

Documento de requerimientos de software

GlowDrive
Fecha: 12/02/2024

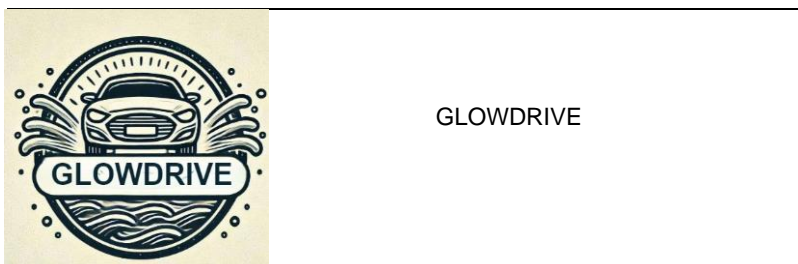


Tabla de contenido

1. Historial de Versiones	5
2. Información del Proyecto	5
3. Aprobaciones.....	6
4. Propósito.....	6
5. Alcance del producto / Software	7
6. Referencias.....	8
7. Funcionalidades del producto	9
8. Clases y características de usuarios	10
8.1 DIAGRAMA DE CASOS	10
8.2 DIAGRAMA DE CLASES	11
8.3 DIAGRAMA DE SECUENCIA	12
8.4 DIAGRAMA DE TIEMPO	16
9. Entorno operativo	17
10. Requerimientos funcionales.....	18
10.1. Gestión de citas	18
10.2 Requerimientos funcionales:	19
10.3 Manejo de errores:.....	19
10.4. Gestión de pagos.....	20
Acciones iniciadoras y comportamiento esperado:.....	20
Requerimientos funcionales:	20



Manejo de errores:.....	21
10.3. Localización de autolavados	21
Acciones iniciadoras y comportamiento esperado:.....	21
Requerimientos funcionales:	22
Manejo de errores:	22
Reglas de negocio	22
Reglas generales del sistema:.....	22
11. Requerimientos de interfaces externas	23
11.1. Interfaces de usuario	23
11.2. Estilo de la interfaz:.....	24
11.3. Interfaces de hardware	24
Dispositivos soportados:	25
Protocolos de comunicación soportados:.....	25
Interacciones con hardware:.....	25
11.4. Interfaces de software	25
11.4.1. Componentes de software externos:.....	26
11.4.2 Interacciones:.....	26
11.5. Interfaces de comunicación	26
Requerimientos de comunicación:	27
12. Requerimientos no funcionales	27
12.1. Rendimiento	28
12.2. Escalabilidad	28



GLOWDRIVE

12.3. Seguridad.....	28
12.4. Disponibilidad.....	29
12.5. Usabilidad.....	29
12.6. Compatibilidad.....	29
12.7. Mantenibilidad.....	29
12.8. Fiabilidad.....	30
13. Otros requerimientos.....	30
13.1. Requerimientos de bases de datos.....	30
13.2. Internacionalización ajustada a Colombia- Bogotá.....	31
13.3. Aspectos legales.....	31
13.4. Requerimientos de seguridad.....	31
13.5. Objetivos de reúso de componentes de software.....	32
14. TAXONOMÍA Y CONTENIDO DEL MANUAL TÉCNICO Y DE OPERACIÓN DEL SISTEMA.....	32
14.1 DIAGRAMA DE ACTIVIDADES.....	32
Glosario.....	33



GLOWDRIVE

1. Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
12/02/24	1.0.0	Todo el grupo	GlowDrive	Inicio
18/02/24	1.0.1	Camilo Ramirez	GlowDrive	Modificaciones
05/03/2024	1.0.2	César Rueda	GlowDrive	Clases y características del Usuario
19/03/2024	1.0.3	Todo el grupo	GlowDrive	Taxonomía y contenido del manual técnico y de operación del sistema
27/03/2024	1.0.4	César Rueda	GlowDrive	Actualización tabla de contenido
29/03/2024	1.5.0	Todo el grupo	GlowDrive	Implementación de diagramas
15/04/2024	1.5.1			Revisión John Jairo Rodriguez
14/09/2024	1.6.1	Todo el grupo	GlowDrive	Modificaciones

2. Información del Proyecto

Empresa / Organización	GlowDrive
Proyecto	Desarrollo Aplicación web
Fecha de preparación	12/02/24
Cliente	
Patrocinador principal	
Gerente / Líder de Proyecto	César Rueda
Gerente / Líder de Análisis de negocio y requerimientos	Santiago Romero



3. Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma

4. Propósito

GlowDrive es un aplicativo que tiene como propósito facilitar un encuentro a la hora de llevar nuestro vehículo a un autolavado debido a que en ciertos casos la persona dueño del vehículo no tiene conocimiento de los puntos en su zona más cercana. (MRR, 2018)

Con el aplicativo se busca crear una base de datos donde se encuentren; precios, calidad del servicio y sugerencias de los puntos más cercanos a la persona, así mismo habrá una opción de reserva para no hacer filas, ya que es incómodo para el cliente (Luna Santos, Santos Quiroz, Carrera Carrasco, González Domínguez, & Santaella Benavente, 2021) y previene inconvenientes con la empresa que brinda el servicio. De la misma forma se desarrollará esta base de datos con los siguientes métodos de programación:

- POO (Programación orientada a objetos), utilizando recolección de datos.
- PHP (backend)
- JSON (Backend)
- MYSQL-SQL(Base de datos)
- CSS/HTML/JavaScript (fronend)



5. Alcance del producto / Software

Este software tiene como objetivo principal **facilitar y agilizar el proceso de agendamiento de servicios de lavado de automóviles** para los usuarios. A través de una aplicación móvil intuitiva, los clientes podrán solicitar y programar sus lavados de manera rápida y sencilla, desde cualquier lugar y dispositivo con conexión a internet.

Beneficios

- **Para los usuarios:**
 - **Comodidad:** Agendamiento de citas desde cualquier lugar y a cualquier hora.
 - **Rapidez:** Proceso de solicitud simplificado.
 - **Control:** Seguimiento del estado de su servicio.
- **Para los negocios:**
 - **Mayor eficiencia:** Optimización de la gestión de citas y recursos.
 - **Incremento de clientes:** Atracción de nuevos clientes y fidelización de los existentes.
 - **Mejora de la imagen:** Presentación de una imagen moderna y tecnológica.

Objetivos Específicos

- **Desarrollar** una aplicación móvil intuitiva y fácil de usar.
- **Integrar** un sistema de pago seguro y eficiente.
- **Implementar** un sistema de gestión de citas que permita a los negocios optimizar sus operaciones.
- **Ofrecer** un servicio de atención al cliente eficaz. (orderry, s.f.)

Alineación con los Objetivos Corporativos

El desarrollo de este software se encuentra alineado con los siguientes objetivos corporativos (Luna Santos, Santos Quiroz, Carrera Carrasco, González Domínguez, & Santaella Benavente, 2021):

- **Expansión del mercado:** Captación de nuevos clientes y aumento de la cuota de mercado.
- **Mejora de la experiencia del cliente:** Oferta de servicios innovadores y personalizados.
- **Optimización de procesos:** Aumento de la eficiencia operativa y reducción de costos.

Consideraciones Adicionales:



- **Alcance funcional:** Define con mayor detalle las funcionalidades específicas que incluirá la aplicación (ej: tipos de lavado, opciones adicionales, notificaciones, etc.).
- **Plataformas:** Especifica las plataformas móviles para las que se desarrollará la aplicación (iOS, Android, ambas).
- **Integraciones:** Indica si se requiere la integración con otros sistemas (ej: sistemas de gestión de clientes, proveedores de pagos).
- **Métricas de éxito:** Establece los indicadores clave de rendimiento (KPIs) que permitirán medir el éxito del proyecto (ej: número de descargas, cantidad de citas agendadas, satisfacción del cliente).

6.

