

Carkier

CONTENIDO

1-. Introducción	4
2-. Resumen	4
Palabras claves:	5
3-. Objetivos	5
Objetivos Generales:	5
Objetivos Específicos:	6
4-. Análisis Del Contexto y Estado Del Arte	7
4-1 Análisis del contexto.....	7
4-1.1 Análisis Tecnológico	7
4-1.2 Análisis Económico.....	9
4-1.3 Análisis sociocultural	11
4-3-4 Análisis Legislativo	13
4.3.5 Contexto social y económico.....	15
4-2 Análisis del arte.....	17
4-2.1. Situación Actual del Mercado de Alquiler de Coches	17
4-2.2 Análisis Competencia	20
Análisis debilidades y fortalezas (DAFO)	22
Innovación.....	23
Sistema de puntos y recompensas	23
Alquiler por horas.....	23
Aplicación móvil intuitiva	24
5-. Análisis De Requisitos	25
5-1 Requisitos Funcionales	25
5-2 Requisitos no funcionales	28
6-. Diseño	29
6.1-. Diagrama Físico Y/O Lógico de Red	29
6.2-. Diagrama Relacional de Base de Datos	30
6.3-. Diagramas de Clases	35
6.3.2-. Crear Carnets de conducir.....	40
6.3.8-. Estado del Vehículo.....	67
6.3-. DIAGRAMA DE INTERFACES.....	85
7-. Planificación	117
Diagrama de Gantt.....	117

8-. Implementación/ Desarrollo	118
8.1 IDE Desarrollo	118
8.2 Lenguajes de Programación.....	119
8.2 Lenguajes de Programación.....	120
8.2.1 Base de Datos.....	120
8.2.2 API	120
8.2.3 Aplicación Móvil.....	120
8.2.4 Aplicación de Escritorio	121
8.3 Librerías Utilizadas	121
8.3.1 MySQL.....	121
8.3.2 API.....	121
8.3.3 Aplicación de Escritorio / Gestión.....	122
8.4 Control de Versiones	122
8.5 Convenciones de Codificación y Estándares Seguidos.....	123
9-. Despliegue y Mantenimiento.....	124
9.1-. Aplicación Móvil	124
9.1.1 Desarrollo	124
9.1.2 Distribución	124
9.2 Aplicación Escritorio	125
9.2.1 Desarrollo	125
9.2.2 Distribución	125
9.3 ACTUALIZACIONES	125
9.3.1 Ciclo de Actualizaciones	125
10-. Pruebas	126
10-. Gestión Económica.....	128
10.1-. Denominación y Categoría	128
10.3-. Instalación.....	128
10.4-. Servicios	128
10.5-. Tipos de clientes	129
10.5-. Nivel de ocupación.....	129
10.5-. Estructura Organizativa	129
10.6-. Análisis Económico	129
11-. Conclusión.....	131

1-. INTRODUCCIÓN

Al planificar unas vacaciones, buscamos que todo sea sencillo y libre de complicaciones. Sin embargo, uno de los mayores desafíos suele ser la movilidad en grandes ciudades, especialmente debido a las restricciones de acceso impuestas por la contaminación ambiental, que limitan el uso de ciertos vehículos.

Esta aplicación busca resolver ese inconveniente ofreciendo servicios respetuosos con las normativas medioambientales, facilitando un desplazamiento cómodo y sostenible para los usuarios.

2-. RESUMEN

El objetivo principal de este proyecto es ofrecer un servicio innovador de alquiler de vehículos diseñado para abordar los desafíos relacionados con la contaminación ambiental, particularmente en zonas de bajas emisiones. Estas áreas, reguladas por estrictas normativas medioambientales, restringen el acceso a vehículos que no cumplan con los estándares necesarios para minimizar el impacto ambiental.

Con esta iniciativa, se busca proporcionar una solución eficiente, práctica y sostenible para los usuarios que necesitan desplazarse en estas zonas, evitando inconvenientes asociados con las restricciones de movilidad. Todos los vehículos incluidos en el servicio están equipados con las etiquetas medioambientales requeridas, lo que garantiza el cumplimiento total de las normativas vigentes.

Además de resolver problemas de accesibilidad, este proyecto promueve un enfoque más consciente hacia la movilidad urbana, fomentando el uso de alternativas responsables y respetuosas con el medio ambiente. Así, no solo se facilita el desplazamiento en las áreas restringidas, sino que también se contribuye a la mejora de la calidad del aire y al bienestar colectivo.

PALABRAS CLAVES:

Ventas, alquiler, vehículos, viajes, vacaciones, transporte, eco, medioambiente e híbrido

3-. OBJETIVOS

OBJETIVOS GENERALES:

Desarrollar un servicio integral de alquiler de vehículos accesible para todos los usuarios con licencia de conducir, ofreciendo una selección de vehículos que cumplan con las normativas medioambientales establecidas para zonas de bajas emisiones. Este servicio tiene como propósito principal facilitar la movilidad sin restricciones, proporcionando una solución práctica y sostenible para los desafíos de desplazamiento en entornos urbanos regulados.

Además, se garantizará una amplia disponibilidad de vehículos para satisfacer diversas necesidades, incluyendo opciones de gasolina, diésel, híbridos y eléctricos. De esta manera, se busca no solo cumplir con las regulaciones medioambientales, sino también adaptarse a las preferencias y requerimientos específicos de cada usuario, promoviendo una experiencia flexible, cómoda y eficiente.

OBJETIVOS ESPECÍFICOS:

- 1) Inicio sesión.
- 2) Creación de clientes.
- 3) Proporcionar vehículos que puedan moverse por cualquier zonal.
- 4) Proporcionar vehículos de todo tipo de clase.
- 5) Permitir a los usuarios reservar el vehículo a través de una aplicación móvil fácil de usar.
- 6) Permitir a los usuarios dejar comentarios de dicho vehículo.
- 7) Responder comentarios.
- 8) Permitir al usuario conseguir puntos para que pueda rebajar más el precio de alguna reserva.
- 9) Poder ver los contratos de ti mismo.
- 10) El sistema no tardaría mucho en mostrar las especificaciones.
- 11) Cifrado de contraseña.
- 12) Intermediarios para las consultas (API)

4-. ANÁLISIS DEL CONTEXTO Y ESTADO DEL ARTE

4-.1 ANÁLISIS DEL CONTEXTO

4-.1.1 ANÁLISIS TECNOLÓGICO

En los últimos años, la tecnología ha jugado un papel fundamental en la transformación de diversos sectores, incluido el de la movilidad. El avance hacia un modelo de transporte más sostenible y conectado refleja un cambio significativo en las preferencias de los consumidores y en las regulaciones gubernamentales para combatir la contaminación.

Uno de los indicadores más destacados de esta transición es el aumento en la popularidad de los vehículos híbridos y eléctricos. Según estudios recientes, las ventas de estos vehículos han crecido aproximadamente un 25% en el último año, impulsadas por su menor impacto ambiental.

Este crecimiento evidencia que cada vez más personas optan por alternativas sostenibles, ya sea por razones ecológicas, económicas o de cumplimiento normativo en las zonas de bajas emisiones (ZBE). Teniendo en cuenta esta tendencia, el proyecto incluirá vehículos híbridos y eléctricos en su oferta, ampliando las opciones disponibles para los clientes y fomentando una movilidad más respetuosa con el medio ambiente.

Por otro lado, la tecnología móvil ha revolucionado la forma en que las personas gestionan sus actividades cotidianas. Hoy en día, los teléfonos inteligentes son herramientas indispensables que permiten realizar tareas como compras, pagos y reservas de servicios con facilidad. En este contexto, la propuesta integrará una aplicación móvil diseñada para que los usuarios puedan alquilar vehículos de manera rápida, sencilla y cómoda desde cualquier lugar. Esta aplicación será

intuitiva y contará con funcionalidades como geolocalización para encontrar vehículos cercanos, opciones de reserva personalizada y sistemas de pago seguros, asegurando así una experiencia eficiente y sin contratiempos.

Además, la adopción de tecnologías emergentes, como la conectividad IoT en vehículos, también será un punto clave en este proyecto. Esta tecnología permite un monitoreo en tiempo real del estado del vehículo, garantizando mayor seguridad y optimización del mantenimiento. Asimismo, se evaluará la incorporación de sistemas avanzados de asistencia al conductor (ADAS), que no solo mejoran la experiencia del usuario, sino que también incrementan la seguridad vial.

Por último, el análisis tecnológico no puede ignorar el impacto de las normativas medioambientales y las infraestructuras de recarga eléctrica. Actualmente, la expansión de puntos de recarga para vehículos eléctricos está en auge, lo que facilita la adopción de esta tecnología y la integración de estos vehículos en nuestro modelo de negocio. De igual forma, los vehículos cuentan con las certificaciones y etiquetas medioambientales necesarias para circular en las ZBE, cumpliendo con las regulaciones vigentes y respondiendo a las necesidades de los usuarios.

En conclusión, el proyecto no solo aprovecha el crecimiento de la movilidad sostenible, sino que también utiliza la tecnología móvil y las innovaciones en transporte para ofrecer un servicio moderno, accesible y alineado con las tendencias actuales del mercado.

4-1.2 ANÁLISIS ECONÓMICO

El sector del alquiler de vehículos ha experimentado un crecimiento sostenido en los últimos años, impulsado por factores como el aumento del turismo y la implementación de normativas medioambientales más estrictas en muchas ciudades.

Estas regulaciones limitan el uso de vehículos contaminantes en zonas urbanas, lo que ha llevado a muchas personas a buscar alternativas de movilidad sostenible. Ante la imposibilidad de adquirir un coche híbrido o eléctrico debido a sus elevados costos, el alquiler de vehículos se presenta como una solución económica y práctica para cumplir con estos requisitos.

La demanda de coches ecológicos en el mercado de alquiler continúa en ascenso, especialmente entre aquellos que buscan cumplir con las restricciones medioambientales sin comprometer su movilidad. Este fenómeno ha contribuido al crecimiento constante del sector, especialmente en temporadas de alta afluencia turística. De acuerdo con datos recientes, los turistas representan aproximadamente el 20% del total de alquileres, lo que convierte al turismo en un motor clave de la industria. Durante los veranos previos, se ha alcanzado un total de 820,000 vehículos alquilados, un indicador significativo del dinamismo del mercado.

En este contexto, el sector del alquiler de vehículos enfrenta un panorama económico favorable, pero no exento de desafíos. Según el presidente de la Federación Nacional de Alquiler de Vehículos (Feneval), Juan Luis Barahona, el crecimiento interanual en el número de vehículos en renta durante la temporada estival se sitúa en torno al 20%. Sin embargo, este incremento no ha sido uniforme en todas las regiones, debido a variaciones en la llegada de turistas y la disponibilidad de vehículos. A pesar de esto, la facturación del sector muestra una tendencia positiva, impulsada por una demanda sostenida tanto del turismo como de las empresas.

No obstante, las empresas de alquiler también enfrentan costos elevados derivados de la adquisición de vehículos, los altos tipos de interés y las nuevas regulaciones aplicables al sector. Estos factores han elevado los gastos operativos, lo que puede afectar la rentabilidad a largo plazo. En respuesta, muchas empresas han optado por diversificar su oferta y centrarse en vehículos híbridos y eléctricos para cumplir con las normativas vigentes y satisfacer las expectativas de los consumidores conscientes del medio ambiente.

Desde una perspectiva económica, este proyecto se alinea con las tendencias actuales del mercado, ofreciendo un servicio de alquiler de vehículos que responde a la creciente demanda de alternativas sostenibles. Al incluir opciones de vehículos híbridos y eléctricos, no solo se busca cumplir con las normativas de bajas emisiones, sino también captar un segmento del mercado que prioriza soluciones ecológicas y rentables. Además, la flexibilidad y accesibilidad del servicio, facilitado a través de una plataforma móvil, permite atender tanto a turistas como a residentes que necesitan movilidad temporal sin incurrir en los altos costos de propiedad de un vehículo.

En conclusión, el análisis económico del sector refleja un crecimiento sostenido impulsado por factores estructurales y coyunturales. Este proyecto se posiciona como una solución competitiva y adaptada a las necesidades actuales del mercado, contribuyendo al desarrollo de una movilidad más sostenible y accesible para todos.

4-.1.3 ANÁLISIS SOCIOCULTURAL

En los últimos años, se ha evidenciado una creciente preocupación social por el impacto de nuestras acciones en el medio ambiente. Este cambio ha impulsado la creación de políticas y regulaciones destinadas a mitigar los efectos de la contaminación. En España, un ejemplo destacado es la Ley del Cambio Climático y Transición Energética, que obliga a todas las ciudades y municipios con más de 50.000 habitantes a establecer una Zona de Bajas Emisiones (ZBE) antes de 2023. Estas zonas restringen el acceso, circulación y estacionamiento de vehículos en áreas específicas con el objetivo de mejorar la calidad del aire y reducir las emisiones de gases de efecto invernadero.

El artículo 14.3 de esta ley define las ZBE como áreas delimitadas por una administración pública en las que se aplican restricciones de circulación basadas en la clasificación de los vehículos según su nivel de emisiones, conforme al Reglamento General de Vehículos vigente. Esta normativa afecta a un gran número de localidades, ya que más del 50% de la población española reside en municipios que superan los 50.000 habitantes. Sin embargo, la implementación ha sido desigual, ya que muchas localidades aún no han cumplido con los requisitos establecidos.

COMUNIDADES AUTÓNOMAS Y MUNICIPIOS AFECTADOS

Las ZBE afectan a comunidades autónomas y ciudades de toda España, como:

- Andalucía: Almería, Cádiz (La Línea de la Concepción), Chiclana de la Frontera, Córdoba, Estepona, Granada, Málaga, Marbella, Sevilla, Torremolinos.
- Aragón, Castilla y León, Castilla-La Mancha, Cataluña, Valencia, Galicia, Madrid, Murcia, Navarra, País Vasco.

Estas zonas delimitadas buscan alinear las acciones locales con los objetivos de sostenibilidad a nivel nacional e internacional, promoviendo el uso de vehículos menos contaminantes y, en muchos casos, fomentando alternativas de transporte como bicicletas, transporte público y vehículos eléctricos.

IMPLICACIONES PARA EL PROYECTO

La implementación de estas ZBE subraya la necesidad de soluciones de movilidad sostenible. La regulación limita el acceso de vehículos que no cumplen con los estándares ambientales, lo que genera una mayor demanda de alternativas como el alquiler de vehículos híbridos y eléctricos. Un servicio que ofrezca esta opción no solo facilitaría la movilidad de las personas en áreas restringidas, sino que también respondería a una necesidad generada por las normativas vigentes.

