizado basado en los filtros aplicados, que puede incluir gráficos, tablas o datos detallados, según la preferencia del usuario. 8. El informe generado se muestra en pantalla o se descarga en un formato de archivo adecuado (por ejemplo, PDF, Excel).
Flujos alternativos	no se encuentran datos que coincidan con los filtros seleccionados, el sistema informa al usuario que no se ha encontrado información y sugiere ajustar los filtros.
Requisitos especiales	Debe ser un empleado o personal autorizado que tenga acceso a la opción de personalizar informes. El usuario obtiene un informe personalizado basado en los filtros seleccionados y puede utilizarlo para el análisis de datos, la toma de decisiones o cualquier otro propósito específico.
Frecuencia	Cada mes o cada que el administrador del sistema los requiera.

Diagrama de clases.

Diagrama de clases para los casos de uso que has proporcionado en el contexto del proyecto Sistema de Servicios Automotrices (SSA):



Establecer y justificar las propiedades de confiabilidad que deberá cumplir el software a desarrollar.

- **Disponibilidad Alta:** El software debe estar disponible y funcionando la mayor parte del tiempo, idealmente 24/7. Esto es crucial en aplicaciones críticas donde la interrupción podría causar pérdidas significativas.
- **Tolerancia a Fallos:** El software debe ser capaz de detectar, reportar y, en la medida de lo posible, recuperarse de los fallos sin una interrupción importante en el servicio.
- **Resistencia a Cargas Elevadas:** El software debe ser capaz de manejar una alta carga de usuarios o transacciones sin degradación significativa del rendimiento o la confiabilidad.
- **Mantenibilidad:** El software debe ser fácil de mantener y actualizar, lo que incluye la corrección de errores, la adición de nuevas características y la aplicación de parches de seguridad.
- **Seguridad:** El software debe ser resistente a ataques y vulnerabilidades conocidas. Debe proteger la integridad y la confidencialidad de los datos y garantizar que solo los usuarios autorizados tengan acceso.
- **Reversibilidad:** El software debe permitir deshacer acciones, restaurar configuraciones anteriores y recuperar datos en caso de errores o pérdida de información.
- **Documentación Clara:** El software debe estar bien documentado, incluyendo manuales de usuario, guías de administración y documentación técnica. La documentación clara facilita la comprensión y el uso efectivo del software.
- **Pruebas Rigurosas:** El software debe someterse a pruebas exhaustivas, que incluyen pruebas de unidad, integración, regresión y pruebas de carga, para garantizar su funcionalidad y confiabilidad.
- **Respaldo y Recuperación de Datos:** El software debe contar con un sistema de respaldo y recuperación de datos eficaz para proteger los datos críticos en caso de pérdida o corrupción.
- **Auditorías y Registro de Eventos:** El software debe ser capaz de realizar auditorías y mantener un registro de eventos que permita rastrear y analizar lo que sucede en el sistema.

Establecer y justificar las propiedades de fiabilidad y disponibilidad que deberá cumplir el software a desarrollar.

Requerimientos de Confianza:

- Disponibilidad: El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, con un tiempo de inactividad mínimo planificado para actualizaciones y mantenimiento programado.
- Respuesta Rápida: El software debe responder de manera rápida a las solicitudes de los usuarios para evitar la pérdida de tiempo y la frustración del usuario.
- Respaldo de Datos: Se debe realizar una copia de seguridad regular de los datos críticos del sistema para garantizar la recuperación en caso de fallos o pérdida de datos.
- Mantenimiento y Actualizaciones: Debe haber un plan de mantenimiento regular y actualizaciones del software para solucionar errores, mejorar la seguridad y agregar nuevas características.