Referencias

- Ahmed, T. (28 de Junio de 2021). *Youtube.com*. Obtenido de Youtube.com: https://www.youtube.com/watch?v=K_64Xmm3OcA
- Booklux. (s.f.). *Booklux.com*. Obtenido de Booklux.com: <https://www.booklux.com/es/car-wash-booking-system>
- Britto, M. S. (12 de Abril de 2024). *youtube.com*. Obtenido de youtube.com: https://www.youtube.com/watch?v=y_TmZtB3Guc
- DanielFard. (29 de Agosto de 2021). *Youtube.com*. Obtenido de Youtube.com: <https://www.youtube.com/watch?v=k8n6FOeUpS8>
- INSICON, C. (09 de Junio de 2022). *youtube.com*. Obtenido de youtube.com: <https://www.youtube.com/watch?v=Q74nxFBCHCI>
- isai-ismael. (2017). *Github.com*. Obtenido de Github.com: <https://github.com/isai-ismael>
- Luna Santos, R., Santos Quiroz, R. A., Carrera Carrasco, E., González Domínguez, I., & Santaella Benavente, M. G. (2021). *Creación e implementación de una aplicación web para la*. Puebla: RITI.



Media, V. (03 de Nov de 2020). *youtube.com*. Obtenido de youtube.com:
<https://www.youtube.com/watch?v=F8dnYNTncoU>

MR, C. d. (08 de Marzo de 2023). *youtube.com*. Obtenido de youtube.com:
<https://www.youtube.com/watch?v=ogYl2BDTEKs>

MRR, V. (17 de Octubre de 2018). *youtube.com*. Obtenido de Yputube.com:
<https://www.youtube.com/watch?v=w8q0C2Fb-Gg>

orderry. (s.f.). *orderry.com*. Obtenido de orderry.com: <https://orderry.com/es/car-detailing-software/>

Rivas, M. (2021). *Github*. Obtenido de Github:
<https://github.com/marcosrivasr/Curso-PHP-MySQL>

Studios, I. (24 de Junio de 2022). *Youtube.com*. Obtenido de Youtube.com:
<https://www.youtube.com/watch?v=p1JFNFMc1ps>

Studios, I. (2 de Junio de 2022). *Youtube.com*. Obtenido de Youtube.com:
https://www.youtube.com/watch?v=ZNbcmbVs5_M

Talleralpha. (s.f.). *Talleralpha.com*. Obtenido de Talleralpha.com:
<https://www.talleralpha.com/auto-lavado/>

7. Funcionalidades del producto

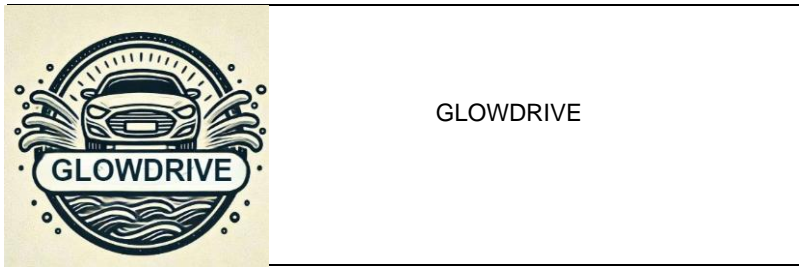
- Registrarse e iniciar sesión
- Opciones de lavado (depende la opción de lavado)
- Función para el vendedor y para el usuario
- Idioma
- Ni Cercanía; permite que los usuarios puedan observar la cercanía de los autolavados, calidad, precio (moto y/o auto)
- Tipo de lavado: plan básico, medio y premium



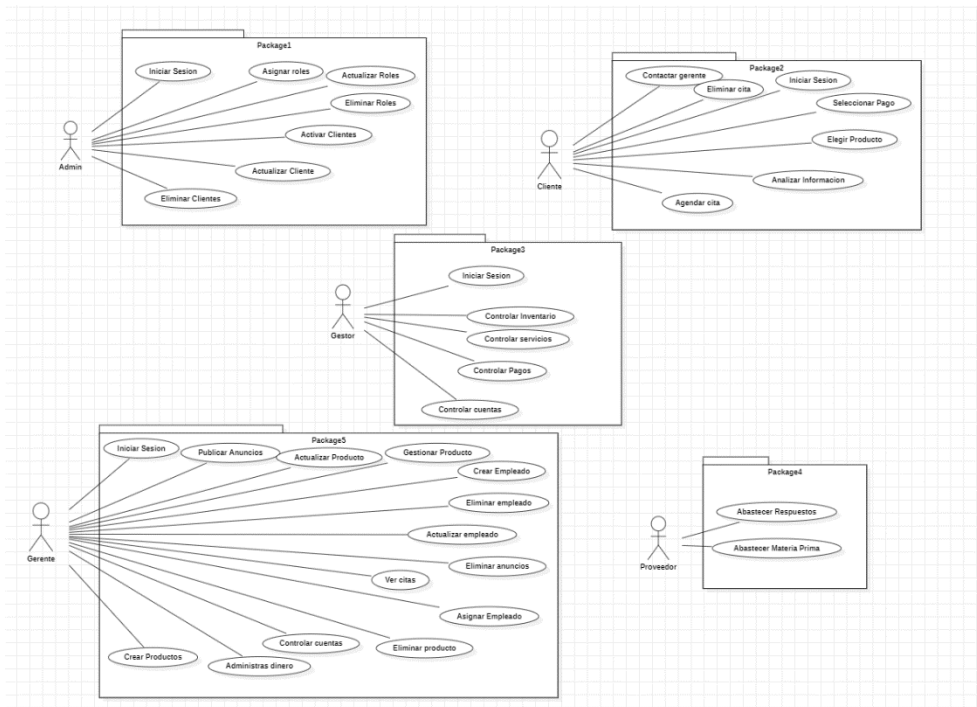
- Como nuestro aplicativo actúa como intermediario, hemos diseñado un sistema que verifica cuándo un servicio ha finalizado. En ese momento, el usuario procede a realizar el pago al gerente encargado del establecimiento. Es en este punto donde entra en juego la opción de "Terminado", que se identifica con el estado "Pagado".
- Fiabilidad (top) opiniones, reseñas.