Por otro lado, este contexto también refuerza la importancia de incorporar tecnología en el acceso a estos servicios. Ofrecer una aplicación móvil para la gestión del alquiler de vehículos ecológicos puede mejorar significativamente la experiencia del usuario, permitiendo que acceda rápidamente a soluciones de transporte adaptadas a las ZBE.

En resumen, las ZBE representan no solo un desafío, sino también una oportunidad para promover un modelo de movilidad más limpio y accesible, alineado con los objetivos sociales y ambientales de España y Europa.

4-.3-.4 ANÁLISIS LEGISLATIVO

La lucha contra el cambio climático es uno de los mayores desafíos globales de nuestro tiempo. En este contexto, los gobiernos, organismos internacionales y la sociedad civil han impulsado diversas medidas para abordar las consecuencias del calentamiento global y mitigar sus efectos. Estas acciones incluyen la promoción del uso de energías limpias, la transición hacia modelos de transporte sostenible y la implementación de marcos normativos más estrictos.

En España, la Ley del Cambio Climático y Transición Energética constituye uno de los pilares legislativos en la respuesta a esta problemática. Esta ley no solo busca cumplir con los compromisos adquiridos en el Acuerdo de París de 2015, sino también transformar el modelo económico y social del país hacia uno más sostenible y resiliente. El Acuerdo de París, adoptado por las Naciones Unidas, establece como objetivo principal limitar el calentamiento global a 1,5 °C por encima de los niveles preindustriales. Sin embargo, según informes del Grupo Intergubernamental de Expertos sobre el Cambio Climático (IPCC), este límite podría ser alcanzado entre 2030 y 2052 si no se adoptan medidas inmediatas y contundentes.

El contexto español añade particularidades importantes: el aumento de la temperatura en España supera en casi 0,5 °C la media global, lo que hace que el país sea especialmente vulnerable a los efectos del cambio climático. Estos incluyen sequías más frecuentes, olas de calor intensas y pérdida de biodiversidad. Ante esta realidad, la ley pone énfasis en acciones que no solo reduzcan las emisiones de gases de efecto invernadero, sino que también refuercen la capacidad del país para adaptarse a los cambios climáticos ya inevitables.

PUNTOS CLAVE DE LA LEY DEL CAMBIO CLIMÁTICO Y TRANSICIÓN ENERGÉTICA

ZONAS DE BAJAS EMISIONES (ZBE):

La normativa exige que todas las ciudades españolas con más de 50,000 habitantes establezcan ZBE, áreas donde se restringe el acceso, circulación y estacionamiento de vehículos contaminantes. Este requisito, que debía implementarse antes de 2023, busca fomentar el uso de vehículos eléctricos e híbridos, así como el transporte público y las opciones de movilidad activa como bicicletas y caminatas. A pesar de ello, un porcentaje significativo de ciudades aún no ha cumplido con esta obligación, reflejando la necesidad de acelerar la implementación y fiscalización.

NEUTRALIDAD CLIMÁTICA:

Uno de los objetivos principales de la ley es alcanzar la neutralidad climática antes de 2050. Esto implica que el país debe lograr un balance entre las emisiones generadas y las absorbidas, apostando por tecnologías de captura de carbono, energías renovables y políticas de conservación de ecosistemas naturales que actúan como sumideros de carbono.

IMPULSO A LA INNOVACIÓN Y LA INDUSTRIA:

La transición energética está estrechamente vinculada a la modernización de la industria y al desarrollo de la investigación y la innovación tecnológica. La ley establece medidas para apoyar la competitividad industrial, fomentando la creación de empleos verdes y nuevas oportunidades económicas. Esto incluye inversiones en sectores como la energía solar y eólica, así como en sistemas de almacenamiento de energía.

PROTECCIÓN DE LOS COLECTIVOS VULNERABLES:

La ley reconoce que la transición energética puede tener impactos desiguales en la población. Por ello, establece mecanismos para proteger a los sectores más vulnerables, ofreciendo ayudas y programas de formación para adaptarse a los nuevos modelos económicos y laborales.

CONSERVACIÓN DE LA BIODIVERSIDAD:

Alcanzar los objetivos climáticos no solo implica reducir las emisiones, sino también proteger y restaurar los ecosistemas naturales. Los bosques, humedales y superficies agrícolas desempeñan un papel crucial en la absorción de carbono y en la mitigación de los efectos del cambio climático.

4.3.5 CONTEXTO SOCIAL Y ECONÓMICO

Además del marco legal, es importante considerar la percepción y participación de la ciudadanía. Según un sondeo de Treedom, el 86% de los españoles está preocupado por el futuro del planeta, y el 85% reconoce que podría hacer más para reducir su huella de carbono. Sin embargo, el alto costo de las opciones sostenibles sigue siendo un obstáculo importante. Un 41% de los encuestados afirma que optarían por alternativas más ecológicas si fueran más asequibles.

Otro desafío radica en la implicación personal. Aunque los españoles se muestran preocupados, un 79% no se siente directamente responsable de tomar medidas, lo que refleja la necesidad de campañas de sensibilización más efectivas. En comparación con otros países europeos, España se encuentra en

una posición intermedia: mientras que un 58% de los británicos no se siente implicado, esta cifra asciende al 82% en Italia.

Por otro lado, la implementación de ZBE y otras medidas legislativas está transformando las dinámicas urbanas. Cada vez más ciudadanos optan por medios de transporte alternativos, como bicicletas eléctricas y sistemas de carsharing. Estas transformaciones, aunque positivas, requieren un esfuerzo coordinado entre los gobiernos locales y nacionales para garantizar su efectividad y accesibilidad.

EL DESAFÍO GLOBAL Y EL PAPEL DE ESPAÑA

La lucha contra el cambio climático no es exclusiva de España; es un esfuerzo global que requiere la colaboración de todos los países, sectores y comunidades. Según el Secretario General de las Naciones Unidas, los próximos diez años serán cruciales para evitar consecuencias irreversibles. Las políticas públicas, las inversiones estratégicas y la cooperación internacional serán determinantes para alcanzar los objetivos climáticos.

En el caso de España, la Ley del Cambio Climático y Transición Energética representa un paso importante, pero su éxito dependerá de su implementación efectiva, el cumplimiento de los plazos establecidos y la colaboración de todos los actores involucrados. Además, el país tiene la oportunidad de posicionarse como líder en transición energética, aprovechando sus recursos naturales y su capacidad de innovación.

En conclusión, las leyes sobre cambio climático y movilidad sostenible reflejan un cambio necesario y urgente en nuestra forma de interactuar con el entorno. Si bien existen desafíos significativos, también hay enormes oportunidades para construir una sociedad más equitativa, resiliente y comprometida con la protección del planeta.

4-.2 ANÁLISIS DEL ARTE

4-.2.1. SITUACIÓN ACTUAL DEL MERCADO DE ALQUILER DE COCHES

En los últimos años, el sector del alquiler de vehículos ha experimentado transformaciones significativas, impulsadas por diversos factores que han moldeado tanto la oferta como la demanda. Estas tendencias reflejan cambios en los hábitos de consumo, avances tecnológicos y la evolución del mercado turístico global. A continuación, se destacan los elementos más relevantes:

1. CRECIMIENTO DE LA DEMANDA

El aumento sostenido del turismo ha sido uno de los principales motores del crecimiento en el mercado del alquiler de vehículos. Año tras año, un número creciente de viajeros busca soluciones de movilidad flexibles y cómodas para explorar nuevos destinos, especialmente aquellos donde el transporte público es limitado o poco práctico. Este fenómeno ha generado una expansión en la oferta de servicios de alquiler, con empresas que buscan adaptarse a las necesidades de un público cada vez más diverso.

Además, los viajes de negocios también han contribuido a este crecimiento. Los profesionales necesitan vehículos que les permitan moverse de manera eficiente entre reuniones y eventos, lo que refuerza la demanda de coches de alquiler. En este contexto, las empresas de alquiler han diversificado su flota para incluir desde modelos económicos hasta opciones de lujo que satisfacen diferentes presupuestos y preferencias.

2. AVANCES EN TECNOLOGÍA MÓVIL

La tecnología móvil ha revolucionado la manera en que los clientes interactúan con los servicios de alquiler de coches. Hoy en día, la mayoría de las personas tienen acceso a teléfonos inteligentes que se han convertido en herramientas esenciales para gestionar diversos aspectos de su vida, incluida la movilidad.

Gracias a las aplicaciones móviles y las plataformas en línea, los usuarios pueden comparar precios, elegir el vehículo que mejor se adapte a sus necesidades y realizar reservas en cuestión de minutos, todo desde la comodidad de su hogar o incluso durante el viaje. Este enfoque digital no solo ha hecho que el proceso sea más rápido y accesible, sino que también ha mejorado la experiencia del cliente al ofrecer funcionalidades adicionales, como:

- Opciones de personalización, permitiendo añadir servicios adicionales como seguros, GPS, sillas para niños, entre otros.
- Mapas interactivos que ayudan a localizar puntos de recogida y entrega cercanos.
- Alertas en tiempo real sobre disponibilidad, promociones y ofertas especiales.
- Atención al cliente instantánea a través de chats o asistentes virtuales.

3. ADAPTACIÓN A NUEVOS HÁBITOS DE CONSUMO

El cambio en las prioridades de los consumidores también ha influido en el sector. Cada vez más personas prefieren opciones de alquiler frente a la compra de un vehículo propio, especialmente en las grandes ciudades donde los costos asociados con el mantenimiento, el seguro y el aparcamiento son elevados. Esta tendencia, conocida como economía compartida, refleja una mentalidad más práctica y sostenible, que ve el alquiler como una alternativa más económica y ecológica.

Asimismo, los consumidores modernos valoran la flexibilidad. Empresas de alquiler han comenzado a ofrecer soluciones innovadoras, como alquileres por horas, días o semanas, y opciones de recogida y entrega en múltiples ubicaciones, incluidas estaciones de tren y aeropuertos. Estas iniciativas han ampliado el alcance del mercado, atrayendo tanto a locales como a turistas.

4. SOSTENIBILIDAD Y CAMBIO CLIMÁTICO

El compromiso con la sostenibilidad también ha impactado en el sector del alquiler de vehículos. Las empresas han comenzado a renovar sus flotas para incluir coches eléctricos e híbridos, alineándose con las normativas ambientales y las preferencias de un público cada vez más consciente del impacto ambiental de sus decisiones. Esta transición no solo mejora la percepción de las empresas frente a los consumidores, sino que también contribuye a la reducción de emisiones de carbono en las áreas urbanas.

En conclusión, el mercado del alquiler de coches se encuentra en una etapa de transformación dinámica. La combinación de una creciente demanda turística, avances tecnológicos, cambios en los hábitos de consumo y un enfoque más sostenible ha impulsado el desarrollo del sector.

Para mantenerse competitivas, las empresas deben continuar innovando y adaptándose a las necesidades cambiantes de los clientes, aprovechando las oportunidades que ofrece este panorama en constante evolución.

4-.2.2 ANÁLISIS COMPETENCIA

Existen varias empresas que ofrecen los mismos servicios que prestamos nosotros.

Competencias Directas: Empresas como Hertz, Enterprise y Avis, estas empresas ofrecen grandes variedades de vehículos.

Competencias Locales: Son compañías mas pequeñas, estas se especializan más en ofrecer más atención y se pueden adaptar mas a las necesidades locales.

CARACTERÍSTICAS PRINCIPALES DE LOS PRODUCTOS:

- Variedad de vehículos (eléctricos, híbridos, gasolina etc.)
- Aplicación para reservar fáciles de utilizar.
- Los servicios al cliente están disponibles casi todo el día

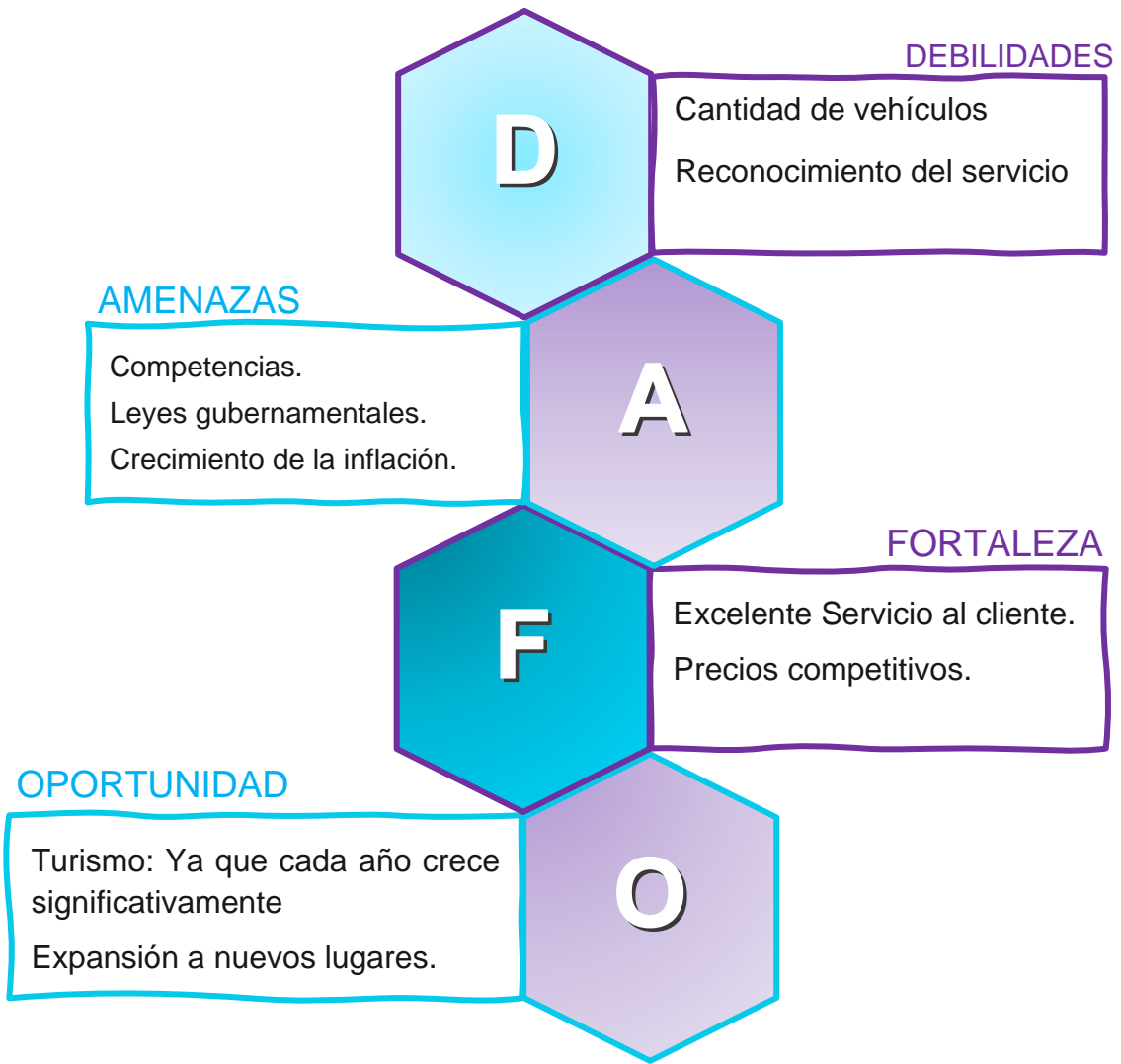
LO MEJOR Y LO PEOR DE LOS PRODUCTOS:

- Lo mejor: Variedad de los vehículos, facilidad de reservar
- Lo peor: Las atenciones a los clientes pueden ser nefastos

LOS PRECIOS:

- Tipo de vehículo: Depende de cual elijas el precio aumentaría o disminuirías. Ya que los coches eléctricos o híbridos serían más caros ya que son más avanzados y tienen menor impacto medio ambiental.
- Ubicación: Dependiendo de donde alquiles los servicios influyen en el precio, en grandes ciudades suelen ser más altos que en ciudades pequeñas debido a la demanda.
- Temporada del Año: La temporada también afecta al precio, si vienes en temporada donde vienen más turistas el precio aumentaría ya que aumentan la demanda estas temporadas serían verano y en algunos casos en invierno ya que es Navidad y la gente visita las grandes ciudades para ver la decoración navideña. Las temporadas donde más barato saldría sería otoño ahí la demanda es menor.