Requerimientos de Seguridad:

- Autenticación y Autorización: El sistema debe implementar un sistema de autenticación sólida para garantizar que solo usuarios autorizados puedan acceder a funciones críticas. Se deben definir roles y permisos apropiados.
- Protección de Datos Sensibles: Los datos sensibles, como la información de los clientes y las facturas, deben estar protegidos mediante encriptación en reposo y en tránsito.
- Prevención de Ataques: El software debe implementar medidas de seguridad para prevenir ataques comunes, como inyección SQL, Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF).
- Auditoría de Acceso: Se debe llevar un registro de las acciones realizadas por los usuarios en el sistema, lo que permitirá la detección de actividades sospechosas o maliciosas.

- **Actualización de Software:** Todos los componentes de software, incluidas las bibliotecas y los frameworks, deben mantenerse actualizados para mitigar vulnerabilidades conocidas.
- **Pruebas de Seguridad:** Antes del lanzamiento y después de cada sprint, se deben realizar pruebas de seguridad exhaustivas para identificar y solucionar posibles vulnerabilidades.
- **Cumplimiento Normativo:** El software debe cumplir con las regulaciones de protección de datos y privacidad de la empresa.
- **Control de Acceso a Bases de Datos:** Se debe implementar un control estricto de acceso a las bases de datos para evitar accesos no autorizados.
- **Seguridad en la Comunicación:** Se debe utilizar HTTPS y otros protocolos seguros para la comunicación entre el cliente y el servidor.
- **Recuperación de Desastres:** El sistema debe tener un plan de recuperación de desastres en caso de fallas catastróficas, como incendios o cortes de energía.

Establecer y justificar las propiedades de seguridad que deberá cumplir el software a desarrollar.

1. **Autenticación y Autorización Robustas:** El software debe contar con un sistema sólido de autenticación para verificar la identidad de los usuarios y asegurar que solo las personas autorizadas tengan acceso. Además, debe gestionar eficazmente la autorización para determinar qué acciones y datos están disponibles para cada usuario.
2. **Cifrado de Datos:** El software debe utilizar cifrado para proteger la confidencialidad de los datos almacenados y transmitidos. Debe implementar protocolos de cifrado seguros para garantizar que la información sensible no sea accesible para terceros no autorizados.
3. **Detección de Intrusiones y Monitorización:** El software debe ser capaz de detectar intentos de intrusión y actividades sospechosas. Además, debe contar con un sistema de monitorización que registre eventos relevantes y permita tomar medidas ante amenazas de seguridad.
4. **Gestión de Vulnerabilidades y Parches:** El software debe mantenerse actualizado y contar con un proceso eficaz de gestión de vulnerabilidades. Debe ser capaz de aplicar parches de seguridad de manera oportuna para corregir vulnerabilidades conocidas.

5. **Protección contra Ataques Comunes:** El software debe implementar defensas contra ataques comunes, como inyección de SQL, cross-site scripting (XSS), y otros ataques de seguridad conocidos.

Listado de requerimientos que se ajustan para garantizar la confiabilidad y seguridad del software a desarrollar.

1. **Requerimiento de Autenticación Robusta:** El sistema debe exigir autenticación segura para garantizar que solo usuarios autorizados tengan acceso.
2. **Requerimiento de Cifrado:** Todos los datos sensibles deben ser cifrados tanto en almacenamiento como en transmisión.
3. **Requerimiento de Detección de Intrusiones:** El software debe detectar y notificar actividades sospechosas o intentos de intrusión.
4. **Requerimiento de Actualización de Parches:** El sistema debe aplicar parches de seguridad de forma regular y oportuna.
5. **Requerimiento de Protección contra Ataques:** El software debe protegerse contra ataques comunes, como inyección de SQL y XSS.
6. **Requerimiento de Contraseñas Fuertes:** Se requieren políticas de contraseñas sólidas y caducidad de contraseñas.
7. **Requerimiento de Registro de Acceso:** El sistema debe mantener un registro de acceso y permitir auditoría.

Clasificación del riesgo del software a desarrollar.

Clasificación del Riesgo del Sistema de Servicios Automotrices (SSA): Alto.