8. Clases y características de usuarios

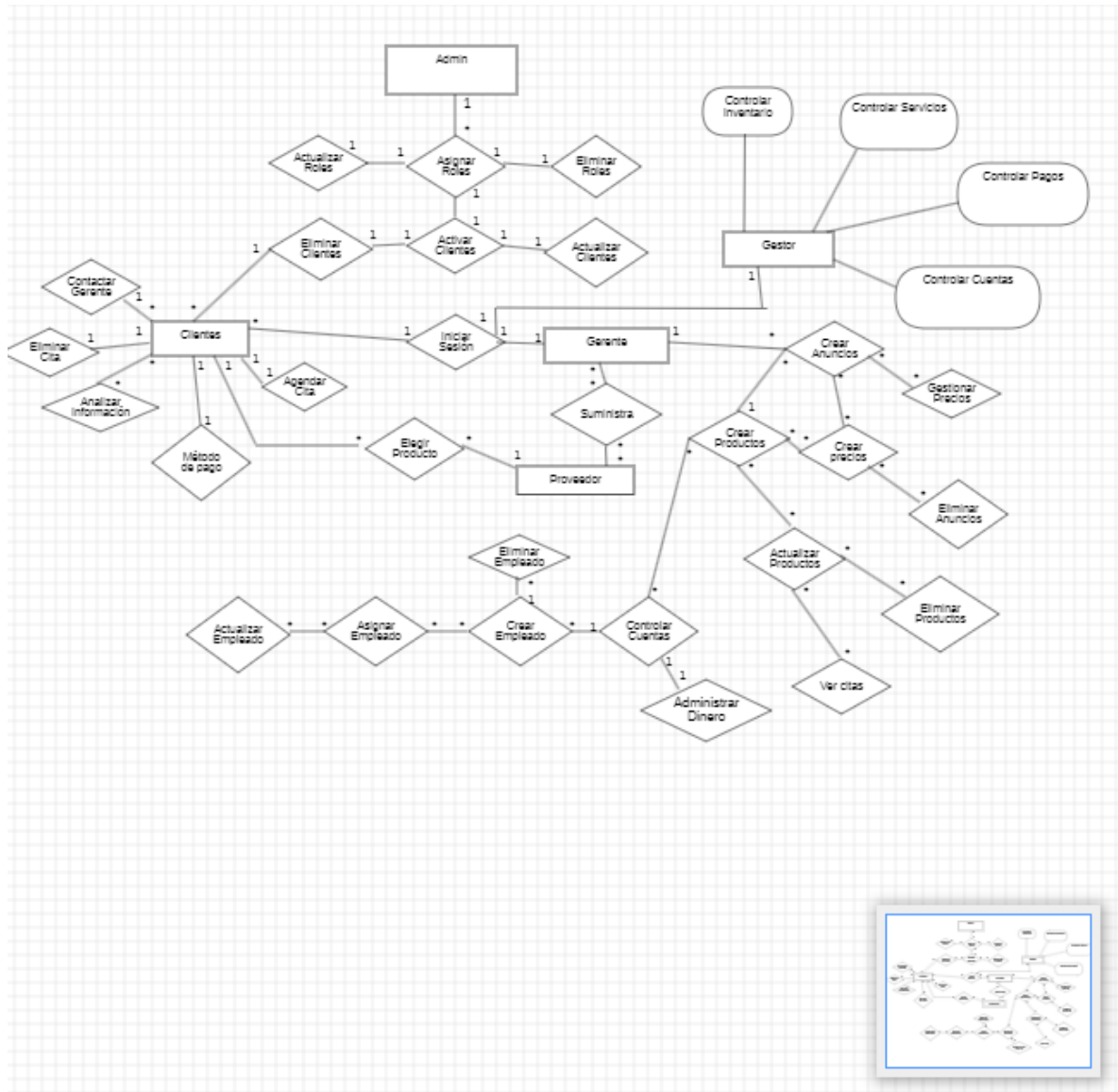
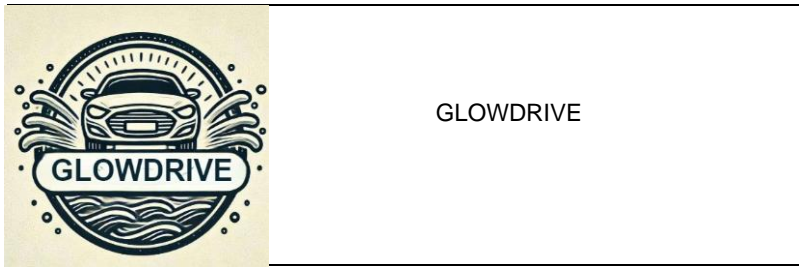
8.1 DIAGRAMA DE CASOS



