Requerimientos funcionales y no funcionales

Requerimientos funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Nombre | Descripción | Criterios de aceptación |
| 1 | Notificaciones de término de mantenimiento | El sistema debe mostrar automáticamente al usuario cuando se haya completado la mantención del vehículo. | La notificación debe enviarse dentro de los 2 minutos posteriores a la finalización del mantenimiento en el sistema. |
| 2 | Sistema de creación de cuentas para clientes | El mecánico debe poder abrir cuentas para clientes mediante un proceso rápido y sencillo, minimizando el tiempo de registro. | Se debe poder registrar al cliente utilizando nombre, correo electrónico, teléfono, Rut y contraseña.  El sistema debe validar la información ingresada antes de crear la cuenta. |
| 3 | Restablecimiento de contraseñas | El sistema debe permitir a los clientes y mecánicos restablecer sus contraseñas de manera rápida y sencilla en caso de que las olviden. | El usuario debe poder solicitar el restablecimiento mediante su correo electrónico.  El sistema debe enviar un enlace para permitir el restablecimiento de la contraseña. |
| 4 | Sistema de inicio de sesión | El sistema debe permitir a los clientes y mecánicos iniciar sesión de forma rápida y sencilla. | El usuario debe poder iniciar sesión usando su correo electrónico y contraseña.  El sistema debe verificar la autenticidad de las credenciales antes de otorgar acceso. |
| 5 | Pantalla de inicio para clientes y mecánicos | El sistema debe mostrar una pantalla de inicio personalizada tanto para clientes como para mecánicos, proporcionando una experiencia de usuario intuitiva y mostrando información relevante para cada grupo. | La pantalla de inicio para clientes debe mostrar las mantenciones en curso, elegir el servicio a realizar, generar QR, agendar hora y un menú personalizado.  La pantalla de inicio para mecánicos debe mostrar el cliente actual, la agenda del día, abrir cuenta para clientes, revisar las cotizaciones y un menú personalizado. |
| 6 | Vista usuario actual | El sistema deberá mostrar una pantalla que muestre el cliente actual dependiendo del Rut. | La vista mostrara 3 diferentes servicios: mantenciones del cliente, cotizaciones actuales y crear nueva mantención las cuales permitirá iniciar una mantención ya creada, aprobar una cotización y crear una nueva mantención. |
| 7 | Sistema de creación de cuentas para mecánicos | El administrador debe poder abrir cuentas para mecánicos mediante un proceso rápido y sencillo, minimizando el tiempo de registro | Se debe poder registrar al mecánico utilizando nombre, correo electrónico, teléfono, y contraseña.  El sistema debe validar la información ingresada antes de crear la cuenta. |
| 8 | Historial de mantenciones para mecánicos | El sistema debe permitir a los mecánicos acceder a un historial detallado de las mantenciones realizadas, con la opción de buscar repuestos utilizados. | El mecánico debe poder ver un listado de las mantenciones realizadas con fecha, tipo de servicio, nombre completo del cliente, email del cliente y nombre del vehículo.  El sistema debe ofrecer un campo de búsqueda para encontrar repuestos específicos dentro del historial. |
| 9 | Historial de mantenciones para clientes | El sistema debe permitir a los clientes ver su historial de mantenciones con la opción de buscar repuestos utilizados en cada mantenimiento. | El cliente debe poder acceder a un listado de las mantenciones realizadas a su vehículo mostrando patente, nombre del cliente, hora de la atención y estado del servicio.  El sistema debe permitir al cliente buscar repuestos usados en los mantenimientos anteriores. |
| 10 | Crear mantención mecánico | El sistema debe permitir a los mecánicos crear una mantención mediante una cotización anteriormente aprobada. | El mecánico podrá crear una mantención mediante una cotización anteriormente aprobada por el cliente. |
| 11 | Crear vehículo cliente | El sistema deberá permitirle al mecánico crear un vehículo vinculado a un cliente | El mecánico podría crear un vehículo que este directamente relacionado al cliente. |
| 12 | Panel de información del cliente y vehículo para mecánicos | El sistema debe proporcionar un panel donde el mecánico pueda ver la información del cliente, el vehículo, productos y servicios adquiridos, y confirmar el inicio de una mantención. | El mecánico debe poder ver los datos personales del cliente, detalles del vehículo y servicios adquiridos previamente.  El sistema debe incluir una opción para que el mecánico inicie un servicio y confirme los detalles con el cliente antes de proceder. |
| 13 | Apartado de configuración | El sistema debe proporcionar un apartado de configuración con las mismas opciones tanto para clientes como para mecánicos. | Los clientes y mecánicos deben poder gestionar opciones relacionadas con su perfil, cambiar contraseña y autos disponibles. |
| 14 | Crear cotización para el mecánico | El sistema deberá permitirle al mecánico crearle una cotización a un cliente en específico. | El mecánico debería poder crear una cotización y enviarla a un cliente o en el caso que el cliente este presencial aprobarla directamente. |
| 15 | Aprobar cotización cliente | El sistema deberá permitir al cliente aprobar o rechazar una cotización enviada. | El cliente podrá aprobar o rechazar una cotización enviada por el mecánico. |
| 16 | Confirmar cotización mecánico | El sistema deberá permitirle al mecánico confirmar la cotización anteriormente aprobada o rechazada. | El mecánico podrá aprobar o rechazar una cotización aprobada o rechazada previamente por el cliente para crear una mantención. |
| 17 | Estados actualizables para los clientes | El sistema debe implementar estados dentro de la app utilizado por el mecánico que permita mantener al cliente informado sobre el estado actual de la mantención de su vehículo, con actualizaciones en tiempo real. | El instructivo debe actualizarse automáticamente según el estado de la mantención del vehículo (ej.: comenzando, en proceso y listo para entregar). |