Justificación:

1. **Datos Sensibles de Clientes:** El SSA gestionará información confidencial de los clientes, incluyendo detalles de contacto, datos del vehículo y posiblemente información de pago. La exposición o pérdida de estos datos puede tener graves repercusiones y, por lo tanto, aumenta el nivel de riesgo.
2. **Requisitos de Seguridad Rigurosos:** Para garantizar la integridad de los datos y la operación segura del sistema, se requiere la implementación de medidas de seguridad sólidas, como la autenticación de usuarios, el cifrado de datos y la protección contra ataques cibernéticos. Estos requisitos complejos aumentan el nivel de riesgo.
3. **Impacto en la Seguridad de los Vehículos:** El SSA puede estar conectado a sistemas de vehículos, lo que introduce un riesgo adicional. Un compromiso de la seguridad del

sistema podría tener consecuencias directas en la seguridad de los vehículos y sus ocupantes.

Listado de requerimientos de seguridad del software a desarrollar.

1. **Requisitos de Autenticación Fortalecida:** Implementar un sistema de autenticación sólido que incluya autenticación de dos factores para garantizar que solo usuarios autorizados tengan acceso al sistema.
2. **Cifrado de Datos Completo:** Aplicar cifrado de extremo a extremo para proteger la confidencialidad de todos los datos, tanto en reposo como en tránsito.
3. **Protección contra Ataques Comunes:** Implementar defensas contra ataques de seguridad comunes, como inyección de SQL, cross-site scripting (XSS) y ataques de fuerza bruta.

Diseñar la arquitectura de protección en capas del software a desarrollar.

1. Capa de Interfaz de Usuario (UI):z

- La interfaz de usuario puede ser un punto de entrada para ataques de ingeniería social, donde los usuarios son engañados para revelar información confidencial.
- Deben implementarse medidas de seguridad para proteger las credenciales de inicio de sesión y garantizar que las interfaces sean resistentes a ataques de suplantación.

2. Capa de Aplicación:

- Aquí se ejecutan las funciones y lógica del software. Los ataques de inyección (como SQL injection) pueden comprometer la seguridad de la aplicación si no se validan y filtran adecuadamente las entradas de usuario.
- Deben aplicarse controles de acceso adecuados para evitar la explotación de funciones por parte de usuarios no autorizados.
- Se deben implementar protecciones contra ataques de Cross-Site Scripting (XSS) que podrían inyectar código malicioso en la interfaz de usuario.

3. Capa de Datos:

- Esta capa almacena información confidencial de clientes y registros de citas. La seguridad de la base de datos es esencial para proteger estos datos.
- Deben aplicarse prácticas de cifrado sólidas para proteger los datos almacenados en la base de datos.
- La gestión de vulnerabilidades y parches es crítica para evitar posibles brechas de seguridad en la capa de datos.

4. Capa de Comunicación:

- Si el sistema se conecta a sistemas de vehículos o realiza transmisiones de datos a través de redes, la capa de comunicación es un punto importante para la seguridad.
- Deben implementarse protocolos de comunicación seguros para proteger la integridad y confidencialidad de los datos transmitidos.
- Se debe proteger contra ataques de intermediarios y garantizar la autenticación adecuada en las comunicaciones.

Identificar las vulnerabilidades asociadas con las opciones de tecnología.

Vulnerabilidades Asociadas a PHP:

Inyección de Código (Injection): Las aplicaciones PHP son susceptibles a ataques de inyección, como la inyección SQL y la inyección de comandos. Es crucial validar y sanitizar las entradas de usuario adecuadamente para prevenir estos ataques. El uso de consultas preparadas o declaraciones vinculadas puede ser una buena práctica para evitar la inyección SQL.

Cross-Site Scripting (XSS): Las aplicaciones web escritas en PHP pueden ser vulnerables a ataques XSS si no se filtran ni escapan correctamente las salidas HTML. Se debe tener cuidado al manejar datos de entrada no confiables y utilizar funciones de escape apropiadas, como `htmlspecialchars()`, para evitar la ejecución de scripts maliciosos en el navegador del usuario.