GLOWDRIVE



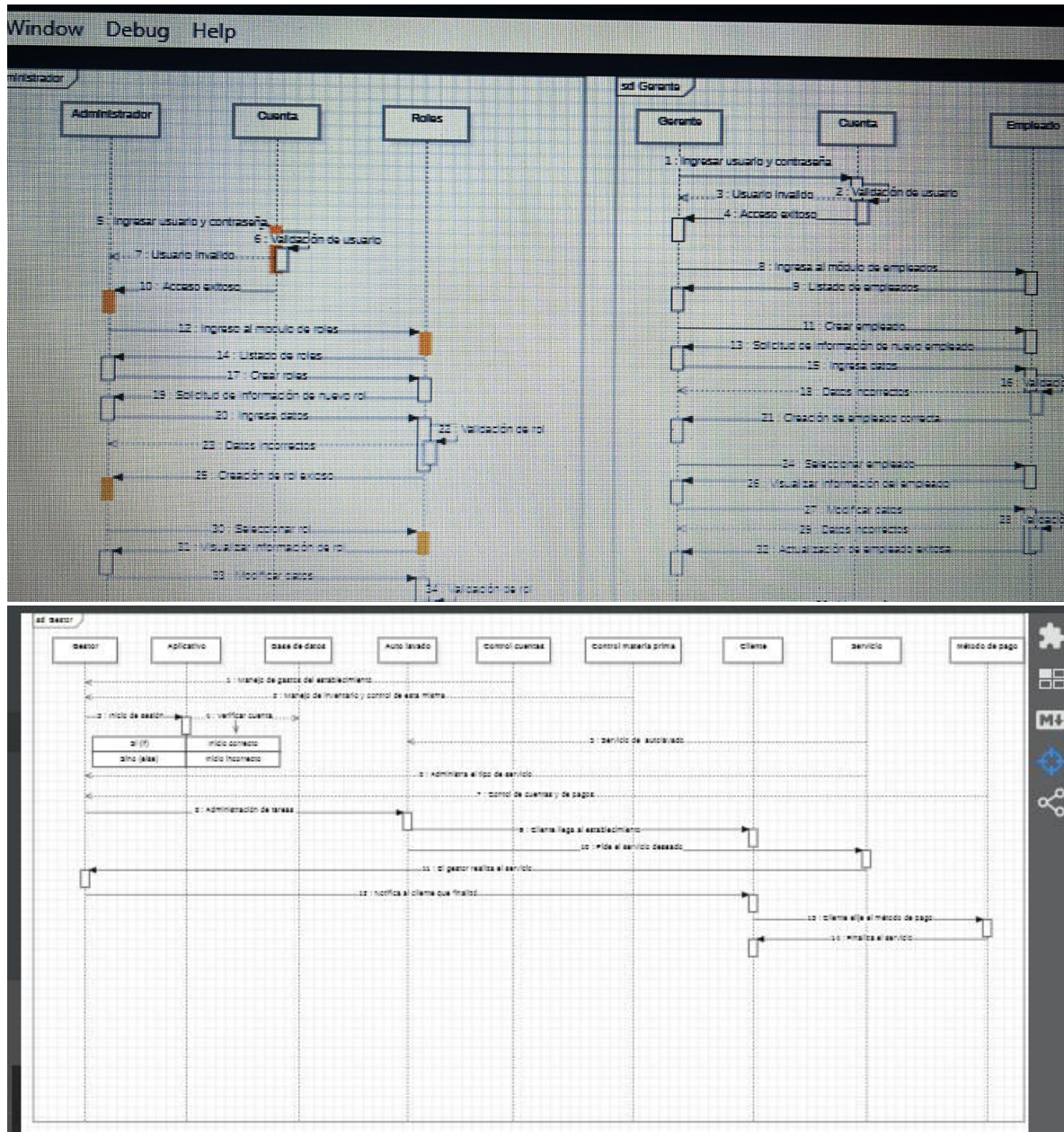
8.2 DIAGRAMA DE CLASES



8.3 DIAGRAMA DE SECUENCIA

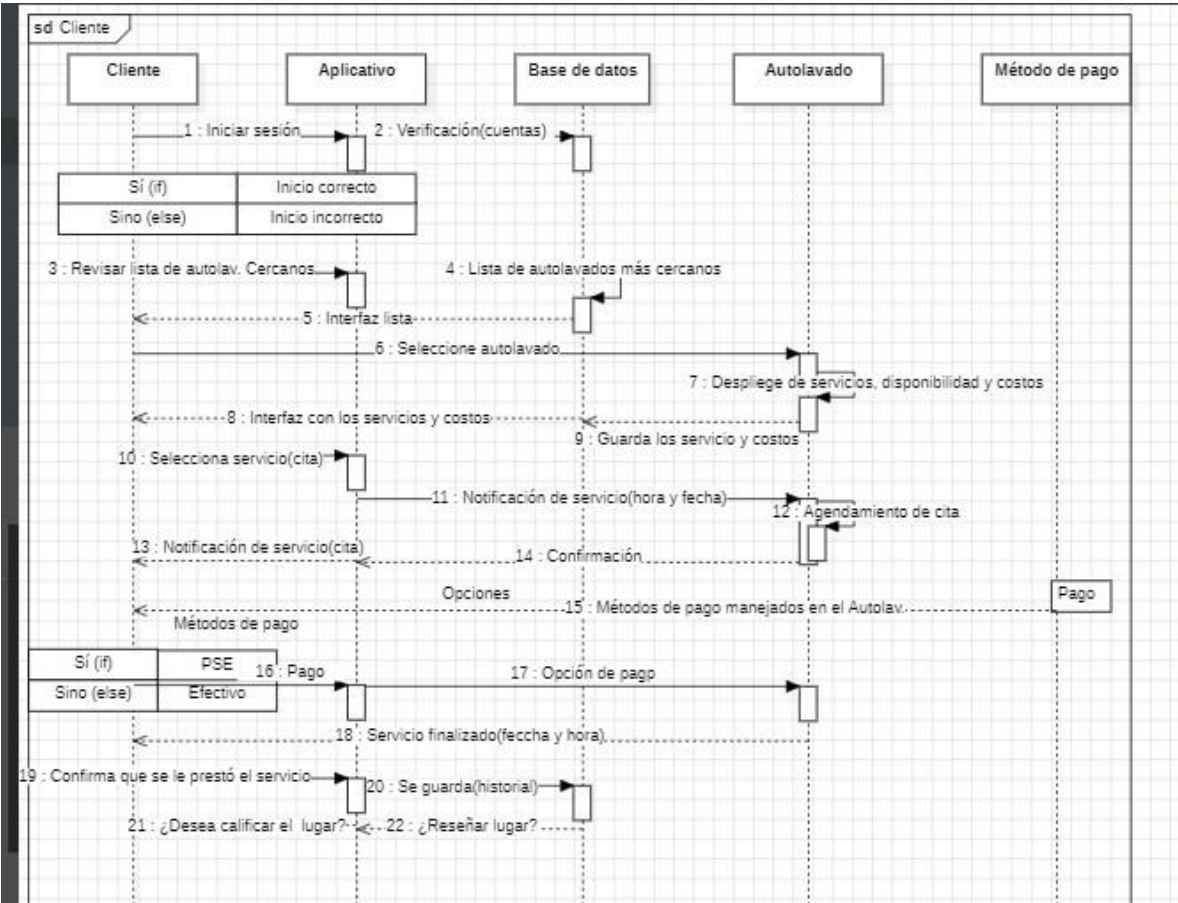


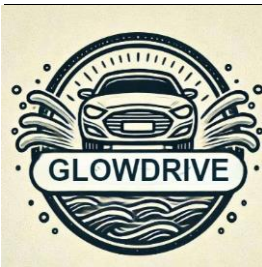
GLOWDRIVE



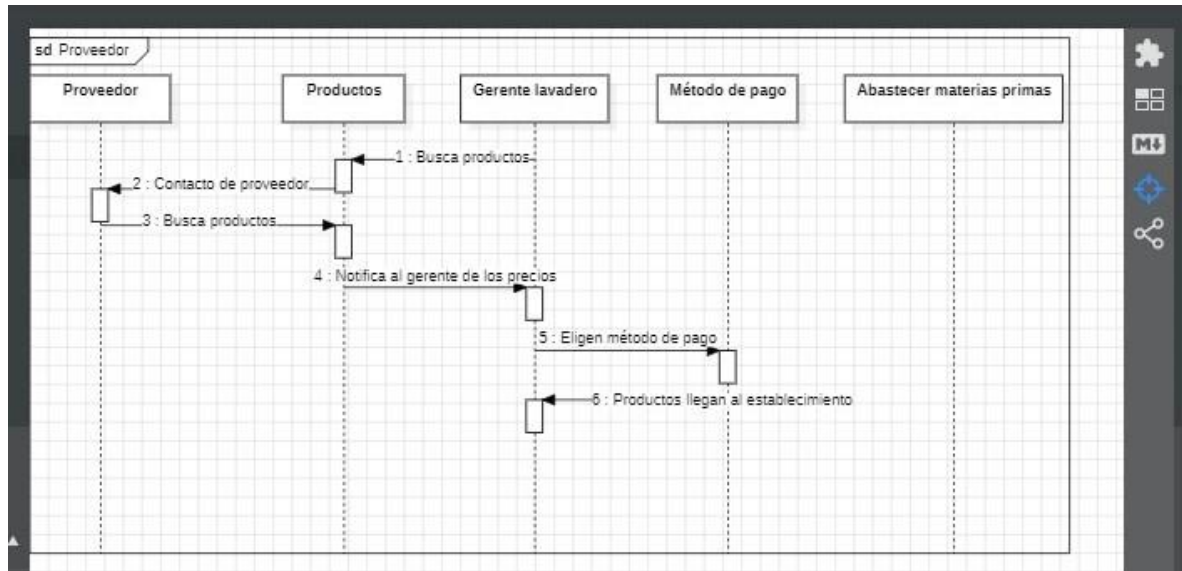


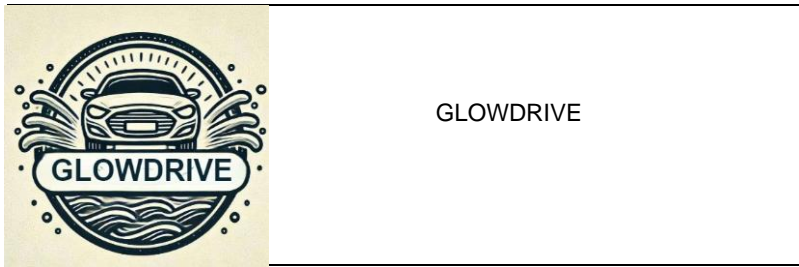
GLOWDRIVE





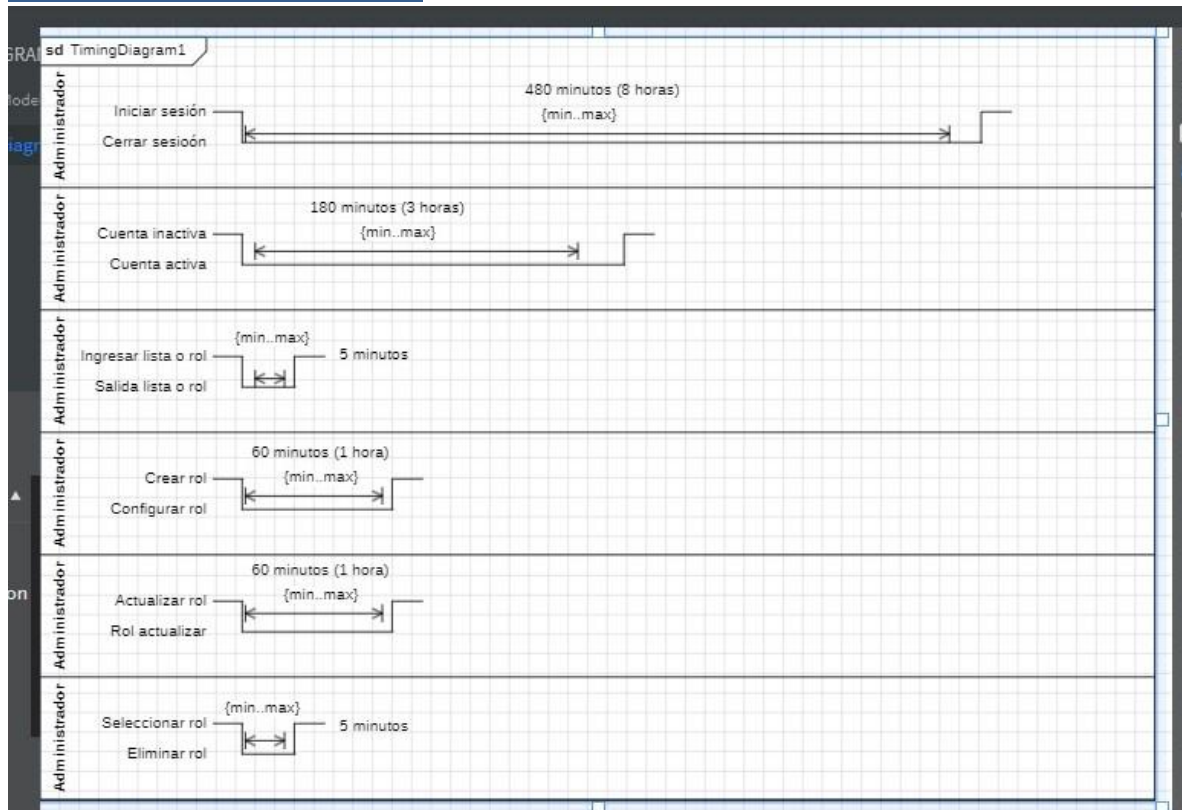
GLOWDRIVE

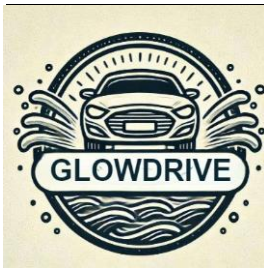




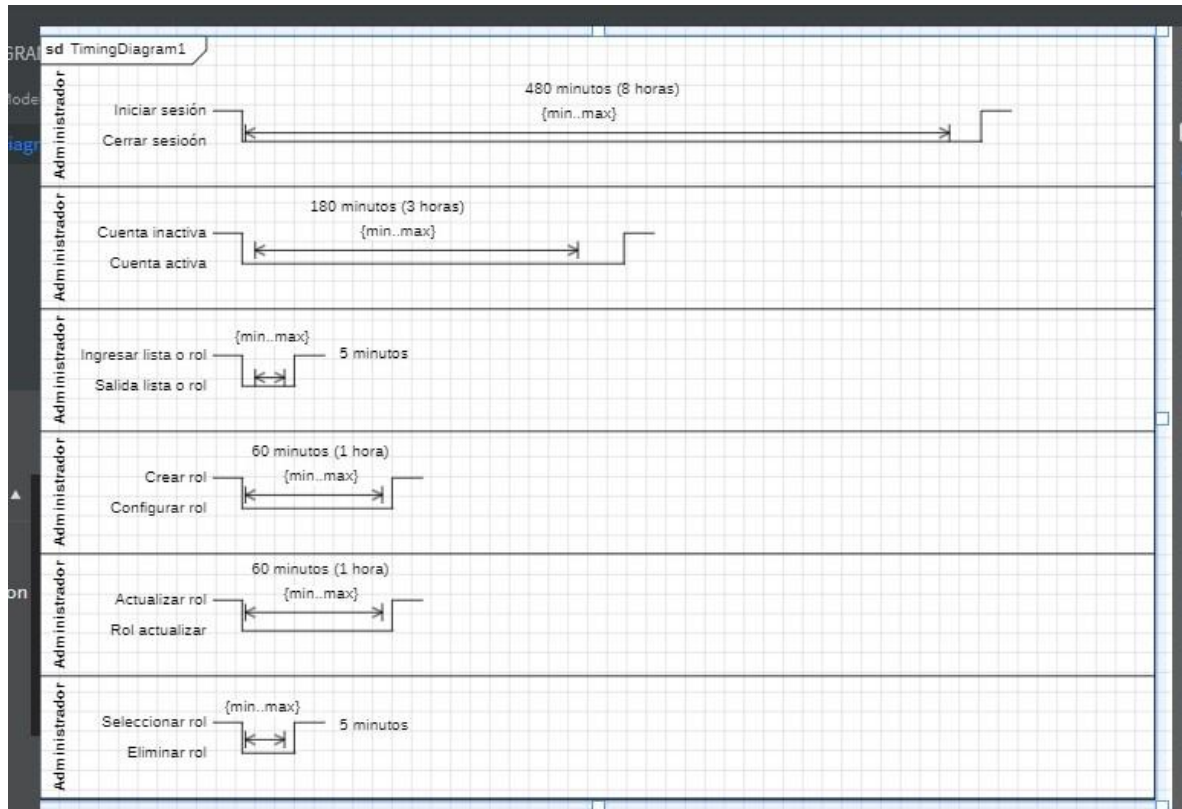
GLOWDRIVE

8.4 DIAGRAMA DE TIEMPO





GLOWDRIVE



9. Entorno operativo

El aplicativo funcionará desde cualquier dispositivo, ya que tiene aplicado su respectivo responsive para que se vea y funcione de una buena manera desde cualquier dispositivo. (Britto, 2024) En requisitos, el único fundamental es que sea cual sea el dispositivo en el que acceda; este tenga acceso a conexión internet. (MRR, 2018)

En cuanto a los gerentes de los autolavados, necesitaran la base de datos xxxx



10. Requerimientos funcionales

Los requerimientos funcionales de GlowDriver describen las actividades clave que el sistema debe realizar para cumplir con las necesidades del negocio. Estas especificaciones detallan el comportamiento del software bajo ciertas condiciones, asegurando que el aplicativo funcione de acuerdo con las expectativas de los usuarios (Luna Santos, Santos Quiroz, Carrera Carrasco, González Domínguez, & Santaella Benavente, 2021).

10.1. Gestión de citas

Descripción: Esta funcionalidad permite a los usuarios del sistema programar citas con los autolavados disponibles, seleccionando servicios y fechas específicas. (Booklux, s.f.)

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

1. El usuario selecciona un autolavado de la lista de opciones cercanas.
2. El usuario selecciona el servicio deseado y una fecha y hora disponibles.
3. El sistema verifica la disponibilidad del autolavado y confirma la cita.



GLOWDRIVE