ANÁLISIS DEBILIDADES Y FORTALEZAS (DAFO)



INNOVACIÓN

La innovación es un pilar clave en la aplicación de alquiler de coches, diseñada para ofrecer una experiencia única y adaptada a las necesidades de nuestros futuros clientes. Hemos implementado varias características que diferencian nuestro servicio del resto y que buscan mejorar la accesibilidad, la comodidad y la satisfacción del usuario.

SISTEMA DE PUNTOS Y RECOMPENSAS

Se ha desarrollado un sistema de puntos que premia la fidelidad de los clientes. Cada vez que utilicen el servicio de alquiler, acumularán puntos que podrán canjear en futuras reservas para obtener descuentos en el precio. Este enfoque no solo incentiva el uso recurrente de la aplicación, sino que también crea una relación más cercana con nuestros clientes, generando una experiencia más personalizada y beneficiosa para ellos.

ALQUILER POR HORAS

En un futuro cercano, planeamos introducir la opción de alquiler por horas, una alternativa flexible que responde a las necesidades de quienes no requieren un



coche durante todo el día. Este servicio permitirá a los usuarios disfrutar de tarifas más ajustadas y aprovechar el tiempo que realmente necesiten el vehículo, ampliando nuestra base de clientes y optimizando la disponibilidad de nuestra flota. Esta funcionalidad será particularmente útil en áreas urbanas, donde la demanda por opciones más breves de alquiler es cada vez mayor.

APLICACIÓN MÓVIL INTUITIVA

La eficiencia en las reservas es fundamental para nosotros. Por ello, hemos diseñado una aplicación móvil que centraliza todos nuestros servicios. A través de esta plataforma, los usuarios podrán:

- Reservar un vehículo en pocos pasos, seleccionando entre una amplia variedad de opciones según sus necesidades.
- Gestionar sus reservas, incluyendo modificaciones, extensiones o cancelaciones.
- Acceder al sistema de puntos, consultando su saldo y eligiendo cuándo y cómo canjearlos.
- Disfrutar de promociones exclusivas, con notificaciones personalizadas sobre ofertas y descuentos.
- Localizar puntos de recogida y entrega con mapas interactivos que facilitan la logística del alquiler.

Esta solución tecnológica no solo mejora la experiencia del cliente, sino que también optimiza nuestros procesos internos, permitiendo una mayor agilidad en el servicio y fomentando una comunicación más directa con nuestros usuarios.

VISIÓN HACIA EL FUTURO

Nuestra estrategia de innovación está orientada a satisfacer las necesidades cambiantes de nuestros clientes, garantizando una experiencia moderna, cómoda y eficiente. Estas iniciativas refuerzan nuestro compromiso con la excelencia y aseguran que nuestra aplicación se mantenga como una referencia en el mercado del alquiler de coches.

5-. ANÁLISIS DE REQUISITOS

5-.1 REQUISITOS FUNCIONALES

Requisito Funcional 1	
Nombre del requisito	Registro de clientes
Tipo	Funcional
Descripción del requisito	Permite registrar clientes con la información personal
Prioridad	Alta
Objetivo relacionado	Objetivo 2

Requisito Funcional 2	
Nombre del requisito	Introducción de datos
Tipo	Requisito
Descripción del requisito	Al iniciar la aplicación al principio se verá un video y luego habrá un inicio sesión donde tendrás que meter correo y contraseña y hasta que no cierres sesión no la meterás de nuevo
Prioridad	Alta
Objetivo relacionado	Objetivo 1

Requisito Funcional 3

Nombre del requisito	Reserva de datos
Tipo	Funcional
Descripción del requisito	Los clientes podrán reservar los vehículos
Prioridad	Alta
Objetivo relacionado	Objetivo 5

Requisito Funcional 4

Nombre del requisito	Escribir/Responder comentarios
Tipo	Funcional
Descripción del requisito	Los clientes podrán escribir comentarios y responder comentarios
Prioridad	Alta
Objetivo relacionado	Objetivo 6, Objetivo 7

Requisito Funcional 5

Nombre del requisito	Ver contratos
Tipo	Funcional
Descripción del requisito	El cliente podrá ver los contratos que ha hecho a lo largo del tiempo
Prioridad	Alta
Objetivo relacionado	Objetivo 9

Requisito Funcional 6	
Nombre del requisito	Ver vehículos
Tipo	Funcional
Descripción del requisito	El cliente podrá ver los vehículos que disponemos
Prioridad	Alta
Objetivo relacionado	Objetivo 3, Objetivo 4

Requisito Funcional 7	
Nombre del requisito	Recompensa Puntos
Tipo	Funcional
Descripción del requisito	Los clientes al reservar los vehículos conseguirán puntos por los días que han alquilado
Prioridad	Media
Objetivo relacionado	Objetivo 8

5-.2 REQUISITOS NO FUNCIONALES**Requisito No Funcional 1**

Nombre del requisito	Intermediario de la base de datos
Tipo	No Funcional
Descripción del requisito	Ambas aplicaciones(Escritorio y Móvil) deben contar con una api como intermediario
Prioridad	Alta
Objetivo relacionado	Objetivo 12

Requisito No Funcional 2

Nombre del requisito	Tiempo de respuesta
Tipo	No Funcional
Descripción del requisito	El sistema no debería tardar más de 2 segundos en mostrar los datos de los vehículos
Prioridad	Media
Objetivo relacionado	Objetivo 10

Requisito No Funcional 3

Nombre del requisito	Seguridad de la información
Tipo	No Funcional
Descripción del requisito	La contraseña se cifrará y en la base de datos se guarda cifrada es decir nadie la podrá ver
Prioridad	Media
Objetivo relacionado	Objetivo 11

Requisito No Funcional 4

Nombre del requisito	Disponible en Ingles
Tipo	No Funcional
Descripción del requisito	La aplicación se traducirá al inglés para que todos los usuarios puedan utilizarla
Prioridad	Deseado

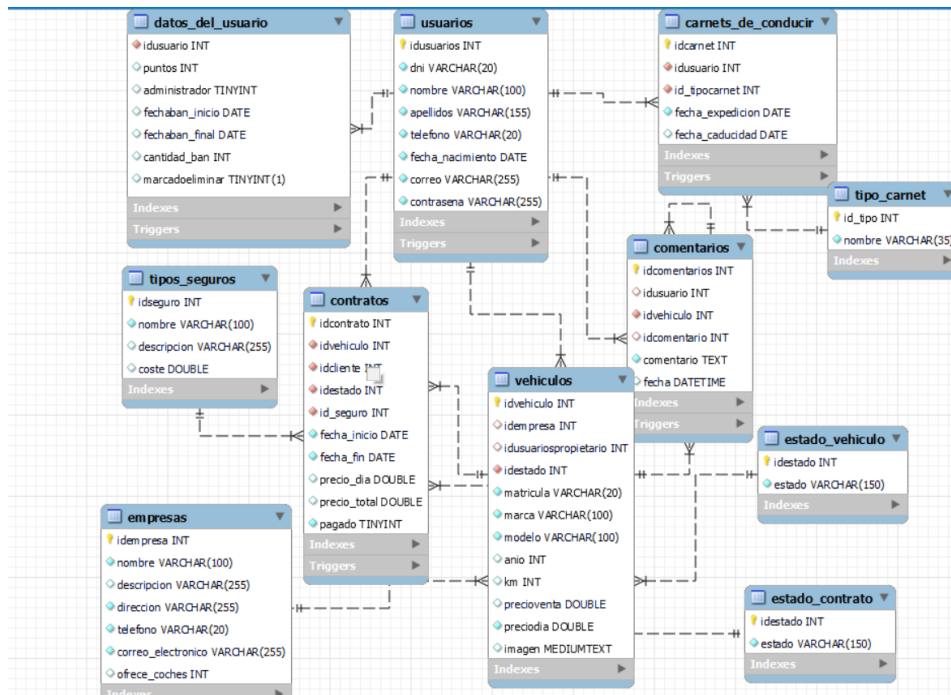
6-. DISEÑO**6.1-. DIAGRAMA FÍSICO Y/O LÓGICO DE RED**

Para llevar a cabo el desarrollo de la aplicación he necesitado hacer:

- Creación de la base de datos: Creación de una base de datos para poder almacenar la información de cada cliente y de nuestros vehículos para el correcto funcionamiento del sistema.
- API: Se ha implementado una api para actuar entre medio de la base de datos y del cliente para facilitar la comunicación.
- Desarrollo de una aplicación de escritorio: Esta aplicación la utilizaran los trabajadores ya que su funcionamiento es para administrar los vehículos, los usuarios, contratos, seguros etc. ...
- Desarrollo de una aplicación de móvil: Esta aplicación la utilizarías todo el mundo que serviría para poder alquilar nuestros servicios

6.2-. Diagrama Relacional de Base de Datos

Para asegurar que los datos estén bien guardados y que podamos saber identificarlos, se ha creado una base de datos con 11 tablas que están relacionadas entre si.



1. TABLA: USUARIOS

- Descripción: Contiene la información principal de los clientes.
- Campos principales:
 - id_usuario (INT, PK): Identificador único para cada usuario.
 - nombre (VARCHAR): Nombre del usuario.
 - apellidos (VARCHAR): Apellidos del usuario.
 - email (VARCHAR): Correo electrónico del usuario.
 - teléfono (VARCHAR): Número de teléfono.

- Restricciones:
 - id_usuario es la clave primaria.
 - email debe ser único para evitar duplicados.
 - Validaciones de formato para email y teléfono.

2. TABLA: CARNETS_DE_CONDUCTIR

- Descripción: Registra los carnets de conducir asociados a cada usuario.
- Campos principales:
 - id_carnet (INT, PK): Identificador único para cada carnet.
 - id_usuario (INT, FK): Relación con la tabla usuarios.
 - fecha_inicio (DATE): Fecha de emisión del carnet.
 - fecha_fin (DATE): Fecha de expiración del carnet.
 - Restricciones:
 - id_usuario es una clave foránea que referencia a usuarios(id_usuario).
 - Asegurar que fecha_fin sea posterior a fecha_inicio.

3. TABLA: TIPO_CARNET

- Descripción: Define los tipos de carnet disponibles (B, A, A1, A2, etc.).
- Campos principales:
 - id_tipo (INT, PK): Identificador único del tipo de carnet.
 - nombre (VARCHAR): Nombre del tipo de carnet.

- Restricciones:
 - id_tipo es la clave primaria.
 - Restricción de unicidad para nombre.

4. TABLA: DATOS_DEL_USUARIO

- Descripción: Almacena información adicional sobre los usuarios, como su estado en el sistema.
- Campos principales:
 - id_usuario (INT, PK, FK): Relación con la tabla usuarios.
 - puntos (INT): Puntos acumulados por el usuario en el sistema.
 - baneado (TINYINT): Indicador de si el usuario está baneado (1 = sí, 0 = no).
 - Restricciones:
 - id_usuario referencia a usuarios(id_usuario).
 - Validación para que puntos sea un valor positivo.

5. TABLA: COMENTARIOS

- Descripción: Permite a los usuarios dejar comentarios sobre los vehículos alquilados y responder a los de otros.
- Campos principales:
 - id_comentario (INT, PK): Identificador único para cada comentario.
 - id_usuario (INT, FK): Usuario que realiza el comentario.
 - id_vehiculo (INT, FK): Vehículo sobre el que se comenta.
 - contenido (TEXT): Texto del comentario.
 - Restricciones:
 - id_usuario referencia a usuarios(id_usuario).

- id_vehiculo referencia a vehículos(id_vehiculo).
- Limitar la longitud máxima de los comentarios si es necesario.

6. TABLA: EMPRESAS

- Descripción: Almacena información de las empresas colaboradoras o proveedoras.
- Campos principales:
 - id_empresa (INT, PK): Identificador único de la empresa.
 - nombre (VARCHAR): Nombre de la empresa.
 - dirección (TEXT): Dirección física de la empresa.
 - Restricciones:
 - id_empresa es la clave primaria.
 - Restricción de unicidad en el nombre.

7. TABLA: VEHÍCULOS

- Descripción: Contiene información sobre todos los vehículos disponibles.
- Campos principales:
 - id_vehiculo (INT, PK): Identificador único del vehículo.
 - marca (VARCHAR): Marca del vehículo.
 - modelo (VARCHAR): Modelo del vehículo.
 - precio_diario (DOUBLE): Costo por día de alquiler.
 - Restricciones:
 - id_vehiculo es la clave primaria.
 - Validación para que precio_diario sea mayor que cero.