Gestión de Sesiones Insegura: La implementación de una gestión de sesiones segura es esencial para prevenir ataques de sesión, como la suplantación. Las sesiones en PHP deben ser configuradas y manejadas de manera segura, utilizando opciones como `session_regenerate_id()` para evitar la fijación de sesiones y asegurando que las cookies de sesión sean configuradas con parámetros seguros.

Vulnerabilidades Asociadas a MySQL:

1. **Inyección SQL:** MySQL es propenso a ataques de inyección SQL si no se validan ni escapan adecuadamente las consultas SQL. Se deben utilizar consultas parametrizadas o declarativas para prevenir esta vulnerabilidad.
2. **Acceso No Autorizado:** Es importante asegurarse de que las cuentas de usuario de MySQL tengan permisos adecuados y que no haya acceso no autorizado a las bases de datos.
3. **Contraseñas Débiles:** Las contraseñas de acceso a la base de datos deben ser seguras y no deben almacenarse en texto plano. Se recomienda utilizar funciones hash seguras y salting para almacenar contraseñas de manera segura.
4. **Exposición de Datos Sensibles:** Las bases de datos MySQL pueden contener datos sensibles, por lo que es fundamental asegurarse de que los datos estén protegidos mediante cifrado y controles de acceso adecuados.
5. **Parcheo y Actualizaciones:** Para evitar vulnerabilidades conocidas en MySQL, es importante mantener el sistema actualizado con las últimas actualizaciones y parches de seguridad.

Establecimiento y justificación del tipo de software.

Características:

Integración Completa: El software SSA integra todas las funciones clave de un taller automotriz, desde la gestión de inventario hasta la programación de citas, facturación, seguimiento de clientes y más. Esta integración garantiza una operación fluida y evita la redundancia de datos.

Personalización y Flexibilidad: El SSA permite la adaptación a las necesidades específicas de cada taller. Puede personalizarse para reflejar los procesos únicos de un taller en particular, asegurando que la solución se ajuste a las particularidades de cada negocio.

Automatización de Procesos: La automatización de tareas repetitivas, como la programación de citas, la gestión de inventario y la facturación, aumenta la eficiencia operativa. Esto libera tiempo para que los empleados se centren en tareas más estratégicas y de mayor valor.

Análisis y Reporting: El SSA proporciona herramientas analíticas avanzadas que permiten a los talleres automotrices realizar un seguimiento detallado del rendimiento. Los informes generados ayudan en la toma de decisiones informada, identificando áreas de mejora y oportunidades de crecimiento.

Escalabilidad: Con la capacidad de crecer y adaptarse a medida que el taller se expande, el SSA es escalable. Puede manejar un volumen creciente de datos y transacciones sin comprometer su rendimiento.

Ventajas:

Eficiencia Operativa Mejorada: La integración y automatización de procesos reducen los tiempos de espera, mejoran la productividad y minimizan los errores.

Toma de Decisiones Informada: Los análisis y reportes proporcionados por el sistema permiten a los gerentes tomar decisiones basadas en datos concretos, en lugar de depender de la intuición.

Mejora en la Experiencia del Cliente: La gestión eficiente de citas, tiempos de reparación y comunicación con los clientes contribuye a una experiencia del cliente mejorada.

Adaptabilidad a Cambios: La flexibilidad del SSA garantiza que el sistema pueda adaptarse a cambios en los procesos comerciales o en la industria automotriz en general.

Desventajas:

Costo Inicial y de Implementación: La adquisición e implementación de un ERP puede tener un costo inicial significativo, especialmente para talleres más pequeños.

Curva de Aprendizaje: La transición a un nuevo sistema puede requerir tiempo y capacitación para el personal, lo que puede afectar temporalmente la productividad.

Dependencia de la Tecnología: La eficacia del SSA depende de la infraestructura tecnológica y puede haber riesgos asociados, como problemas de seguridad o interrupciones del sistema.