4. El usuario recibe una notificación confirmando la cita programada.

10.2 Requerimientos funcionales:

(Luna Santos, Santos Quiroz, Carrera Carrasco, González Domínguez, & Santaella Benavente, 2021)

- REQ-1: El sistema debe permitir que los usuarios vean una lista de autolavados cercanos según su ubicación actual.
- REQ-2: El sistema debe verificar la disponibilidad del autolavado seleccionado y mostrar las fechas y horas disponibles para programar la cita.
- REQ-3: El sistema debe enviar notificaciones de confirmación al usuario una vez que la cita ha sido agendada.
- REQ-4: El sistema debe gestionar la actualización del estado de la cita (aceptada, rechazada, terminada).

10.3 Manejo de errores:

(Luna Santos, Santos Quiroz, Carrera Carrasco, González Domínguez, & Santaella Benavente, 2021)

- Si el autolavado no tiene disponibilidad en la fecha seleccionada, el sistema debe mostrar un mensaje de error y sugerir fechas alternativas.
- Si se ingresan datos inválidos (ej. fecha pasada), el sistema debe mostrar un mensaje de error y evitar la programación de la cita.



10.4. Gestión de pagos

Descripción: El sistema permite que los gerentes de autolavados gestionen el pago de los servicios, y que los usuarios realicen los pagos una vez finalizado el servicio.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

1. El gerente marca el servicio como "Terminado" una vez que se completa la cita.
2. El usuario procede a realizar el pago por el servicio recibido.
3. El sistema genera un recibo digital para ambas partes (usuario y gerente).
4. El estado de la cita cambia a "Pagado" y el historial se actualiza.

Requerimientos funcionales:

- REQ-5: El sistema debe permitir a los gerentes marcar una cita como terminada.
- REQ-6: El sistema debe generar un recibo digital que incluya detalles del servicio, fecha, monto pagado y estado de la cita.