8. TABLA: ESTADO_VEHICULO

- Descripción: Describe el estado actual de un vehículo (disponible, alquilado, estropeado, etc.).
- Campos principales:
 - id_estado (INT, PK): Identificador único del estado.
 - estado (VARCHAR): Descripción del estado (ej., "disponible").
 - Restricciones:
 - id_estado es la clave primaria.
 - Validar que los estados sean valores predefinidos.

9. TABLA: TIPOS_SEGUROS

- Descripción: Detalla los diferentes tipos de seguros disponibles al realizar un contrato.
- Campos principales:
 - id_seguro (INT, PK): Identificador único del seguro.
 - descripcion (TEXT): Detalles del seguro.
 - costo (DOUBLE): Costo asociado al seguro.
 - Restricciones:
 - Validar que costo sea mayor que cero.

10. TABLA: CONTRATOS

- Descripción: Registra todos los contratos de alquiler realizados.
- Campos principales:
 - id_contrato (INT, PK): Identificador único del contrato.
 - id_usuario (INT, FK): Usuario que realiza el contrato.
 - id_vehiculo (INT, FK): Vehículo alquilado.
 - fecha_inicio (DATE): Fecha de inicio del alquiler.

- fecha_fin (DATE): Fecha de finalización del alquiler.
- Restricciones:
 - fecha_fin debe ser posterior a fecha_inicio.
 - id_usuario y id_vehiculo deben existir en sus respectivas tablas.

11. TABLA: ESTADO_CONTRATO

- Descripción: Indica el estado actual del contrato (activo, finalizado, cancelado).
- Campos principales:
 - id_estado (INT, PK): Identificador único del estado.
 - estado (VARCHAR): Descripción del estado (ej., "activo").
 - Restricciones:
 - id_estado es la clave primaria.
 - Validar que los estados sean valores predefinidos

6.3-. Diagramas de Clases

6.3.1-. USUARIOS

La tabla Usuarios en una base de datos es una estructura diseñada para almacenar la información de los usuarios que interactúan con el sistema. La información que se guarda incluye:

1. idusuarios: Este es el identificador único de cada usuario y se utiliza como la clave primaria de la tabla. Esto asegura que cada usuario tenga un registro único que puede ser identificado de manera inequívoca.

2. dni: El Documento Nacional de Identidad (DNI) es un identificador personal de cada usuario que permite diferenciarlos en contextos más amplios y, a menudo, sirve como método de autenticación.

3. nombre y apellidos: Estos campos almacenan el nombre y los apellidos del usuario, permitiendo identificar a las personas de manera más comprensible para los humanos.

4. teléfono: Este campo guarda el número de contacto del usuario, que puede ser utilizado para comunicaciones o notificaciones.

5. fecha nacimiento: Permite guardar la fecha de nacimiento del usuario. Este dato puede ser utilizado para validar la mayoría de edad, proporcionar servicios personalizados o cualquier otra función que dependa de la edad del usuario.

6. correo: El correo electrónico del usuario, que generalmente se utiliza para la autenticación, la recuperación de contraseñas o las comunicaciones oficiales.

7. contraseña: Este campo almacena una versión encriptada de la contraseña del usuario. Las contraseñas nunca se guardan en texto plano para garantizar la seguridad de la información personal de los usuarios.

RELACIONES CON OTRAS TABLAS

La tabla Usuarios está relacionada con varias otras tablas dentro de la base de datos, lo que indica su importancia dentro del sistema. Estas relaciones incluyen:

- Carnets de conducir: Cada usuario puede estar asociado con uno o varios carnets de conducir.
- Datos del usuario: Aquí se almacena información adicional sobre el estado del usuario, como si es administrador, si está baneado, su puntuación, entre otros.

- Vehículos: Los usuarios pueden ser propietarios de vehículos o estar relacionados con ellos de otras maneras, como a través de contratos.
 - Contratos: Los usuarios pueden ser clientes en contratos de alquiler de vehículos.
 - Comentarios: Los usuarios pueden dejar comentarios sobre vehículos u otros elementos.
-

MÉTODOS ASOCIADOS

Los métodos asociados a la tabla Usuarios permiten interactuar con los datos de esta tabla, ya sea para crear, leer, actualizar o eliminar registros. Estos métodos son fundamentales para implementar las funcionalidades del sistema.

1. ObtenerUsuariosCorreoContraseña:

- Este método toma como parámetros el correo y la contraseña de un usuario.
- Devuelve un registro que contiene todos los atributos del usuario si las credenciales son correctas.
- Su uso principal es en el proceso de inicio de sesión. Por ejemplo, cuando un usuario introduce su correo y contraseña en un formulario, este método verifica si las credenciales coinciden con un registro existente y devuelve los datos correspondientes.

2. ObtenerUsuarioPorId:

- Este método recibe como parámetro el identificador único (idusuarios) de un usuario y devuelve toda la información asociada a ese usuario.
- Es útil cuando se necesita acceder a los datos de un usuario específico, como al mostrar un perfil en detalle.

3. ObtenerUsuarios:

- Este método devuelve todos los registros de la tabla de usuarios.

- Se utiliza principalmente en aplicaciones administrativas o de escritorio, donde un administrador necesita visualizar la lista completa de usuarios registrados en el sistema.

4. ObtenerUsuarioPorDni:

- Este método permite buscar un usuario específico utilizando su DNI como parámetro.
- Es especialmente útil en escenarios donde el DNI es un identificador relevante, como en procesos legales o de identificación personal.

5. ObtenerCorreoAndContrasenaAdmin:

- Este método funciona de manera similar a ObtenerUsuariosCorreoContraseña, pero está diseñado específicamente para verificar las credenciales de los administradores.
- Si un administrador introduce su correo y contraseña en el sistema, este método valida las credenciales y permite el acceso a funcionalidades exclusivas del rol administrativo.

6. ModificarUsuario:

- Este método se utiliza para actualizar los datos de un usuario existente.
- Recibe como parámetros los datos del usuario que se desean modificar (por ejemplo, cambiar el correo, el teléfono o la contraseña) y realiza los cambios correspondientes en la base de datos.

7. CrearUsuario:

- Este método se utiliza para crear un nuevo registro de usuario.
- Es utilizado durante el proceso de registro, cuando un nuevo usuario completa un formulario para crear una cuenta. Los datos proporcionados se guardan en la tabla Usuarios.

8. EliminarUsuario:

- Este método permite eliminar un registro de usuario de la base de datos.

- Recibe como parámetro el idusuarios del usuario que se desea eliminar.
- Se utiliza principalmente en aplicaciones administrativas para gestionar cuentas de usuarios, como en casos de usuarios que ya no forman parte del sistema.

EJEMPLO DE USO PRÁCTICO

1) Inicio de sesión de un usuario común:

- El usuario introduce su correo y contraseña en la interfaz de la aplicación.
- El sistema llama al método ObtenerUsuariosCorreoContraseña con estos datos.
- Si las credenciales son válidas, el sistema recupera los datos del usuario y permite el acceso.

2) Gestión de usuarios en una aplicación administrativa:

- Un administrador accede a la aplicación de escritorio.
- El sistema utiliza ObtenerUsuarios para cargar y mostrar la lista completa de usuarios.
- Si el administrador selecciona un usuario, el sistema puede utilizar ObtenerUsuarioPorId para mostrar información detallada.
- El administrador puede actualizar los datos de un usuario utilizando UpdateUsuario o eliminarlo con DeleteUsuario.

3) Registro de un nuevo usuario:

- Un nuevo usuario completa un formulario de registro.
- El sistema llama a SaveUsuario para guardar los datos en la base de datos.

RESUMEN

En resumen, la tabla Usuarios es el núcleo de la gestión de identidades en el sistema. Está diseñada para almacenar información fundamental de los usuarios y se relaciona con múltiples aspectos del sistema. Los métodos asociados proporcionan las herramientas necesarias para interactuar con estos datos y ofrecer funcionalidades tanto a los usuarios finales como a los administradores del sistema.

6.3.2-. CREAR CARNETS DE CONDUCIR

La tabla Carnets de Conducir está diseñada para almacenar la información relacionada con los permisos de conducir de los usuarios. La información contenida en esta tabla incluye:

- 1) idcarnet: Es el identificador único para cada carnet de conducir, y se utiliza como clave primaria. Esto asegura que cada carnet tiene un registro único dentro de la tabla.
- 2) idusuario: Este es un identificador que se refiere al usuario propietario del carnet. Está relacionado con la tabla Usuarios y permite asociar un carnet con un usuario específico. Este campo es una clave foránea que enlaza a la tabla de Usuarios.
- 3) id_tipocarnet: Es un identificador que hace referencia al tipo de carnet (por ejemplo, carnet de conducir de automóvil, motocicleta, etc.). Está relacionado con la tabla Tipo_Carnet y permite asociar un carnet con un tipo específico de licencia.
- 4) fecha_expedicion: Almacena la fecha en que el carnet fue expedido. Es un dato crucial para saber cuándo un carnet fue emitido y, posiblemente, calcular su caducidad.
- 5) fecha_caducidad: Guarda la fecha en la que el carnet de conducir caduca. Este campo es esencial para gestionar la validez del carnet y determinar cuándo un usuario debe renovarlo.

RELACIONES CON OTRAS TABLAS

La tabla Carnets de Conducir está relacionada con las siguientes tablas:

- Usuarios: Cada carnet de conducir está asociado a un usuario específico a través del campo idusuario. Esto implica que un usuario puede tener uno o varios carnets de conducir, dependiendo del tipo de licencia que haya obtenido.

- Tipo_Carnet: Este campo id_tipocarnet se relaciona con la tabla Tipo_Carnet, lo que permite identificar el tipo específico de carnet (por ejemplo, tipo A, tipo B, etc.).

MÉTODOS ASOCIADOS

Los métodos asociados a esta tabla permiten realizar operaciones de lectura, actualización, inserción y eliminación sobre los registros de Carnets de Conducir. Estos métodos son esenciales para interactuar con la tabla y gestionar los carnets de conducir de los usuarios. A continuación, se describe cada uno de estos métodos:

1) ObtenerCarnets:

- Este método devuelve todos los registros de la tabla Carnets de Conducir.
- Se utiliza, por ejemplo, en la aplicación de escritorio donde se necesita mostrar todos los carnets existentes.
- Es útil para los administradores o para consultas generales dentro del sistema.

2) ObtenerCarnetsPorId:

- Este método recibe como parámetro el idcarnet (identificador único de un carnet de conducir).
- Devuelve los detalles del carnet asociado con ese id.
- Se utiliza cuando se necesita obtener un carnet específico, por ejemplo, para ver los detalles de un carnet en particular.

3) ObtenerCarnetsPorIdUsuarios:

- Este método toma como parámetro el idusuario y devuelve todos los carnets asociados a ese usuario.
- Es útil cuando se quiere obtener todos los carnets de conducir que un usuario posee. Un usuario puede tener múltiples carnets si tiene distintos tipos de licencias.

4) ModificarCarnet:

- Este método permite actualizar los datos de un carnet de conducir existente.
- Recibe los nuevos valores para el carnet (por ejemplo, nueva fecha de caducidad, o si cambia el tipo de carnet) y actualiza el registro correspondiente en la base de datos.
- Se puede utilizar, por ejemplo, cuando un usuario renueva su carnet o si hay un error en la información almacenada.

5) CrearCarnet:

- Este método se utiliza para crear un nuevo registro de carnet de conducir en la base de datos.
- Recibe los datos necesarios para crear un carnet (como el idusuario, id_tipocarnet, fecha_expedicion, fecha_caducidad, etc.) y los guarda en la base de datos.
- Se usa cuando un nuevo usuario obtiene un carnet o cuando un usuario renueva su carnet.

6) EliminarCarnet:

- Este método elimina un carnet de conducir de la base de datos.
- Recibe como parámetro el idcarnet y elimina el registro correspondiente.
- Es útil cuando se necesita borrar un carnet, por ejemplo, si un carnet es inválido o si un usuario lo pierde y no lo va a renovar.

EJEMPLO DE USO PRÁCTICO

1. Obtener todos los carnets en una aplicación administrativa:

- Un administrador necesita revisar todos los carnets de conducir registrados en el sistema.
- Utiliza el método ObtenerCarnets, que devolverá todos los registros de la tabla y los mostrará en una interfaz de administración.

2. Obtener los carnets de un usuario específico:

- Un usuario tiene varios carnets (por ejemplo, uno para conducir automóviles y otro para motocicletas).

- El sistema usa `ObtenerCarnetsPorIdUsuarios` pasando el `idusuario` del usuario para recuperar todos los carnets asociados a esa persona.

3. Actualizar un carnet:

- Un usuario renueva su carnet, lo que implica actualizar su fecha de caducidad.
- El sistema usa `UpdateCarnet` pasando el `idcarnet` y la nueva fecha de caducidad para actualizar el carnet en la base de datos.

4. Crear un nuevo carnet:

- Un usuario acaba de obtener su carnet de conducir.
- El sistema usa `SaveCarnet` para crear un nuevo registro en la base de datos, asociando el nuevo carnet con el `idusuario` correspondiente y con el tipo de carnet adecuado.

5. Eliminar un carnet:

- Un carnet de conducir ya no es válido o ha sido cancelado por alguna razón (por ejemplo, el usuario lo perdió y no lo renovó).
- El sistema usa `DeleteCarnet` pasando el `idcarnet` para eliminar el registro del carnet en la base de datos.

RESUMEN

La tabla Carnets de Conducir es esencial para gestionar la información de los permisos de conducir de los usuarios dentro del sistema. Contiene datos clave como el identificador del carnet, el usuario al que pertenece, el tipo de carnet y las fechas de expedición y caducidad. Los métodos asociados permiten obtener, actualizar, crear y eliminar carnets de conducir, lo que facilita la gestión de los permisos de los usuarios en la plataforma. Estos métodos son utilizados en diversas situaciones, desde la creación de nuevos carnets hasta la gestión administrativa de los existentes.

6.3.3- TIPOS_CARNET DE CONDUCIR

La tabla Tipos_Carnet de Conducir está diseñada para almacenar los diferentes tipos de permisos de conducir disponibles en el sistema. Cada tipo de carnet se caracteriza por un identificador único y una descripción del tipo de carnet (como A, B, A1, A2, etc.).

ESTRUCTURA DE LA TABLA

- 1) idtipo: Es la clave primaria de la tabla y se utiliza para identificar de forma única cada tipo de carnet. Este campo es un valor numérico o alfanumérico que representa un tipo específico de carnet de conducir.
- 2) tipo_carnet: Este campo contiene el tipo o la categoría del carnet. Se almacena como una cadena de texto que puede ser, por ejemplo, 'A', 'B', 'A1', 'A2', etc., dependiendo de las diferentes categorías de licencias que maneje el sistema.