Justificación:

La elección de un Sistema de Gestión Empresarial (ERP) para talleres automotrices está respaldada por la necesidad de una solución integral que aborde los desafíos específicos de la gestión en la industria automotriz. La integración de funciones clave, la flexibilidad para adaptarse a diversos modelos de negocios y la capacidad de análisis avanzado hacen del SSA una elección estratégica para mejorar la eficiencia operativa y la calidad del servicio. Aunque existen desafíos, como el costo inicial y la curva de aprendizaje, los beneficios a largo plazo en términos de eficiencia y toma de decisiones informada superan estas limitaciones, posicionando al SSA como una herramienta clave en la transformación de los talleres automotrices. Además, la metodología ágil Scrum en la gestión de recursos respalda la implementación exitosa y la adaptabilidad continua del software a medida que evolucionan las necesidades del taller.

Establecer y describir los riesgos del proyecto a desarrollar

Tipos de Riesgos	Riesgos posibles
Tecnológico	Conexión de internet deficiente (1) Falla de equipo de uno de los miembros del equipo (2) Poco conocimiento en el lenguaje de programación por parte de algunos miembros del equipo (3)
Personal	Falta de capacitación en el software utilizado (4) Falta de tiempo por miembros del equipo por trabajos, otros proyectos en curso, asuntos familiares y de salud (5)
De organización	Posible separación del equipo (6) Poca organización por diferente disponibilidad de horarios (7)
Herramientas	Problemas de compatibilidad de librerías para un modulo del proyecto (8)
Requerimientos	Cambio en la estructuración y el proyecto final (9)
Estimación	Se subestima el tiempo requerido para el desarrollo del software (10) Se subestima el tamaño del software (11)

Establecer y describir estrategias para gestionar los riesgos del proyecto

Identificación de los tipos de riesgo en el proyecto

Riesgo	Probabilidad	Efecto
Conexión de internet deficiente (1)	Moderada	Tolerable
Falla de equipo de uno de los miembros del equipo (2)	Moderada	Tolerable

Poco conocimiento en el lenguaje de programación por parte de algunos miembros del equipo (3)	Alta	Grave
Falta de capacitación en el software utilizado (4)	Moderada	Grave
Falta de tiempo por miembros del equipo por trabajos, otros proyectos en curso, asuntos familiares y de salud (5)	Alta	Grave
Posible separación del equipo (6)	Moderada	Grave
Poca organización por diferente disponibilidad de horarios (7)	Baja	Tolerable
Problemas de compatibilidad de librerías para un módulo del proyecto (8)	Moderada	Tolerable
Cambio en la estructuración y el proyecto final (9)	Moderada	Grave
Se subestima el tiempo requerido para el desarrollo del software (10)	Alta	Catastrófico
Se subestima el tamaño del software (11)	Alta	Grave

Análisis de los tipos de riesgo del proyecto

Riesgo	Probabilidad	Efecto
Conexión de Internet deficiente (1)	Moderada	Tolerable
Falla de equipo de uno de los miembros del equipo (2)	Moderada	Tolerable
Poco conocimiento del lenguaje de programación de parte de algunos miembros del equipo (3)	Alta	Grave
Falta de capacitación en el sw usado (4)	Moderada	Grave

Falta de tiempo de miembros del equipo por diferentes motivos (5)	Alta	Grave
Posible separación del equipo (6)	Moderada	Grave
Poca organización por diferentes disponibilidades de horarios (7)	Baja	Tolerable
Problemas de compatibilidades de librerías para un modelo del proyecto (8)	Moderada	Tolerable
Cambio en la estructuración y el proyecto final (9)	Moderada	Grave
Se subestima el tiempo requerido para el desarrollo del proyecto (10)	Alta	Catastrófico
Se subestima el tamaño del sw (11)	Alta	Grave