GLOWDRIVE

- REQ-7: El sistema debe permitir que los usuarios vean y descarguen el recibo una vez que el servicio esté pagado.
- REQ-8: El sistema debe actualizar el estado de la cita a "Pagado" después de realizarse el pago.

Manejo de errores:

- Si el pago no se procesa correctamente, el sistema debe notificar al usuario y permitir intentarlo nuevamente.
- Si el servicio no ha sido marcado como "Terminado", el sistema no debe permitir que el usuario proceda al pago. (isai-ismael, 2017)

10.3. Localización de autolavados

Descripción: Esta funcionalidad permite a los usuarios localizar autolavados cercanos utilizando la API de Leaflet y visualizar su ubicación en un mapa interactivo. (Studios, Youtube.com, 2022)

Prioridad: Media

Acciones iniciadoras y comportamiento esperado:

1. El usuario accede a la sección de búsqueda de autolavados.



2. El sistema utiliza la API de Leaflet para mostrar un mapa con los autolavados cercanos.
3. El usuario puede hacer clic en un autolavado en el mapa para ver información adicional y agendar una cita. (Ahmed, 2021)

Requerimientos funcionales:

- REQ-9: El sistema debe mostrar un mapa interactivo utilizando la API de Leaflet con la ubicación de los autolavados registrados.
- REQ-10: El sistema debe permitir que los usuarios seleccionen un autolavado del mapa y vean sus detalles (nombre, dirección, servicios, etc.).
- REQ-11: El sistema debe permitir que los administradores del autolavado ingresen la latitud y longitud de su establecimiento. (Luna Santos, Santos Quiroz, Carrera Carrasco, González Domínguez, & Santaella Benavente, 2021)

Manejo de errores:

- Si la API de Leaflet no responde, el sistema debe mostrar un mensaje de error e intentar cargar el mapa nuevamente. (Luna Santos, Santos Quiroz, Carrera Carrasco, González Domínguez, & Santaella Benavente, 2021)
- Si la ubicación ingresada por el administrador es incorrecta, el sistema debe permitir editar la latitud y longitud del autolavado.

Reglas de negocio

Reglas generales del sistema:



- Solo los gerentes o dueños de autolavados tienen la capacidad de modificar el estado de una cita.
- El sistema debe garantizar que un usuario no pueda agendar dos citas en el mismo autolavado en la misma franja horaria.
- Los pagos solo pueden realizarse si el servicio ha sido marcado como "Terminado" por el gerente.

Estas reglas aseguran el correcto funcionamiento de *GlowDriver*, evitando conflictos y asegurando que las funciones se realicen de manera segura y transparente.

11. Requerimientos de interfaces externas

El sistema GlowDriver interactúa con varios componentes externos, tanto a nivel de usuario como a nivel de hardware y software. A continuación, se describen las características y especificaciones de cada una de las interfaces externas necesarias para su correcto funcionamiento. (Luna Santos, Santos Quiroz, Carrera Carrasco, González Domínguez, & Santaella Benavente, 2021)

11.1. Interfaces de usuario

Esta sección describe las características visuales y funcionales que la interfaz de usuario (UI) de GlowDriver debe cumplir para garantizar una experiencia fluida e intuitiva. (isai-ismael, 2017)

- Diseño responsivo: El sistema debe ser accesible desde dispositivos móviles y de escritorio, adaptándose automáticamente a diferentes tamaños de pantalla.
- Estándares de GUI (Interfaz Gráfica de Usuario):
- Organización de pantalla: La disposición debe ser clara y centrada en la facilidad de uso. Los menús deben estar situados en la parte superior o lateral, permitiendo acceso rápido a las funciones principales.



- Botones: Todos los botones de acción deben ser de fácil identificación, con etiquetas descriptivas (por ejemplo, "Agendar cita", "Ver servicios", "Confirmar pago"). Los botones deben ser lo suficientemente grandes para su uso en dispositivos táctiles.
- Funciones comunes: En todas las pantallas, debe haber accesos visibles a la barra de búsqueda, al perfil del usuario, y a la ayuda o soporte técnico.

11.2. Estilo de la interfaz:

(Rivas, 2021)

- Colores: La paleta de colores debe incluir tonos azules marinos y contrastes que aseguren buena visibilidad en cualquier dispositivo.
 - Fuentes: Las fuentes deben ser legibles, preferentemente sans-serif, con tamaños adecuados para facilitar la lectura en dispositivos móviles.
-
- Ejemplos de pantallas:
 - Pantalla principal: Presentará el mapa interactivo mostrando los autolavados cercanos.
 - Pantalla de agendamiento: Permitirá seleccionar el servicio, fecha, y horario.
 - Pantalla de pagos: Confirmación de los pagos realizados con detalles del servicio y la opción de descargar recibos.

11.3. Interfaces de hardware

(INSICON, 2022)



El sistema GlowDriver debe ser compatible con una variedad de dispositivos de hardware para garantizar su uso amplio.

Dispositivos soportados:

- Computadores personales (PC y Mac)
- Dispositivos móviles (smartphones y tablets con iOS y Android)

Protocolos de comunicación soportados:

- El sistema debe ser capaz de comunicarse con dispositivos móviles y ordenadores a través de Wi-Fi y redes móviles (3G, 4G, 5G).

Interacciones con hardware:

- Los gerentes de los autolavados podrán acceder a la plataforma desde dispositivos móviles para gestionar citas y marcar servicios como terminados.

11.4. Interfaces de software

Esta sección detalla las interacciones del sistema GlowDriver con otros componentes de software.



11.4.1. Componentes de software externos:

- Bases de datos: El sistema se conecta a una base de datos MySQL para gestionar la información de usuarios, citas, servicios, autolavados, y pagos.
- Sistema operativo: GlowDriver debe ser compatible con sistemas operativos modernos, incluyendo Windows, macOS, Android, y iOS.
- Librerías y herramientas:
- Leaflet API: Utilizado para mostrar mapas interactivos en la plataforma.
- Composer: Para la gestión de dependencias de PHP.
- PHP y JavaScript: Para la lógica del servidor y las interacciones de la aplicación.

11.4.2 Interacciones:

- GlowDriver se comunicará con la API de Leaflet para obtener y mostrar la ubicación de los autolavados en tiempo real.

11.5. Interfaces de comunicación

El sistema GlowDriver requiere varias funciones de comunicación para interactuar de manera eficiente con los usuarios y servicios externos.



Requerimientos de comunicación:

- Email: El sistema debe enviar correos electrónicos de confirmación para citas, notificaciones de estado (aceptado, rechazado, pagado), y recibos.
- Navegadores web: El sistema debe ser accesible desde navegadores como Google Chrome, Firefox, Safari y Edge.
- Protocolos de comunicación de red: GlowDriver utilizará HTTP y HTTPS para la comunicación segura entre el cliente (usuario) y el servidor.

Formatos de mensajería:

- Los correos electrónicos enviados por el sistema deben estar en formato HTML y proporcionar una versión de texto simple como alternativa.
- Requerimientos de seguridad y encriptación:
- Todos los datos sensibles (como credenciales de usuario y transacciones) deben ser encriptados mediante HTTPS y TLS.
- Las comunicaciones de pago deben cumplir con los estándares de seguridad PCI-DSS para proteger la información financiera de los usuarios.

12. Requerimientos no funcionales

Los requerimientos no funcionales son aquellos que especifican criterios para evaluar la operación del sistema *GlowDriver*, más allá de las funcionalidades específicas. Estos determinan la calidad, rendimiento y otros factores críticos que aseguran una experiencia óptima para los usuarios.



12.1. Rendimiento

- El sistema debe cargar y mostrar el mapa interactivo de los autolavados en menos de 4 segundos para garantizar una experiencia fluida, incluso en dispositivos móviles con conexiones a internet de baja velocidad.
- Las consultas a la base de datos deben realizarse en un tiempo inferior a 1 segundo para evitar demoras al gestionar citas y pagos.
- El sistema debe ser capaz de manejar hasta 1,000 usuarios simultáneos sin experimentar caídas de servicio o retrasos significativos.