RELACIONES CON OTRAS TABLAS

La tabla Tipos_Carnet de Conducir está relacionada principalmente con la tabla Carnets de Conducir, donde cada carnet de conducir debe estar asociado con un tipo de carnet. Esta relación se establece a través del campo id_tipocarnet en la tabla Carnets de Conducir, que actúa como una clave foránea que referencia el idtipo de la tabla Tipos_Carnet.

MÉTODOS ASOCIADOS A LA TABLA

Los métodos de esta tabla permiten interactuar con los tipos de carnet de conducir, realizando operaciones de lectura, actualización, inserción y eliminación. A continuación, se detallan los métodos disponibles:

1. ObtenerTiposCarnet:

- Este método devuelve todos los tipos de carnet registrados en la tabla Tipos_Carnet.
- Es útil cuando se necesita obtener una lista completa de todos los tipos de carnet disponibles en el sistema. Este método se

utilizaría, por ejemplo, en una interfaz administrativa donde se muestren todos los tipos de carnet para poder seleccionar o gestionar alguno de ellos.

2. ObtenerTiposCarnetsPorId:

- Este método recibe como parámetro el idtipo (identificador del tipo de carnet) y devuelve los detalles del tipo de carnet correspondiente a ese identificador.
- Se utiliza cuando se necesita obtener información específica sobre un tipo de carnet, por ejemplo, cuando se quiere mostrar los detalles de un tipo de carnet particular en una interfaz.

3. ObtenerTipoCarnetPorTipo:

- Este método toma como parámetro el nombre del tipo de carnet (por ejemplo, 'B') y devuelve el idtipo y el tipo carnets correspondientes a ese tipo.
- Es útil cuando se necesita consultar la información asociada a un tipo específico de carnet, por ejemplo, cuando un usuario o administrador busca un tipo de carnet en particular por su nombre.

4. ModificarTipoCarnet:

- Este método permite actualizar un tipo de carnet de conducir existente.
- Recibe el idtipo y los nuevos valores para actualizar el tipo de carnet (por ejemplo, cambiar la descripción del tipo de carnet o corregir alguna información).
- Es útil cuando se necesita actualizar la descripción de un tipo de carnet, por ejemplo, si se cambian las categorías de las licencias o se corrige un error.

5. CrearTipoCarnet:

- Este método permite crear un nuevo tipo de carnet de conducir en la base de datos.
- Recibe los datos necesarios, como el nombre o la categoría del carnet (por ejemplo, 'A', 'B', 'A1', etc.), y los guarda en la tabla.
- Se utiliza cuando se desea añadir un nuevo tipo de carnet a la base de datos, por ejemplo, cuando se introduce una nueva categoría de licencias en el sistema.

6. EliminarTipoCarnet:

- Este método elimina un tipo de carnet de conducir de la base de datos.
- Recibe como parámetro el idtipo y elimina el tipo de carnet correspondiente.
- Se utiliza cuando un tipo de carnet debe ser eliminado del sistema, por ejemplo, si una categoría de licencia ya no está disponible o si se han cometido errores en la base de datos que requieren su eliminación.

EJEMPLOS DE USO PRÁCTICO

1) Obtener todos los tipos de carnet:

- En una aplicación administrativa, un administrador desea ver todos los tipos de carnet disponibles en el sistema (como 'A', 'B', 'A1', etc.).
- Utiliza el método ObtenerTiposCarnet, que devolverá una lista de todos los tipos de carnet registrados en la base de datos.

2) Obtener un tipo de carnet específico por su id:

- Un usuario o administrador quiere obtener detalles sobre el tipo de carnet con id 3.
- Se utiliza el método ObtenerTiposCarnetsPorId(3), que devolverá la información sobre ese tipo de carnet.

3) Obtener el id y el tipo de carnet por el nombre del tipo:

- Un administrador busca el tipo de carnet con el nombre 'B'.

- Se utiliza el método `ObtenerTipoCarnetPorTipo('B')`, que devolverá el idtipo y la descripción ('B') del tipo de carnet.
- 4) Actualizar un tipo de carnet:
- Un tipo de carnet necesita ser actualizado (por ejemplo, se cambia el nombre o la categoría).
 - El administrador o el sistema utiliza el método `UpdateTipoCarnet` para actualizar el tipo de carnet correspondiente. Por ejemplo, podría cambiarse 'A1' a una nueva categoría con nombre actualizado.
- 5) Crear un nuevo tipo de carnet:
- El sistema introduce un nuevo tipo de carnet, como 'C' para una categoría de licencia de camión.
 - Se utiliza el método `SaveTipoCarnet` para agregar el nuevo tipo de carnet a la base de datos.
- 6) Eliminar un tipo de carnet:
- Un tipo de carnet ya no es válido o se ha eliminado del sistema (por ejemplo, la categoría 'A1' ya no es válida).
 - El administrador o el sistema utiliza `DeleteTipoCarnet` para eliminar este tipo de carnet de la base de datos.

RESUMEN

La tabla `Tipos_Carnet de Conducir` es fundamental para gestionar las diferentes categorías o tipos de licencias que existen dentro del sistema. Cada tipo de carnet tiene un identificador único y un nombre que describe la categoría del carnet. Los métodos asociados a esta tabla permiten obtener, actualizar, crear y eliminar tipos de carnet, lo que facilita la gestión de las categorías de licencias disponibles para los usuarios. Estos métodos son esenciales para mantener actualizada la base de datos de tipos de carnet y garantizar que los usuarios y administradores puedan interactuar de forma eficiente con la información de los tipos de licencia.

6.3.4-. DATOS DEL USUARIO

La tabla Datos del Usuario es una tabla crucial para almacenar información adicional sobre los usuarios, que no está directamente relacionada con los datos básicos como nombre, apellidos o correo electrónico, pero sí con información sobre su comportamiento dentro del sistema, como los puntos que tienen, su rol de administrador, su estado de baneo y otros atributos relacionados con la administración de usuarios.

ESTRUCTURA DE LA TABLA

1) idusuario:

- Es la clave principal de la tabla, y se refiere al idusuario que también existe en la tabla Usuarios. Este campo actúa como una clave foránea que establece una relación entre la tabla Datos del Usuario y la tabla Usuarios. Cada usuario tiene un conjunto específico de datos relacionados con su comportamiento, como los puntos, su estado de administrador, si está baneado, etc.

2) puntos:

- Este campo almacena la cantidad de puntos que tiene un usuario. Podría ser una puntuación obtenida de alguna actividad dentro de la aplicación, como por ejemplo un sistema de recompensas o un sistema de reputación.

3) administrador:

- Este campo almacena un valor que indica si el usuario es un administrador del sistema. El valor podría ser un booleano (verdadero o falso), donde true indica que el usuario tiene privilegios administrativos y false indica que no los tiene.

4) fechaban_inicio:

- Este campo almacena la fecha de inicio del periodo de baneo de un usuario, en caso de que esté baneado. Es una fecha que indica cuándo comenzó el baneo.

5) fechaban_final:

- Similar al campo fechaban_inicio, este campo almacena la fecha de finalización del baneo. Indica cuándo termina el periodo de

baneo del usuario, después del cual se le podrá reactivar en el sistema.

6) cantidad_ban:

- Este campo almacena la cantidad de veces que un usuario ha sido baneado. Esto podría ser útil para llevar un control del comportamiento de los usuarios y decidir si un baneo repetido tendrá alguna implicación adicional (por ejemplo, un baneo permanente tras varios baneos).

7) marcadoEliminar:

- Este campo almacena un valor que indica si el usuario está marcado para ser eliminado del sistema. Puede ser un valor booleano donde true indica que el usuario está marcado para eliminación, pero todavía no ha sido eliminando, y false indica lo contrario. Esto podría ser útil para manejar el proceso de eliminación de usuarios de forma controlada.

RELACIONES CON OTRAS TABLAS

La tabla Datos del Usuario tiene una relación directa con la tabla Usuarios mediante el campo idusuario, que es una clave foránea que hace referencia a la clave primaria idusuarios de la tabla Usuarios. Esta relación permite vincular los datos adicionales del usuario (como los puntos, su rol de administrador y su estado de baneo) con los datos personales del usuario (nombre, correo, teléfono, etc.).

Métodos asociados a la tabla

Los métodos proporcionados permiten gestionar la información de los usuarios, actualizar su estado, y realizar acciones de baneo o eliminación. A continuación, se detallan los métodos disponibles:

1) ObtenerDatosDelUsuarioPorId:

- Este método recibe el idusuario como parámetro y devuelve todos los datos del usuario relacionados con ese identificador, como los puntos, el estado de administrador, la fecha de inicio y finalización del baneo, etc.
- Se utiliza cuando se necesita obtener información detallada sobre un usuario específico. Por ejemplo, en una interfaz de administración, cuando se hace clic en un usuario, este método puede ser llamado para mostrar su perfil completo con los datos adicionales.

2) ObtenerDatosDelUsuarios:

- Este método devuelve todos los datos de los usuarios guardados en la tabla Datos del Usuario.
- Es útil en una aplicación administrativa de escritorio cuando se desea ver la información de todos los usuarios y su estado (puntos, si están baneados, si son administradores, etc.).

3) ModificarDatosDelUsuario:

- Este método permite actualizar los datos del usuario en la tabla Datos del Usuario. Se puede utilizar para actualizar atributos como los puntos del usuario, su estado de administrador, o incluso las fechas de baneo.
- Este método se puede usar, por ejemplo, cuando un administrador desee cambiar los puntos de un usuario, actualizar la fecha de fin de baneo o cambiar su rol de administrador.

4) BanearUsuario:

- Este método sirve para banear a un usuario. Recibe el idusuario como parámetro y establece la fecha de inicio del baneo, así como la fecha final de baneo.
- Es útil cuando un administrador desea prohibir temporalmente a un usuario acceder al sistema debido a un comportamiento inapropiado o violaciones de las reglas. Se puede establecer la duración del baneo para que el usuario no pueda usar la plataforma durante ese periodo.

5) MarcarEliminar:

- Este método se usa para marcar un usuario para eliminación. Recibe el idusuario y cambia el campo marcadoEliminar a true, indicando que el usuario está en proceso de eliminación, aunque todavía no se haya eliminado físicamente de la base de datos.
- Es útil cuando se necesita manejar la eliminación de un usuario de forma gradual, lo que puede ser necesario para garantizar que la eliminación se realice correctamente o para tener una opción de deshacer en caso de que la eliminación sea accidental.

EJEMPLOS DE USO PRÁCTICO

1- Obtener datos adicionales de un usuario:

- En una interfaz administrativa, un administrador puede querer ver los puntos, el rol de administrador y si un usuario está baneado.
- Se utilizaría el método ObtenerDatosDelUsuarioPorId pasando el idusuario del usuario deseado para obtener su información.

2- Obtener todos los datos de los usuarios:

- El administrador de un sistema quiere consultar el estado de todos los usuarios, para ver, por ejemplo, quiénes están baneados o quiénes son administradores.
- Se utilizaría el método ObtenerDatosDelUsuarios, que devolvería una lista con todos los usuarios y sus detalles.

3- Actualizar el estado de un usuario:

- Un administrador decide actualizar la cantidad de puntos de un usuario o cambiar su estado de baneo.
- El administrador usa el método UpdateDatosDelUsuario para actualizar la información del usuario según sea necesario.

4- Banear a un usuario:

- Un usuario ha violado las reglas del sistema y un administrador decide banearlo por 30 días.
- El administrador usa el método BanearUsuario pasando el idusuario y configurando las fechas de inicio y fin del baneo.

5- Marcar a un usuario para eliminación:

- Un usuario ha solicitado eliminar su cuenta, o un administrador decide eliminar un usuario por alguna razón.
- Se usa el método MarcarEliminar para marcar al usuario para eliminación, sin eliminarlo inmediatamente, lo que permite realizar una revisión o una reversión de la eliminación si es necesario.

RESUMEN

La tabla Datos del Usuario es fundamental para gestionar la información adicional sobre los usuarios, como los puntos, su rol de administrador, y su estado de baneo. Los métodos proporcionados permiten gestionar y actualizar estos datos de manera eficiente, permitiendo una administración robusta de los usuarios, con funciones para banear, eliminar o actualizar la información relacionada con su comportamiento dentro del sistema. Esta tabla también ofrece la capacidad de llevar un control sobre la actividad de los usuarios y permitir una gestión dinámica de los accesos y privilegios.

6.3.5-. COMENTARIOS

La tabla Comentarios almacena los comentarios realizados por los usuarios sobre vehículos y las respuestas a dichos comentarios. Este sistema permite que los usuarios interactúen con la plataforma, compartiendo sus opiniones sobre los vehículos disponibles y respondiendo a los comentarios de otros usuarios. A continuación, te explico en detalle cómo se estructura esta tabla y los métodos asociados que permiten gestionar los comentarios y las respuestas.

ESTRUCTURA DE LA TABLA

La tabla Comentarios tiene los siguientes atributos:

1) idcomentario:

- Este es el identificador único de cada comentario. Es la clave principal de la tabla y permite distinguir cada comentario de los demás. Es utilizado para realizar operaciones de obtención, actualización o eliminación de un comentario específico.

2) idusuario:

- Este campo hace referencia al idusuario que realizó el comentario. Es una clave foránea que vincula el comentario con la tabla Usuarios, permitiendo identificar quién hizo el comentario.

3) idvehiculo:

- Este campo hace referencia al idvehiculo sobre el que se realiza el comentario. Es una clave foránea que se conecta con la tabla Vehículos, indicando el vehículo al que corresponde el comentario. Este campo es importante para organizar los comentarios por vehículo.

4) idcomentariopadre:

- Este campo hace referencia al idcomentario de un comentario al que se está respondiendo. Es una clave foránea que crea una relación entre los comentarios y sus respuestas. Si un comentario no es una respuesta a otro, este campo puede ser nulo. Si un comentario es una respuesta, el idcomentariopadre almacena el idcomentario del comentario original al que se está respondiendo.

5) comentario:

- Este campo almacena el texto del comentario en sí. Contiene lo que el usuario ha escrito sobre el vehículo o como respuesta a otro comentario.

6) fecha:

- Este campo almacena la fecha en que se realizó el comentario. Se utiliza para ordenar los comentarios según la fecha de creación, permitiendo ver los comentarios más recientes primero.