| 18 | Cambiar estado mantención mecánico | El sistema deberá permitirle al mecánico cambiar el estado de una mantención anteriormente creada. | El mecánico podrá cambiar el estado de una mantención anteriormente creada de un cliente especifico. |
| 19 | Agenda para mecánicos | El sistema debe permitir que los mecánicos tengan acceso a una agenda diaria integrada con todas las mantenciones programadas. | La agenda debe mostrar todas las citas y trabajos asignados al mecánico para el día.  El sistema debe permitir al mecánico ver los detalles de cada mantención y acceder a la información del cliente y vehículo. |
| 20 | Ver mantención actual para el mecánico | El sistema deberá permitir ver la lista de las mantenciones actuales del mecánico. | El mecánico deberá ver el estado de la mantención y actualizar dicho estado. |
| 21 | Apartado de reservas de horas | El sistema debe permitir a los clientes reservar horas de mantención directamente desde la aplicación. | El cliente debe poder seleccionar una fecha y hora disponible para la mantención escogiendo al mecánico que va a hacer la reserva. |
| 22 | Aprobar o rechazar una reserva para mecánico | El sistema deberá permitir aprobar o rechazar una reserva vista por el mecánico. | El mecánico deberá poder aprobar o rechazar una reserva hecha por el cliente. |
| 23 | Autocompletado en panel del mecánico | El sistema debe implementar un panel para mecánicos con campos de autocompletado para "vehículo" basados en datos previamente almacenados. | Los campos de “vehículo” deben sugerir los vehículos registrados en base a la marca y modelo más comunes en el mercado. |
| 24 | Sistema de reseñas para clientes | El sistema debe permitir a los clientes entregar una reseña de los servicios recibidos mediante una clasificación de estrellas y comentarios, siendo esta funcionalidad opcional. | El cliente debe poder dejar una reseña con un rango de 1 a 5 estrellas.  El sistema debe permitir al cliente escribir un comentario adicional.  La opción de dejar una reseña debe ser opcional y no obligatoria. |
| 25 | QR con información del cliente | El sistema debe permitir al cliente generar un código QR con los datos actualizados, que será utilizado por el mecánico durante el servicio. | El cliente debe poder generar el código QR desde su propio perfil.  El código QR debe contener información relevante como matrícula, tipo de vehículo, historial de mantenciones y próximos servicios. |
| 26 | Escaneo de patente | El sistema debe permitir a los mecánicos escanear la patente del vehículo para obtener información sobre el mismo, incluyendo el historial de mantenciones y datos relevantes para el servicio. | El mecánico debe poder escanear la patente del vehículo desde la app utilizando la cámara del celular.  El sistema debe mostrar automáticamente los datos del vehículo, historial de servicios y repuestos relacionados al mecánico. |
| 27 | Escanear QR mecánico y buscar por Rut del cliente | El sistema deberá permitir escanear el código QR del cliente o ingresar el Rut para obtener información del cliente. | El mecánico debería poder escanear el QR del cliente o poder ingresar su Rut. |

Requerimientos no funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Nombre | Descripción | Criterios de aceptación |
| 1 | Disponibilidad | El sistema debe estar disponible el 99.5% del tiempo durante el horario de atención de los talleres. | La plataforma debe garantizar una disponibilidad mínima del 99.5% durante el horario de funcionamiento de los talleres.  En caso de mantenimiento programado, se debe notificar con al menos 24 horas de anticipación. |
| 2 | Compatibilidad | El sistema debe ser compatible con las versiones más utilizadas del sistema operativo Android, así como con los navegadores más utilizados (Chrome, Safari, Edge y Firefox). | La app debe funcionar correctamente en dispositivos con Android 9.0 o superior.  El sistema web debe garantizar una experiencia de usuario consistente en Chrome, Safari, Edge y Firefox en sus dos versiones más recientes. |
| 3 | Escalabilidad | El sistema debe ser escalable para soportar hasta 5,000 usuarios simultáneos sin afectar el rendimiento o la disponibilidad del servicio. | El sistema debe poder aumentar su capacidad automáticamente para soportar peaks de tráfico sin deterioro en el rendimiento.  El servicio debe garantizar un tiempo de respuesta constante incluso con el aumento en el número de usuarios simultáneos. |
| 4 | Mantenibilidad | El código del sistema debe seguir estándares de programación y buenas prácticas, como Clean Code, para asegurar que pueda ser mantenido y actualizado fácilmente por cualquier miembro del equipo. | El código debe estar bien documentado, con comentarios claros y concisos donde sea necesario.  Deben seguirse convenciones de nombres consistentes y estructuras de carpetas organizadas. |
| 5 | Portabilidad | El sistema debe ser portable y desplegable en diferentes entornos de ejecución como servidores locales, nube pública (AWS, Azure) y privada sin necesidad de realizar modificaciones importantes en el código. | La app debe estar preparada para funcionar en diferentes servicios de nube y servidores locales, utilizando contenedores (por ejemplo, Docker) para facilitar el despliegue.  El sistema debe ser independiente de la plataforma, pudiendo migrar a diferentes entornos sin alterar la funcionalidad principal.  Los entornos de prueba y producción deben ser fácilmente replicables. |
| 6 | Recuperación ante fallos | En caso de fallo del sistema, la recuperación debe realizarse en un máximo de 30 - 40 minutos, y no debe haber pérdida de datos de los clientes o mecánicos. | Deben implementarse mecanismos de respaldo automático de la base de datos cada hora.  El sistema debe contar con un plan de contingencia que permita la restauración de datos en menos de 30 minutos.  En caso de caída, los usuarios deben recibir notificaciones apropiadas y estimación de tiempo de recuperación. |