Planificación del riesgo que se encuentra en el proyecto

Riesgo	Estrategia
Conexión de internet	Divide las tareas en segmentos que puedan ser completados independientemente, minimizando la dependencia de una conexión constante.
Falla de equipo de uno de los miembros del equipo	Fomentar la capacidad de trabajar de forma remota, de manera que los miembros del equipo puedan seguir contribuyendo al proyecto incluso si están temporalmente sin acceso a su equipo principal.
Falta de conocimiento en un lenguaje de programación	<p>Establecer un entorno de desarrollo unificado con herramientas que faciliten la colaboración y la escritura de código eficiente.</p> <p>Implementa un programa continuo de capacitación en el lenguaje de programación utilizado en el proyecto.</p>

Falta de capacitación en el software utilizado	Diseñar un programa de capacitación estructurado que aborde las áreas identificadas de falta de conocimiento en el software del equipo.
Falta de tiempo entre los miembros por diferentes motivos	Realizar una planificación detallada de los recursos necesarios para el proyecto, considerando los posibles compromisos externos de los miembros del equipo.
	Fomentar una comunicación abierta dentro del equipo para que los miembros puedan informar sobre posibles conflictos de tiempo con anticipación.
Posible separación del equipo	Aborda proactivamente cualquier problema potencial antes de que se agrande.
Poca organización por diferente disponibilidad de horarios	<p>Considera la implementación de horarios de trabajo flexibles que se ajusten a las diferentes disponibilidades de los miembros del equipo.</p> <p>Implementa herramientas colaborativas en línea que permitan a los miembros del equipo colaborar de manera asíncrona.</p>
Problemas de compatibilidad de librerías	Herramientas de gestión de dependencias, como gestores de paquetes, para garantizar que las librerías y sus versiones sean gestionadas de manera controlada.
Cambio en la estructuración y el proyecto final	Comunicar cualquier cambio en la estructuración del proyecto de manera clara y transparente a todos los miembros del equipo y partes interesadas y explica las razones detrás del cambio y el impacto esperado.
Se subestima el tiempo requerido para el desarrollo del software	Proporcionar estimaciones detalladas para cada tarea, teniendo en cuenta aspectos como la complejidad, la dependencia y la experiencia requerida.

Tabla de monitoreo de riesgo del proyecto

Tipo de riesgo	Indicadores potenciales
----------------	-------------------------

Conexión de Internet	Entregas retrasadas debido a problemas de conexión; dependencia excesiva de una conexión constante.
Falla de Equipo de un Miembro	Contribución reducida al proyecto debido a la falta de acceso al equipo principal; retrasos en tareas asignadas.

Falta de Conocimiento en un Lenguaje de Programación	Dificultades en la colaboración eficiente; errores frecuentes en el código; falta de progreso en áreas específicas del desarrollo.
Falta de Capacitación en el Software Utilizado	Errores recurrentes relacionados con el software; falta de comprensión de las funcionalidades clave; baja eficiencia en el uso del software.
Falta de Tiempo entre los Miembros por Diferentes Motivos	Retrasos en las entregas debido a conflictos de tiempo; falta de comunicación sobre posibles problemas de programación.
Posible Separación del Equipo	Disminución de la colaboración; tensiones no resueltas afectando el trabajo en equipo.
Poca Organización por Diferente Disponibilidad de Horarios	Descoordinación en las tareas; falta de alineación en la ejecución de actividades planificadas.
Problemas de Compatibilidad de Librerías	Errores en el software relacionados con la compatibilidad de librerías; conflictos recurrentes durante la implementación.
Cambio en la Estructuración y el Proyecto Final	Confusión entre los miembros del equipo sobre el enfoque del proyecto; resistencia al cambio no comunicado.

Se Subestima el Tiempo Requerido para el Desarrollo del Software

Retrasos en las entregas; falta de cumplimiento de plazos; insatisfacción del cliente debido a expectativas no cumplidas.

Manual de Usuario.

Fuente del sistema

<https://drive.google.com/file/d/1e4LYcNfuOj6OQCEHUKZbIQwKDAVazacx/view?usp=sharing>

Link de videotutorial de uso del programa.