RELACIONES CON OTRAS TABLAS

La tabla Comentarios se relaciona con varias otras tablas del sistema:

- Usuarios: Cada comentario está asociado a un usuario específico que lo ha realizado. Esta relación se establece a través del campo idusuario.
- Vehículos: Los comentarios están asociados a vehículos específicos, lo que permite que los usuarios dejen opiniones sobre un vehículo en particular. La relación se establece a través del campo idvehiculo.
- Comentarios (respuestas a comentarios): La tabla también permite que los usuarios respondan a otros comentarios. Esto se logra mediante el campo idcomentariopadre, que crea una relación jerárquica entre los comentarios y sus respuestas.

MÉTODOS ASOCIADOS A LA TABLA

Los métodos proporcionados permiten realizar diversas operaciones con los comentarios, como obtener comentarios por su id, asociarlos con vehículos, o actualizarlos. A continuación, se detallan los métodos disponibles:

1. ObtenerComentariosPorId:

- Este método recibe el idcomentario como parámetro y devuelve los datos completos del comentario correspondiente a ese id. Esto es útil cuando se necesita acceder a un comentario específico, ya sea para mostrarlo en detalle o para realizar alguna operación con él, como su eliminación o actualización.

2. ObtenerComentariosPorIdVehiculo:

- Este método recibe el idvehiculo y devuelve todos los comentarios asociados a ese vehículo en particular. Esto permite a los usuarios ver todas las opiniones que se han dado sobre un vehículo, lo que

puede ser útil, por ejemplo, para un perfil de vehículo en la aplicación o página web.

3. `ObtenerComentariosPorIdRespuesta`:

- Este método recibe el `idcomentariopadre` (el id del comentario original) y devuelve los comentarios que son respuestas a ese comentario específico. Esto permite acceder a todas las respuestas de un comentario dado, manteniendo la relación entre comentarios y respuestas.

4. `ObtenerComentarios`:

- Este método devuelve todos los comentarios almacenados en la tabla `Comentarios`. En una aplicación de escritorio, por ejemplo, puede utilizarse para mostrar una lista de todos los comentarios realizados por los usuarios, posiblemente con filtros para buscar comentarios de un vehículo específico o de un usuario determinado.

5. `ModificarComentarios`:

- Este método permite actualizar un comentario existente. El usuario puede editar su propio comentario (por ejemplo, corregir un error tipográfico o actualizar su opinión). El método recibe el `idcomentario` y el nuevo texto del comentario, y actualiza la entrada correspondiente en la tabla.

6. `CrearComentarios`:

- Este método se utiliza para crear un nuevo comentario. Se pasa el `idusuario`, `idvehiculo`, `idcomentariopadre` (si es una respuesta), el texto del comentario y la fecha. El método inserta un nuevo comentario en la base de datos. Es útil cuando un usuario deja un comentario por primera vez sobre un vehículo o responde a otro comentario.

7. EliminarComentarios:

- Este método permite eliminar un comentario existente. Se recibe el idcomentario y se elimina la entrada correspondiente en la base de datos. Este método se utilizaría si un usuario decide eliminar un comentario que hizo anteriormente, o si un administrador decide eliminar un comentario inapropiado o que infrinja las normas.

RELACIONES Y FLUJO DE COMENTARIOS Y RESPUESTAS

Es importante destacar que los comentarios pueden tener respuestas, lo que crea una estructura jerárquica en los datos. Los usuarios pueden dejar comentarios sobre vehículos, y otros usuarios pueden responder a esos comentarios. Para gestionar esta relación, el campo idcomentariopadre es clave, ya que permite rastrear cuál comentario original tiene respuestas asociadas.

Por ejemplo:

- Si el idcomentario de un comentario es 100 y el idcomentariopadre de una respuesta es 100, entonces esa respuesta está directamente relacionada con el comentario original. Si el idcomentariopadre es NULL, significa que el comentario no tiene respuestas y es un comentario principal.

EJEMPLOS DE USO PRÁCTICO

- 1) Obtener un comentario por su id:
 - Un usuario o administrador desea ver un comentario específico. Usaría el método ObtenerComentariosPorId para obtener todos los detalles de ese comentario.
 - Ejemplo: Ver detalles de un comentario específico que un usuario dejó sobre un vehículo.
- 2) Obtener comentarios por vehículo:
 - Un usuario o administrador desea ver todos los comentarios de un vehículo específico. Usaría el método ObtenerComentariosPorIdVehiculo para obtener todos los comentarios asociados a ese vehículo.
 - Ejemplo: Mostrar todos los comentarios y respuestas sobre un vehículo en la página de detalles del vehículo.
- 3) Obtener respuestas a un comentario:

- Un usuario desea ver todas las respuestas a un comentario específico. Usaría el método `ObtenerComentariosPorIdRespuesta` pasando el `idcomentariopadre` del comentario original para obtener todas las respuestas a ese comentario.
 - Ejemplo: Ver todas las respuestas a un comentario específico en un hilo de discusión.
- 4) Actualizar un comentario:
- Un usuario desea corregir o modificar su propio comentario. Usaría el método `ModificarComentarios` pasando el `idcomentario` y el nuevo texto del comentario.
 - Ejemplo: Un usuario cambia su opinión sobre un vehículo y actualiza su comentario.
- 5) Eliminar un comentario:
- Un usuario o administrador desea eliminar un comentario. Usaría el método `EliminarComentarios` pasando el `idcomentario` correspondiente.
 - Ejemplo: Un administrador elimina un comentario inapropiado sobre un vehículo.

RESUMEN

La tabla `Comentarios` y sus métodos permiten gestionar los comentarios y respuestas de los usuarios sobre los vehículos, brindando una plataforma para la interacción entre usuarios a través de opiniones. Los métodos disponibles permiten obtener, actualizar, guardar y eliminar comentarios, así como gestionar las respuestas a esos comentarios, creando una estructura jerárquica que facilita la navegación entre comentarios y respuestas.

6.3.6-. EMPRESAS

La tabla `Empresas` almacena la información sobre las empresas que están asociadas a los vehículos, ya sea como proveedores de vehículos o como entidades que colaboran de alguna manera en el sistema. A continuación, se describe la estructura de esta tabla y los métodos asociados que permiten gestionar las empresas.

ESTRUCTURA DE LA TABLA

La tabla Empresas tiene los siguientes atributos:

1) idEmpresa:

- Este es el identificador único de cada empresa. Es la clave primaria de la tabla y se utiliza para identificar de manera única a cada empresa en el sistema.

2) nombre:

- El nombre de la empresa. Este campo es utilizado para almacenar el nombre con el que se identifica la empresa dentro del sistema.

3) descripción:

- Este campo contiene una breve descripción sobre la empresa, explicando a qué se dedica o su rol dentro del sistema. Proporciona más contexto sobre la empresa.

4) dirección:

- Este campo guarda la dirección física de la empresa. Es útil para saber dónde está localizada y puede ser importante para gestiones logísticas o de contacto.

5) teléfono:

- El número de teléfono de contacto de la empresa. Permite tener un canal de comunicación directo con la empresa.

6) correo electronico:

- La dirección de correo electrónico de la empresa. Es otro medio de contacto y puede ser utilizado para comunicaciones oficiales o soporte.

7) ofrece coches:

- Este campo indica si la empresa ofrece vehículos. Puede ser un valor booleano (1 o 0) que indica si la empresa tiene vehículos disponibles para el sistema o si solo colabora de alguna otra manera.

RELACIONES CON OTRAS TABLAS

La tabla Empresas tiene una relación con la tabla Vehículos. Las empresas pueden ofrecer vehículos, por lo que la información de las empresas estará vinculada a los vehículos disponibles en el sistema. Esta relación puede implicar que cada vehículo esté asociado con una empresa que lo ofrece.

MÉTODOS ASOCIADOS A LA TABLA

Los métodos proporcionados permiten gestionar la información de las empresas en la base de datos. A continuación, se detallan los métodos disponibles:

1. ObteneEmpresaPorId:

- Este método recibe el idEmpresa como parámetro y devuelve todos los datos de la empresa correspondiente a ese id. Este método es útil para obtener la información completa de una empresa específica cuando se conoce su id.

2. ObtenerEmpresas:

- Este método devuelve una lista de todas las empresas almacenadas en la base de datos. Se utilizaría en la aplicación de escritorio para mostrar todas las empresas disponibles en el sistema.

3. ObtenerEmpresaPorNombre:

- Este método recibe el nombre de una empresa como parámetro y devuelve los datos de la empresa que coincide con ese nombre. Este método es útil cuando se desea obtener información sobre una empresa a partir de su nombre en lugar de su id.

4. ModificarEmpresa:

- Este método permite actualizar la información de una empresa existente. Se pasa el idEmpresa y los nuevos datos (nombre, descripción, dirección, etc.) y actualiza la entrada correspondiente en la base de datos.

5. CrearEmpresa:

- Este método se utiliza para crear una nueva empresa en la base de datos. Se pasan los datos de la empresa (nombre, descripción, dirección, teléfono, correo electrónico, y si ofrece coches) y el método inserta una nueva fila en la tabla Empresas.

6. EliminarEmpresa:

- Este método permite eliminar una empresa de la base de datos. Se pasa el idEmpresa de la empresa que se desea eliminar y el método elimina la entrada correspondiente de la tabla Empresas.

EJEMPLOS DE USO PRÁCTICO

1- Obtener información de una empresa por su id:

- Si se desea obtener todos los detalles de una empresa específica, se puede utilizar el método ObteneEmpresaPorId pasando el idEmpresa de la empresa que se desea consultar.
- Ejemplo: Ver detalles de una empresa que ofrece vehículos.

2- Obtener una lista de todas las empresas:

- Si se necesita ver todas las empresas registradas en el sistema, se puede utilizar el método ObtenerEmpresas para obtener la lista completa de empresas.
- Ejemplo: Mostrar todas las empresas en la aplicación de escritorio para que el administrador pueda gestionarlas.

3- Buscar una empresa por su nombre:

- Si se desea encontrar una empresa en específico por su nombre, el método ObtenerEmpresaPorNombre permite hacerlo de forma rápida.
- Ejemplo: Buscar una empresa específica para ver si tiene vehículos disponibles para los usuarios.

4- Actualizar información de una empresa:

- Si una empresa cambia de dirección, número de teléfono, o cualquier otro dato relevante, se puede usar el método `ModificarEmpresa` para actualizar la información en la base de datos.
- Ejemplo: Actualizar la dirección de una empresa en el sistema.

5- Crear una nueva empresa:

- Si se desea agregar una nueva empresa que empieza a ofrecer vehículos, se utiliza el método `CrearEmpresa` para registrar a la nueva empresa en el sistema.
- Ejemplo: Un nuevo proveedor de vehículos se registra en el sistema.

6- Eliminar una empresa:

- Si una empresa deja de ofrecer vehículos o se desea eliminar su información del sistema, se utiliza el método `EliminarEmpresa` pasando el `idEmpresa` correspondiente.
- Ejemplo: Eliminar una empresa que ya no tiene vehículos disponibles.

RESUMEN

La tabla `Empresas` y sus métodos permiten gestionar las empresas asociadas a los vehículos dentro del sistema. Esto incluye la capacidad de agregar nuevas empresas, actualizar sus datos, obtener información de empresas por su `id` o nombre, y eliminar empresas cuando ya no son necesarias. Este sistema facilita la gestión de proveedores de vehículos y otras entidades que colaboran en el sistema.

6.3.7-. VEHÍCULOS

La tabla Vehículos almacena la información relevante sobre los vehículos que están disponibles en el sistema y están asociados a las empresas, usuarios y otros elementos del sistema. A continuación, se detalla la estructura de la tabla, las relaciones con otras tablas y los métodos que permiten gestionar estos vehículos.

ESTRUCTURA DE LA TABLA

La tabla Vehículos tiene los siguientes atributos:

1) idvehiculo:

- Este es el identificador único de cada vehículo en el sistema. Es la clave primaria de la tabla y se utiliza para identificar de manera única a cada vehículo en el sistema.

2) idempresa:

- Este campo contiene el id de la empresa a la que pertenece el vehículo. La relación entre las tablas Vehículos y Empresas indica que cada vehículo está asociado a una empresa que lo posee.

3) idusuariopropietario:

- Este campo almacena el id del usuario propietario del vehículo. Se relaciona con la tabla Usuarios, indicando que un usuario puede ser propietario de uno o más vehículos.

4) idestado:

- Este campo contiene el id del estado del vehículo (por ejemplo, disponible, alquilado, en mantenimiento, etc.). Este campo se relaciona con la tabla Estado_vehiculo, que define los posibles estados de un vehículo.

5) matricula:

- La matrícula del vehículo, que debe ser única y sirve para identificar el vehículo en el sistema.

6) marca:

- La marca del vehículo (por ejemplo, Toyota, Ford, etc.).

7) modelo:

- El modelo del vehículo (por ejemplo, Corolla, Focus, etc.).

8) año:

- El año de fabricación del vehículo.

9) km:

- La cantidad de kilómetros recorridos por el vehículo, útil para determinar el desgaste y la vida útil del mismo.

10) precioventa:

- El precio de venta del vehículo.

11) preciodia:

- El precio por día de alquiler del vehículo, en caso de que sea alquilado.

12) imagen:

- Una imagen que representa al vehículo. Puede ser una URL o una referencia a la ubicación de la imagen del vehículo.

RELACIONES CON OTRAS TABLAS

1) Empresas:

- Cada vehículo está asociado a una empresa que lo posee. La relación entre Vehículos y Empresas permite identificar cuál empresa es responsable de cada vehículo.

2) Usuarios:

- El idusuariopropietario relaciona el vehículo con el usuario que es su propietario. Esta relación puede ser importante para conocer qué usuarios tienen vehículos a su nombre.

3) Comentarios:

- Los vehículos pueden recibir comentarios de los usuarios. La relación con la tabla Comentarios permite registrar las opiniones y valoraciones sobre el vehículo.

4) Estado vehiculo:

- El campo idestado en la tabla Vehículos se relaciona con la tabla Estado_vehiculo, que define el estado en el que se encuentra cada vehículo (disponible, en mantenimiento, alquilado, etc.).

5) Contrato:

- Los vehículos están vinculados con los contratos de alquiler. Los contratos que incluyen un vehículo se relacionan con la tabla Contrato, que guarda información sobre el alquiler de los vehículos.

MÉTODOS ASOCIADOS A LA TABLA

Los métodos proporcionados permiten gestionar los vehículos dentro del sistema. A continuación, se detallan los métodos disponibles:

1) ObtenerVehiculos:

- Este método devuelve todos los vehículos almacenados en la base de datos. Se utilizaría en la aplicación de escritorio para mostrar una lista de todos los vehículos disponibles.