12.2. Escalabilidad

- *GlowDriver* debe ser diseñado de manera modular, permitiendo la fácil adición de nuevas funcionalidades y módulos (por ejemplo, expansión a otros tipos de servicios relacionados con vehículos).
- El sistema debe permitir la integración de nuevos autolavados sin afectar el rendimiento general, soportando la expansión a nuevas ciudades o regiones.

12.3. Seguridad

- Toda la información de los usuarios, incluidas credenciales y datos personales, debe ser almacenada y transmitida de forma encriptada utilizando SSL/TLS.
- El sistema debe garantizar la autenticación de usuarios a través de contraseñas seguras y, de ser posible, ofrecer autenticación multifactor (2FA) para los gerentes de autolavados.
- Los pagos deben cumplir con los estándares de seguridad PCI-DSS, asegurando la protección de la información financiera de los usuarios.



12.4. Disponibilidad

- El sistema *GlowDriver* debe garantizar una disponibilidad del 99.9% anual, permitiendo la operación continua del servicio para la programación de citas y gestión de autolavados.
- En caso de mantenimiento planificado, los usuarios deben ser notificados con 48 horas de antelación.

12.5. Usabilidad

- La interfaz debe ser intuitiva y fácil de usar para cualquier tipo de usuario, sin requerir capacitación previa.
- *GlowDriver* debe ser accesible en dispositivos con pantalla táctil, asegurando que todas las funcionalidades sean utilizables mediante gestos simples y botones bien visibles.
- La aplicación debe cumplir con estándares de accesibilidad, garantizando que usuarios con discapacidades puedan interactuar con el sistema (por ejemplo, compatibilidad con lectores de pantalla).

12.6. Compatibilidad

- El sistema debe ser compatible con los principales navegadores web (Google Chrome, Firefox, Opera y Edge) y las versiones más recientes de los mismos.
- Debe garantizarse la compatibilidad en dispositivos móviles con sistemas operativos Android (versión 6.0 en adelante) e iOS (versión 12.0 en adelante).

12.7. Mantenibilidad

- El código de *GlowDriver* debe seguir buenas prácticas de desarrollo (por ejemplo, uso de patrones de diseño, arquitectura limpia) para facilitar futuras actualizaciones y modificaciones.



- La documentación del código debe ser clara y detallada, incluyendo comentarios dentro del código y manuales técnicos para desarrolladores.

12.8. Fiabilidad

- El sistema debe asegurar la integridad de los datos almacenados, garantizando que no haya pérdida de información durante las transacciones o fallos del sistema.
- Los usuarios deben recibir notificaciones inmediatas en caso de cualquier error del sistema, y estos deben ser registrados para análisis y solución rápida.

13. Otros requerimientos

Requerimientos no cubiertos en ninguna otra sección del documento de requerimientos de software, por ejemplo: Requerimientos de bases de datos, internacionalización, legales y objetivos de reúso de componentes de software.

13.1. Requerimientos de bases de datos

- La base de datos será relacional y utilizará MySQL como sistema de gestión.
- Las tablas deberán tener índices adecuados para mejorar la eficiencia de las consultas, especialmente en aquellas con grandes volúmenes de datos, como las tablas citas y reservas.
- Todas las relaciones entre tablas deberán respetar las reglas de normalización, y se deben definir llaves foráneas para asegurar la integridad de los datos.
- Las transacciones se utilizarán en operaciones críticas para garantizar la consistencia de los datos.



13.2. Internacionalización ajustada a Colombia- Bogotá

- La aplicación *GlowDriver* solo estará disponible en español y será específica para usuarios en Bogotá, D.C.
- Los formatos de fecha y moneda deberán estar en el formato colombiano (DD/MM/AAAA y pesos colombianos - COP).
- Los textos de la interfaz de usuario estarán en español y los términos regionales deben ser fácilmente entendibles para los usuarios locales de Bogotá.

13.3. Aspectos legales

- *GlowDriver* deberá cumplir con las regulaciones colombianas de protección de datos, en especial la Ley 1581 de 2012 sobre la protección de datos personales.
- Los usuarios deberán aceptar los términos y condiciones de uso, junto con la política de privacidad, antes de registrarse en la aplicación.
- Toda la información personal deberá ser almacenada de manera segura y protegida, cumpliendo con las normativas locales y garantizando la confidencialidad de los datos.
- Se deben adoptar medidas de encriptación para asegurar el almacenamiento de datos personales, como contraseñas y datos financieros de los usuarios.

13.4. Requerimientos de seguridad

- Se debe implementar un sistema de auditoría que registre todos los cambios críticos realizados en la plataforma, como modificaciones en los datos por parte de gerentes o administradores de autolavados.
- Los intentos fallidos de acceso al sistema deberán ser registrados y notificados para revisión.



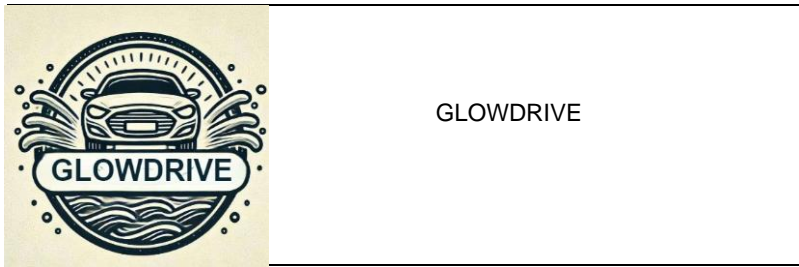
- Información sensible, como las contraseñas de los usuarios, será almacenada utilizando algoritmos de hash seguros, como bcrypt, para garantizar su seguridad.

13.5. Objetivos de reúso de componentes de software

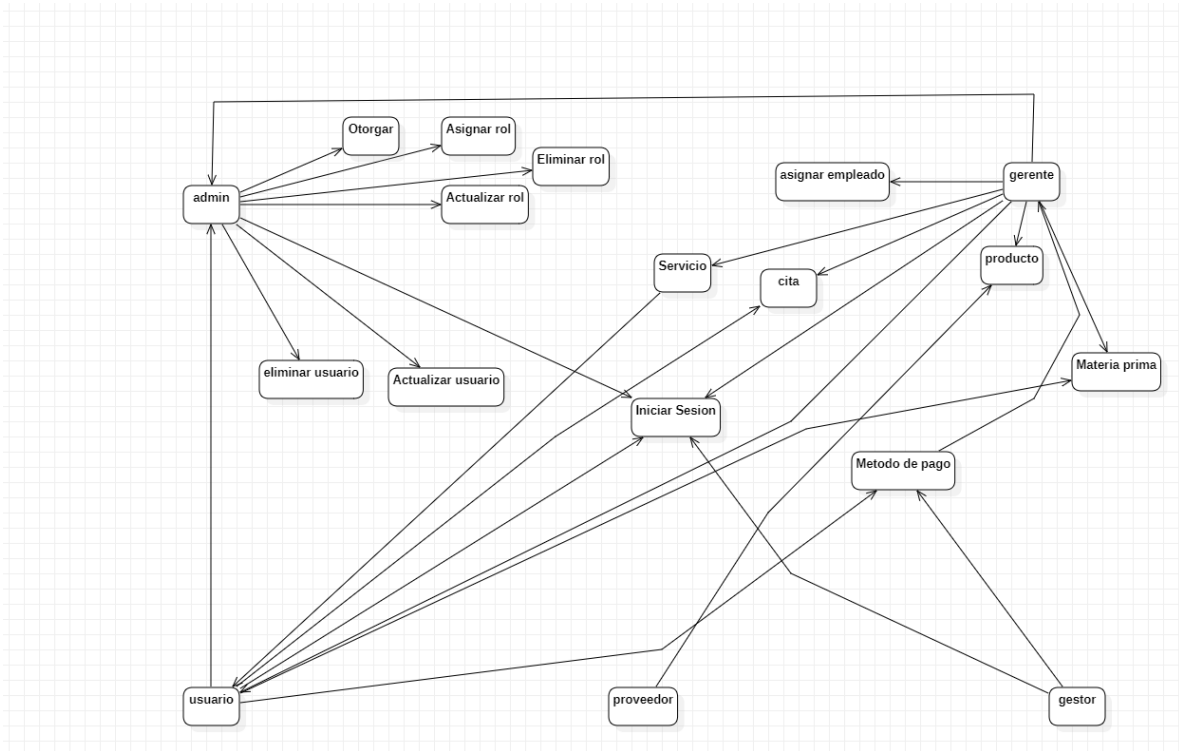
- El sistema será desarrollado de manera modular, para facilitar el reúso de componentes en otros proyectos relacionados a *GlowDriver* en Colombia.
- Las funcionalidades comunes, como el manejo de usuarios y la autenticación, se implementarán como módulos reutilizables que podrán integrarse fácilmente en futuros desarrollos locales.
- Se promoverá el uso de librerías y frameworks de código abierto, cuando sea posible, para reducir costos y mejorar la capacidad de mantenimiento del sistema.