<https://www.youtube.com/watch?v=vRPZYovjVeQ>

Bibliografía

<https://hoplasoftware.com/mysql-sistema-de-gestion-de-bases-de-datos-relacionales/>

<https://blog.comparasoftware.com/que-es-planning-poker-en-scrum/>

Anexos

a) Autoevaluaciones

Autoevaluación		
Nombre del Estudiante: Rafael Albino Jovel Alfaro		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	10
2	Participé en forma activa en el trabajo de equipo	10
3	Mantuve comunicación con el equipo	10
4	Cumplí a tiempo con las actividades designadas	10
5	Aporté ideas de calidad	10
Total		50

Autoevaluación		
Nombre del Estudiante: Kevin Omar Diaz Pineda		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	8
2	Participé en forma activa en el trabajo de equipo	8
3	Mantuve comunicación con el equipo	8

4	Cumplí a tiempo con las actividades designadas	9
5	Aporté ideas de calidad	10
Total		43

Autoevaluación		
Nombre del Estudiante: Angel Iván Romero Hernández		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	10
2	Participé en forma activa en el trabajo de equipo	8
3	Mantuve comunicación con el equipo	8
4	Cumplí a tiempo con las actividades designadas	9
5	Aporté ideas de calidad	10
Total		45

Autoevaluación		
Nombre del Estudiante: Adiel Enoc Puentes Monge		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	8
2	Participé en forma activa en el trabajo de equipo	8
3	Mantuve comunicación con el equipo	8
4	Cumplí a tiempo con las actividades designadas	10
5	Aporté ideas de calidad	10
Total		44

b) Coevaluaciones

Coevaluación

Nombre del Evaluador: Adiel Enoc Puentes		
Nombre del Evaluado: Ángel Iván Romero		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		49

Coevaluación		
Nombre del Evaluador: Adiel Enoc Puentes		
Nombre del Evaluado: Rafael Albino Jovel		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		49

Coevaluación		
Nombre del Evaluador: Adiel Enoc Puentes		
Nombre del Evaluado: Kevin Omar Diaz Pineda		
N°	Aspecto a Evaluar	Rúbrica

		(Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		48

Coevaluación		
Nombre del Evaluador: Kevin Omar Diaz Pineda		
Nombre del Evaluado: Ángel Iván Romero		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	9
5	Antes de entregar la tarea, fue revisado por el evaluado	10
Total		49

Coevaluación		
Nombre del Evaluador: Kevin Omar Diaz Pineda		
Nombre del Evaluado: Rafael Albino Jovel		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10

2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		49

Coevaluación		
Nombre del Evaluador: Kevin Omar Diaz Pineda		
Nombre del Evaluado: Adiel Enoc Puentes		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	9
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	9
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	10
Total		48
Coevaluación		
Nombre del Evaluador: Angel Ivan Romero		
Nombre del Evaluado: Rafael Albino Jovel		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10

4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		49

Coevaluación		
Nombre del Evaluador: Angel Ivan Romero		
Nombre del Evaluado: Kevin Omar Diaz		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		49

Coevaluación		
Nombre del Evaluador: Angel Ivan Romero		
Nombre del Evaluado: Adiel Enoc Puentes		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	9
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	9
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9

Total	47
--------------	----

Coevaluación		
Nombre del Evaluador: Rafael Albino Jovel Alfaro		
Nombre del Evaluado: Angel Ivan Romero		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		49

Coevaluación		
Nombre del Evaluador: Rafael Albino Jovel Alfaro		
Nombre del Evaluado: Kevin Omar Diaz		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	10
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	10
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		49

Coevaluación		
Nombre del Evaluador: Rafael Albino Jovel Alfaro		

Nombre del Evaluado: Adiel Enoc Puentes		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	9
2	Demostró respeto y tolerancia hacia las opiniones de los demás	10
3	Aportó al desarrollo del proyecto	9
4	Propicia un clima de trabajo agradable	10
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		47