2) ObtenerVehiculosPorId:

- Este método recibe el idvehiculo como parámetro y devuelve todos los datos del vehículo correspondiente a ese id. Este método es útil cuando se desea obtener la información detallada de un vehículo específico.

3) ObtenerVehiculosPorMatricula:

- Este método recibe la matrícula de un vehículo como parámetro y devuelve los datos del vehículo que tiene esa matrícula. Es útil cuando se desea buscar un vehículo por su matrícula.

4) ModificarVehiculo:

- Este método permite actualizar la información de un vehículo existente. Se pasa el idvehiculo y los nuevos datos del vehículo (marca, modelo, año, matrícula, precio, etc.) para actualizar la entrada correspondiente en la base de datos.

5) CrearVehiculo:

- Este método se utiliza para crear un nuevo vehículo en la base de datos. Se pasan los datos del vehículo (empresa, propietario, estado, matrícula, marca, modelo, etc.) y el método inserta una nueva fila en la tabla Vehículos.

6) EliminarVehiculo:

- Este método permite eliminar un vehículo de la base de datos. Se pasa el idvehiculo del vehículo que se desea eliminar y el método elimina la entrada correspondiente de la tabla Vehículos.

EJEMPLOS DE USO PRÁCTICO

1) Obtener información de un vehículo por su id:

- Si se desea obtener todos los detalles de un vehículo específico, se puede utilizar el método ObtenerVehiculosPorId pasando el idvehiculo de dicho vehículo.
- Ejemplo: Ver los detalles de un vehículo en la aplicación de escritorio.

2) Obtener un vehículo por su matrícula:

- Si se conoce la matrícula de un vehículo, se puede utilizar el método ObtenerVehiculosPorMatricula para obtener los datos del vehículo.
- Ejemplo: Buscar un vehículo específico para verificar su disponibilidad o detalles.

3) Actualizar la información de un vehículo:

- Si un vehículo cambia de propietario, o si se actualizan sus datos, se puede utilizar el método UpdateVehiculo para modificar su información en la base de datos.
- Ejemplo: Actualizar el precio de venta o alquiler de un vehículo.

4) Crear un nuevo vehículo:

- Si se adquiere un nuevo vehículo para la empresa, se puede usar el método SaveVehiculo para registrar el vehículo en el sistema.
- Ejemplo: Añadir un nuevo vehículo a la flota de alquiler.

5) Eliminar un vehículo:

- Si un vehículo ya no se encuentra disponible o ha sido vendido, se puede utilizar el método DeleteVehiculo para eliminarlo del sistema.
- Ejemplo: Eliminar un vehículo de la base de datos cuando se retira de la venta o alquiler.

RESUMEN

La tabla Vehículos y sus métodos proporcionan las funcionalidades necesarias para gestionar la flota de vehículos en el sistema, permitiendo agregar, actualizar, obtener y eliminar vehículos. Estos métodos son clave para mantener el sistema de vehículos organizado y accesible, facilitando tareas como la visualización de vehículos, la actualización de su estado o datos, y la eliminación de vehículos que ya no están en uso. Las relaciones con otras tablas, como Empresas, Usuarios, Comentarios, Estado_vehiculo y Contrato, hacen que esta tabla sea central en la gestión de la flota de vehículos dentro del sistema.

6.3.8-. ESTADO DEL VEHÍCULO

La tabla Estado del Vehículo se encarga de gestionar los diferentes estados en los que pueden encontrarse los vehículos, lo cual es crucial para saber si un vehículo está disponible para alquilar, si está ocupado o si está fuera de servicio debido a un problema mecánico u otra razón. A continuación, se detalla la estructura de la tabla, las relaciones con otras tablas y los métodos disponibles para gestionar los estados de los vehículos.

ESTRUCTURA DE LA TABLA

La tabla Estado del Vehículo tiene los siguientes atributos:

1) idestado:

- Este es el identificador único del estado del vehículo. Es la clave primaria de la tabla y se utiliza para identificar de manera única cada estado en el sistema.

2) estado:

- Este campo contiene una descripción del estado del vehículo. Los posibles valores pueden incluir:
 - Disponible: El vehículo está listo para alquilar.
 - Alquilado: El vehículo está actualmente en alquiler.
 - En mantenimiento: El vehículo está fuera de servicio debido a reparaciones o mantenimiento.
 - Inactivo: El vehículo no está disponible por alguna otra razón.
 - Otros estados definidos por el sistema.

RELACIÓN CON OTRAS TABLAS

La tabla Estado del Vehículo está directamente relacionada con la tabla Vehículos a través del campo idestado. Cada vehículo tendrá asignado un estado que indica su disponibilidad o condición.

MÉTODOS ASOCIADOS A LA TABLA

Los métodos proporcionados permiten gestionar los estados de los vehículos en el sistema. A continuación, se detallan los métodos disponibles:

1) ObtenerEstadoVehiculos:

- Este método devuelve todos los estados de los vehículos que están almacenados en la base de datos. Se utilizaría en la aplicación de escritorio para mostrar una lista de todos los estados disponibles.

2) ObtenerEstadoVehiculosPorId:

- Este método recibe el idestado como parámetro y devuelve todos los datos del estado correspondiente a ese id. Es útil para obtener el estado de un vehículo en particular si se conoce su identificador.

3) ObtenerEstadoVehiculosPorEstado:

- Este método recibe el nombre del estado (por ejemplo, "Disponible", "Alquilado", etc.) y devuelve el idestado

correspondiente a ese estado. Es útil cuando se quiere buscar el estado por su descripción.

4) ModificarEstadoVehiculo:

- Este método permite actualizar el estado de un vehículo existente. Se pasa el idestado y los nuevos datos del estado (como “Disponible”, “En mantenimiento”, etc.) para actualizar la entrada correspondiente en la base de datos.

5) CrearEstadoVehiculo:

- Este método se utiliza para crear un nuevo estado para los vehículos en la base de datos. Se pasan los datos del estado (por ejemplo, “Alquilado”, “Disponible”) y el método inserta una nueva fila en la tabla EstadoVehiculo.

6) EliminarEstadoVehiculo:

- Este método permite eliminar un estado de vehículo de la base de datos. Se pasa el idestado del estado que se desea eliminar y el método elimina la entrada correspondiente de la tabla EstadoVehiculo. Este método es útil cuando se desean eliminar estados obsoletos o innecesarios.

EJEMPLOS DE USO PRÁCTICO

1) Obtener todos los estados de los vehículos:

- Si se desea mostrar una lista de todos los estados posibles para los vehículos, se puede utilizar el método ObtenerEstadoVehiculos en la aplicación de escritorio.
- Ejemplo: Mostrar en un desplegable los diferentes estados de los vehículos en la aplicación de administración.

2) Obtener el estado de un vehículo por su ID:

- Si se desea conocer el estado de un vehículo en particular, se puede utilizar el método `ObtenerEstadoVehiculosPorId` pasando el `idestado` del vehículo.
- Ejemplo: Consultar el estado de un vehículo específico al seleccionar su ID.

3) Obtener un estado por su nombre:

- Si se conoce el estado del vehículo (por ejemplo, "Alquilado") y se desea obtener su ID para asociarlo con un vehículo, se puede usar el método `ObtenerEstadoVehiculosPorEstado`.
- Ejemplo: Buscar el `idestado` para el estado "Disponible" y asignarlo a un vehículo.

4) Actualizar el estado de un vehículo:

- Si el estado de un vehículo cambia (por ejemplo, de "Disponible" a "Alquilado"), se puede utilizar el método `UpdateEstadoVehiculo` para actualizar la información en la base de datos.
- Ejemplo: Cambiar el estado de un vehículo a "En mantenimiento" cuando esté fuera de servicio.

5) Crear un nuevo estado:

- Si se necesita agregar un nuevo estado (por ejemplo, "En reparación"), se puede utilizar el método `SaveEstadoVehiculo` para añadir este estado a la tabla.
- Ejemplo: Crear un nuevo estado "En reparación" para gestionar vehículos que requieren arreglos.

6) Eliminar un estado:

- Si un estado ya no es necesario (por ejemplo, si se cambian las categorías de estado de los vehículos), se puede utilizar el método `EliminarEstadoVehiculo` para eliminar el estado de la base de datos.

- Ejemplo: Eliminar un estado “Inactivo” si ya no se utiliza en el sistema.

RESUMEN

La tabla Estado del Vehículo es fundamental para gestionar la disponibilidad y el estado de los vehículos dentro del sistema. Los métodos disponibles permiten realizar una amplia variedad de operaciones, desde consultar los estados existentes hasta actualizar, crear o eliminar estados para los vehículos. Esta gestión es clave para que los usuarios y administradores puedan ver la disponibilidad de los vehículos y tomar decisiones basadas en su condición (alquilado, disponible, en mantenimiento, etc.).

6.3.9-. TIPOS DE SEGURO

La tabla Seguros gestiona los diferentes tipos de seguros que los usuarios pueden elegir al contratar un servicio. Los seguros pueden variar en precio y cobertura, lo que afecta directamente al costo final del contrato. Esta tabla tiene los atributos y métodos necesarios para gestionar los seguros de manera eficiente en el sistema.

ESTRUCTURA DE LA TABLA

La tabla Seguros tiene los siguientes atributos:

1) idseguro:

- Este es el identificador único del seguro, es la clave primaria de la tabla y se utiliza para identificar de manera única cada seguro en el sistema.

2) nombre:

- El nombre del seguro, que podría ser algo como “Seguro Básico”, “Seguro Completo”, “Seguro a Todo Riesgo”, etc. Este campo describe el tipo de seguro.

3) descripción:

- Una breve descripción del seguro, que especifica las coberturas o características del seguro ofrecido, como lo que cubre, las exclusiones, etc.

4) coste:

- El coste asociado al seguro. Este valor puede variar dependiendo del tipo de seguro elegido y afecta el precio final del contrato de alquiler del vehículo.

RELACIÓN CON OTRAS TABLAS

La tabla Seguros está directamente relacionada con la tabla Contratos, ya que los usuarios seleccionan un seguro cuando firman un contrato para alquilar un vehículo. La relación permite saber qué seguro está asociado con cada contrato y permite calcular el costo total teniendo en cuenta el seguro elegido.

MÉTODOS ASOCIADOS A LA TABLA

Los métodos proporcionados permiten gestionar los seguros en el sistema, desde su obtención hasta su actualización y eliminación. A continuación, se detallan los métodos disponibles:

1) ObtenerSeguros:

- Este método devuelve todos los seguros almacenados en la base de datos. Se utilizaría en la aplicación de escritorio para mostrar una lista de todos los seguros disponibles para que el usuario elija.

2) ObtenerSegurosPorId:

- Este método recibe el idseguro como parámetro y devuelve todos los datos del seguro correspondiente a ese id. Es útil para obtener los detalles de un seguro específico, como su nombre, descripción y coste.

3) ObtenerSegurosPorNombre:

- Este método recibe el nombre del seguro y devuelve los datos del seguro correspondiente a ese nombre. Por ejemplo, si un usuario busca un “Seguro Completo”, este método devolverá los datos de ese seguro en particular.

4) ModificarSeguro:

- Este método permite actualizar un seguro existente. Se pasa el idseguro y los nuevos datos del seguro (como el nombre, descripción y coste) para actualizar la entrada correspondiente en la base de datos.

5) CrearSeguro:

- Este método se utiliza para crear un nuevo seguro en la base de datos. Se pasan los datos del seguro (nombre, descripción y coste) y el método inserta una nueva fila en la tabla Seguros.

6) EliminarSeguro:

- Este método permite eliminar un seguro de la base de datos. Se pasa el idseguro del seguro que se desea eliminar y el método elimina la entrada correspondiente de la tabla Seguros. Este método es útil cuando se desea eliminar un seguro que ya no se ofrece o que ha quedado obsoleto.

EJEMPLOS DE USO PRÁCTICO

1) Obtener todos los seguros disponibles:

- Si se desea mostrar una lista de todos los seguros disponibles para que un cliente elija, se puede utilizar el método ObtenerSeguros en la aplicación de escritorio.

- Ejemplo: Mostrar en un formulario de contratación todos los seguros disponibles con sus precios.

2) Obtener un seguro por su ID:

- Si se desea conocer los detalles de un seguro específico (por ejemplo, si un cliente desea saber más sobre un seguro por su ID), se puede utilizar el método `ObtenerSegurosPorId` pasando el `idseguro` del seguro.
- Ejemplo: Consultar los detalles de un seguro particular al seleccionar su ID.

3) Obtener un seguro por su nombre:

- Si se conoce el nombre del seguro y se desea obtener su ID y otros detalles, se puede utilizar el método `ObtenerSegurosPorNombre` pasando el nombre del seguro (por ejemplo, "Seguro Básico").
- Ejemplo: Buscar el `idseguro` para el seguro "Seguro Completo" y asignarlo al contrato del usuario.

4) Actualizar un seguro existente:

- Si el precio o la descripción de un seguro cambia (por ejemplo, si se modifica el coste de un "Seguro a Todo Riesgo"), se puede utilizar el método `UpdateSeguro` para actualizar los datos en la base de datos.
- Ejemplo: Cambiar el precio del seguro de "Seguro Básico" debido a un ajuste en los costos.

5) Crear un nuevo seguro:

- Si se introduce un nuevo tipo de seguro (por ejemplo, un nuevo seguro "Seguro de Pérdida Total"), se puede utilizar el método `SaveSeguro` para agregar este nuevo seguro a la base de datos.
- Ejemplo: Crear un nuevo seguro que cubra daños totales a vehículos.

6) Eliminar un seguro:

- Si un seguro ya no está disponible (por ejemplo, si se discontinúa el “Seguro Económico”), se puede utilizar el método DeleteSeguro para eliminar ese seguro de la base de datos.
- Ejemplo: Eliminar un seguro de la base de datos que ya no se ofrece a los clientes.

RESUMEN

La tabla Seguros es esencial para la gestión de los seguros disponibles al contratar un servicio de alquiler de vehículos. Los métodos asociados permiten realizar operaciones clave como obtener, crear, actualizar y eliminar seguros. La información gestionada en esta tabla afecta el costo de los contratos de alquiler, ya que cada cliente puede elegir un seguro con distintas coberturas, lo que influye en el precio final del servicio.

6.3.10-. CONTRATOS

La tabla Contratos guarda la información relacionada con los contratos de servicio que los clientes han firmado para alquilar un vehículo, incluyendo detalles sobre el vehículo alquilado, el cliente, el seguro seleccionado, el estado del contrato y los costos asociados.