14. TAXONOMÍA Y CONTENIDO DEL MANUAL TÉCNICO Y DE OPERACIÓN DEL SISTEMA

14.1 DIAGRAMA DE ACTIVIDADES



GLOWDRIVE



Glosario

Descripción de términos y siglas necesarias para el entendimiento del documento de requerimientos de software.

1. API (Application Programming Interface):



Es un conjunto de reglas que permite a las aplicaciones interactuar entre sí. Las APIs permiten la comunicación entre diferentes programas y servicios sin necesidad de que los desarrolladores conozcan los detalles internos de los sistemas.

2. Autenticación:

Proceso mediante el cual un sistema verifica la identidad de un usuario, normalmente utilizando credenciales como nombre de usuario y contraseña.

3. Autorización:

Proceso que determina si un usuario tiene permisos para acceder a ciertos recursos o realizar acciones dentro de una aplicación, basándose en sus roles o privilegios.

4. Backend:

Es la parte del software que no es visible para los usuarios y que maneja toda la lógica, las bases de datos, y la integración con otras aplicaciones o servicios.

5. Base de datos relacional:

Un tipo de base de datos que organiza los datos en tablas, las cuales están relacionadas entre sí mediante llaves primarias y llaves foráneas.

6. Bcrypt:

Es una función de hash utilizada para encriptar contraseñas de manera segura. Es ampliamente usada por los desarrolladores para proteger los datos sensibles de los usuarios.

7. Bug:

Un error o defecto en un programa que impide que funcione correctamente.

8. CRUD (Create, Read, Update, Delete):



Acrónimo que describe las operaciones básicas que se pueden realizar sobre datos en una base de datos o sistema. En español: Crear, Leer, Actualizar, Eliminar.

9. CSS (Cascading Style Sheets):

Un lenguaje que se utiliza para describir la presentación de un documento HTML. Se utiliza para darle estilo y diseño a las páginas web.

10. Deployment:

Proceso de mover una aplicación desde el entorno de desarrollo a un entorno de producción donde los usuarios finales podrán acceder a ella.

11. Frontend:

Es la parte del software que los usuarios pueden ver e interactuar, como las páginas web, los botones, formularios y demás elementos visuales.

12. Framework:

Un conjunto de herramientas y librerías predefinidas que facilita el desarrollo de software, ofreciendo soluciones comunes a problemas frecuentes en la programación.

13. Git:

Sistema de control de versiones utilizado para rastrear los cambios en el código y permitir la colaboración entre múltiples desarrolladores.

14. GitHub:

Plataforma basada en Git donde los desarrolladores pueden alojar sus repositorios de código y colaborar en proyectos de software.

15. HTTP (Hypertext Transfer Protocol):

Protocolo utilizado para la transmisión de datos en la web. Es el lenguaje que los navegadores web utilizan para comunicarse con los servidores.



16. HTTPS (Hypertext Transfer Protocol Secure):

Versión segura de HTTP, que utiliza encriptación para proteger los datos transmitidos entre el navegador y el servidor.

17. JSON (JavaScript Object Notation):

Formato de intercambio de datos ligero y fácil de leer. Es muy utilizado para enviar y recibir datos entre servidores y aplicaciones.

18. Live Server:

Extensión para editores de código como VSCode que permite a los desarrolladores lanzar un servidor local para ver los cambios en tiempo real mientras desarrollan una aplicación web.

19. MySQL:

Sistema de gestión de bases de datos relacional que se utiliza para almacenar y gestionar grandes cantidades de información. Es muy popular en el desarrollo web.

20. NPM (Node Package Manager):

Gestor de paquetes para Node.js que permite a los desarrolladores instalar y compartir librerías y módulos de JavaScript para facilitar el desarrollo de aplicaciones.

21. PDO (PHP Data Objects):

Es una extensión de PHP que permite conectarse a bases de datos de manera segura y eficiente, utilizando una interfaz consistente para acceder a diferentes bases de datos.

22. Repositorio:

Es un espacio donde se almacena y gestiona el código de un proyecto de software, comúnmente alojado en plataformas como GitHub o GitLab.



23. REST (Representational State Transfer):

Un estilo de arquitectura que se utiliza para diseñar APIs. En las APIs REST, los recursos se representan mediante URLs y se accede a ellos mediante métodos HTTP como GET, POST, PUT y DELETE.

24. Responsividad:

Capacidad de una aplicación web de adaptarse a diferentes tamaños de pantalla y dispositivos, ofreciendo una experiencia de usuario óptima en móviles, tablets y escritorios.

25. SQL (Structured Query Language):

Lenguaje utilizado para gestionar y manipular bases de datos relacionales. Se utiliza para realizar consultas, actualizaciones, eliminaciones e inserciones de datos en una base de datos.

26. UI (User Interface):

Interfaz de usuario. Es el conjunto de elementos gráficos y visuales con los que el usuario interactúa dentro de un sistema o aplicación.

27. UX (User Experience):

Experiencia de usuario. Hace referencia a cómo se siente y percibe el usuario al interactuar con un producto o sistema.

28. Token:

Unidad de autenticación que se utiliza para validar las acciones de un usuario en una aplicación sin necesidad de que el usuario envíe sus credenciales en cada solicitud. Los tokens pueden ser parte de sistemas de seguridad como JWT (JSON Web Tokens).

29. VSCode (Visual Studio Code):



Editor de código ligero y popular entre los desarrolladores, que soporta una gran cantidad de lenguajes de programación y ofrece extensiones para mejorar el flujo de trabajo.

30. XAMPP:

Paquete de software que incluye Apache, MySQL, PHP y Perl, utilizado para crear servidores locales de desarrollo, especialmente en el desarrollo de aplicaciones web en PHP.

31. API Leaflet:

Biblioteca JavaScript de código abierto para mapas interactivos. Se utiliza en el aplicativo *GlowDriver* para mostrar la ubicación de los autolavados en un mapa interactivo.

32. SSL/TLS: Secure Sockets Layer / Transport Layer Security:

Protocolos criptográficos diseñados para proporcionar seguridad en la transmisión de datos a través de redes, como la web. TLS es el sucesor de SSL y se usa para encriptar la comunicación entre el navegador y el servidor.

2FA.Two-Factor Authentication:

Autenticación de dos factores, una medida de seguridad que requiere dos métodos diferentes de verificación para acceder a una cuenta o sistema. Usualmente combina algo que el usuario sabe (como una contraseña) con algo que el usuario tiene (como un código enviado a su teléfono).