ESTRUCTURA DE LA TABLA

La tabla Contratos contiene los siguientes atributos:

1) idcontrato:

- Este es el identificador único del contrato, es la clave primaria de la tabla y se utiliza para identificar de manera única cada contrato en el sistema.

2) idvehiculo:

- El idvehiculo es la referencia al vehículo que ha sido alquilado bajo el contrato. Se relaciona con la tabla Vehículos.

3) idcliente:

- El idcliente es el identificador del cliente que ha firmado el contrato. Se relaciona con la tabla Usuarios (clientes).

4) idestado:

- El idestado indica el estado actual del contrato (por ejemplo, activo, cancelado, finalizado). Se relaciona con la tabla EstadoContrato.

5) id_seguro:

- El id_seguro hace referencia al seguro elegido para el contrato. Se relaciona con la tabla Seguros.

6) fecha_inicio:

- La fecha_inicio indica el comienzo del contrato, es decir, el día en que el cliente comienza a alquilar el vehículo.

7) fecha_final:

- La fecha_final indica la fecha en que finaliza el contrato y, por lo tanto, el período de alquiler del vehículo.

8) precio_dia:

- El precio_dia es el costo diario de alquilar el vehículo, basado en el vehículo alquilado y el seguro seleccionado.

9) precio_total:

- El `precio_total` es el costo total del contrato, que se calcula multiplicando el `precio_día` por la duración del contrato (diferencia entre `fecha_inicio` y `fecha_final`) y sumando el costo del seguro.

10) pagado:

- El campo `pagado` indica si el contrato ha sido completamente pagado. Este campo podría ser un valor booleano (verdadero/falso) o un valor numérico (por ejemplo, 1 para pagado, 0 para no pagado).

RELACIÓN CON OTRAS TABLAS

La tabla Contratos está relacionada con las siguientes tablas:

- Vehículos: Cada contrato está vinculado a un vehículo específico que ha sido alquilado.
- Usuarios: Cada contrato está asociado a un cliente que firma el contrato.
- Seguros: El contrato está vinculado a un seguro específico que el cliente ha elegido para su alquiler.
- EstadoContrato: El contrato tiene un estado que puede indicar si está activo, finalizado, cancelado, etc.

MÉTODOS ASOCIADOS A LA TABLA

Los métodos proporcionados permiten gestionar los contratos de manera eficiente en el sistema. A continuación, se detallan los métodos disponibles:

1) ObtenerContratos:

- Este método devuelve todos los contratos almacenados en la base de datos. Se utilizaría en la aplicación de escritorio para mostrar una lista de contratos existentes.

- Ejemplo: Mostrar todos los contratos activos y sus detalles, como vehículos alquilados, fechas, precios, etc.

2) ObtenerContratosPorId:

- Este método recibe el idcontrato como parámetro y devuelve todos los detalles de ese contrato específico. Es útil para obtener los detalles completos de un contrato al identificarlo por su ID.
- Ejemplo: Ver los detalles de un contrato específico para revisar la información de un alquiler pasado.

3) ModificarContratos:

- Este método permite actualizar un contrato existente. Se pasa el idcontrato y los nuevos datos del contrato (como fechas, precios, estado, etc.) y se actualiza la entrada correspondiente en la base de datos.
- Ejemplo: Modificar las fechas de un contrato si el cliente solicita una extensión en el alquiler del vehículo.

4) CrearContratos:

- Este método se utiliza para crear un nuevo contrato. Se pasan los datos del contrato (vehículo, cliente, seguro, fechas, precios, etc.) y el método inserta una nueva fila en la tabla Contratos.
- Ejemplo: Crear un nuevo contrato cuando un cliente alquila un vehículo y selecciona un seguro.

5) EliminarContratos:

- Este método permite eliminar un contrato de la base de datos. Se pasa el idcontrato del contrato que se desea eliminar y el método elimina la entrada correspondiente de la tabla Contratos.
- Ejemplo: Eliminar un contrato si el cliente cancela su alquiler o si el contrato es inválido.

EJEMPLOS DE USO PRÁCTICO

1) Obtener todos los contratos existentes:

- Si un administrador desea ver todos los contratos firmados, se puede utilizar el método `ObtenerContratos` para obtener una lista de todos los contratos.
- Ejemplo: Mostrar en la interfaz de administración todos los contratos vigentes, cancelados o finalizados.

2) Obtener un contrato por su ID:

- Si un cliente o administrador desea revisar los detalles de un contrato específico, se puede usar el método `ObtenerContratosPorId` pasando el `idcontrato`.
- Ejemplo: Ver los detalles de un contrato de alquiler por ID, como las fechas de inicio y finalización, el precio total y el estado.

3) Actualizar un contrato:

- Si se requiere modificar algún dato de un contrato (por ejemplo, extender la fecha de finalización del alquiler o actualizar el estado del contrato), se puede utilizar el método `UpdateContratos`.
- Ejemplo: Cambiar la fecha final de un contrato si el cliente decide quedarse más tiempo con el vehículo.

4) Crear un nuevo contrato:

- Si un cliente decide alquilar un vehículo, se puede utilizar el método `CrearContratos` para crear un nuevo contrato con la información proporcionada.
- Ejemplo: Registrar un nuevo contrato cuando un cliente alquila un coche y elige un seguro.

5) Eliminar un contrato:

- Si un contrato debe eliminarse (por ejemplo, si un cliente cancela su alquiler antes de que comience), se puede utilizar el método EliminarContratos.
- Ejemplo: Eliminar un contrato si el cliente decide no seguir adelante con el alquiler.

RESUMEN

La tabla Contratos es fundamental para gestionar las relaciones entre los clientes, los vehículos alquilados y los seguros seleccionados. Los métodos asociados permiten obtener, crear, actualizar y eliminar contratos de forma eficiente. La información almacenada en esta tabla es crucial para el seguimiento y la gestión de las operaciones de alquiler de vehículos, incluyendo el cálculo de precios y el estado de los contratos.

6.3.11-. ESTADO CONTRATO

La tabla EstadoContrato guarda la información relacionada con el estado de los contratos, indicando si un contrato está en estado “inicializado”, “finalizado”, “cancelado” u otros posibles estados relevantes. Este estado se usa para gestionar el ciclo de vida de los contratos en el sistema.

ESTRUCTURA DE LA TABLA

La tabla EstadoContrato contiene los siguientes atributos:

1) idestado:

- Este es el identificador único del estado del contrato, es la clave primaria de la tabla y se utiliza para identificar de manera única cada estado del contrato.

2) estado:

- El estado describe el estado actual del contrato. Los valores comunes podrían ser “Inicializado”, “Activo”, “Finalizado”, “Cancelado”, o cualquier otro estado relevante según las necesidades del sistema.

RELACIÓN CON OTRAS TABLAS

La tabla EstadoContrato está relacionada con la tabla Contratos, ya que cada contrato tiene un estado específico. El estado del contrato determina su ciclo de vida, si está activo, finalizado, pendiente o en cualquier otro estado definido.

MÉTODOS ASOCIADOS A LA TABLA

Los métodos proporcionados permiten gestionar los estados de los contratos de manera eficiente en el sistema. A continuación, se detallan los métodos disponibles para interactuar con la tabla EstadoContrato:

1) ObtenerEstadoContratos:

- Este método devuelve todos los estados de contrato almacenados en la base de datos. Es útil para visualizar todos los posibles estados de los contratos en la aplicación de escritorio.
- Ejemplo: Mostrar una lista de todos los estados de contrato, como “Activo”, “Finalizado”, “Cancelado”, etc.

2) ObtenerEstadoContratosPorId:

- Este método recibe el idestado como parámetro y devuelve el estado del contrato correspondiente. Es útil para obtener el estado de un contrato específico utilizando su identificador de estado.
- Ejemplo: Ver el estado de un contrato específico al buscarlo por su idestado.

3) ObtenerEstadoContratosPorEstado:

- Este método recibe el estado (por ejemplo, “Finalizado” o “Activo”) y devuelve todos los contratos que se encuentran en ese estado. Esto permite obtener todos los contratos de un estado específico.
- Ejemplo: Ver todos los contratos que están en estado “Finalizado”.

4) ModificarEstadoContrato:

- Este método permite actualizar el estado de un contrato existente. Se pasa el idestado y el nuevo valor del estado y el método actualizará el estado del contrato correspondiente en la base de datos.
- Ejemplo: Cambiar el estado de un contrato de “Activo” a “Finalizado” cuando el cliente devuelva el vehículo.

5) CrearEstadoContrato:

- Este método se utiliza para crear un nuevo estado para los contratos. Se pasan los datos del estado (como el nombre del estado) y el método inserta una nueva fila en la tabla EstadoContrato.
- Ejemplo: Crear un nuevo estado de contrato, como “Pendiente de pago”, si se define un estado adicional.

6) EliminarEstadoContratos:

- Este método permite eliminar un estado de contrato de la base de datos. Se pasa el idestado del estado que se desea eliminar y el método elimina la entrada correspondiente de la tabla EstadoContrato.
- Ejemplo: Eliminar un estado no utilizado, como “En espera”, si ya no es relevante.

EJEMPLOS DE USO PRÁCTICO

1) Obtener todos los estados de contrato:

- Si un administrador desea ver todos los posibles estados que pueden tener los contratos, se puede utilizar el método `ObtenerEstadoContratos`.
- Ejemplo: Mostrar una lista de todos los estados posibles, como "Activo", "Finalizado", "Cancelado".

2) Obtener un estado de contrato por ID:

- Si se desea obtener el estado de un contrato específico, se puede usar el método `ObtenerEstadoContratosPorId` pasando el `idestado` del estado.
- Ejemplo: Ver el estado de un contrato de alquiler por `idestado`, como "Activo".

3) Obtener todos los contratos en un estado específico:

- Si se desea obtener todos los contratos que están en un estado específico, se puede usar el método `ObtenerEstadoContratosPorEstado`.
- Ejemplo: Ver todos los contratos que han finalizado.

4) Actualizar el estado de un contrato:

- Si un contrato cambia de estado (por ejemplo, de "Activo" a "Finalizado"), se puede usar el método `UpdateEstadoContrato`.
- Ejemplo: Cambiar el estado de un contrato a "Finalizado" cuando el cliente devuelve el vehículo.

5) Crear un nuevo estado para los contratos:

- Si es necesario agregar un nuevo estado de contrato, como “En espera”, se puede usar el método `SaveEstadoContrato`.
- Ejemplo: Crear un nuevo estado de contrato como “Pendiente de validación”.

6) Eliminar un estado de contrato:

- Si un estado ya no es necesario o no se usa más, se puede usar el método `DeleteEstadoContratos`.
- Ejemplo: Eliminar un estado de contrato no utilizado como “En espera”.

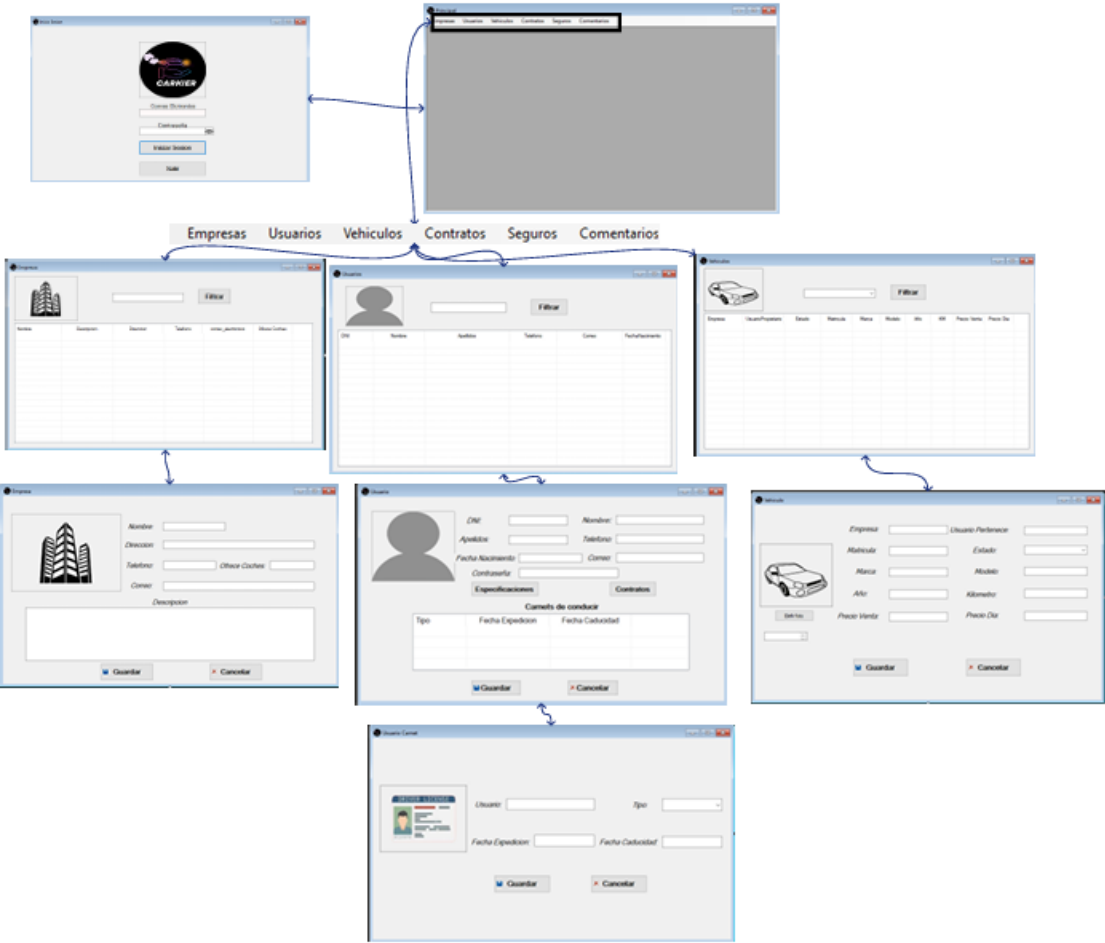
RESUMEN

La tabla `EstadoContrato` es esencial para gestionar los diferentes estados de los contratos en el sistema, lo que permite hacer un seguimiento adecuado de cada contrato durante su ciclo de vida. Los métodos asociados permiten interactuar con los estados de los contratos, incluyendo la posibilidad de obtener, crear, actualizar y eliminar estados. Esta tabla juega un papel clave en la organización y seguimiento de los contratos en la aplicación.

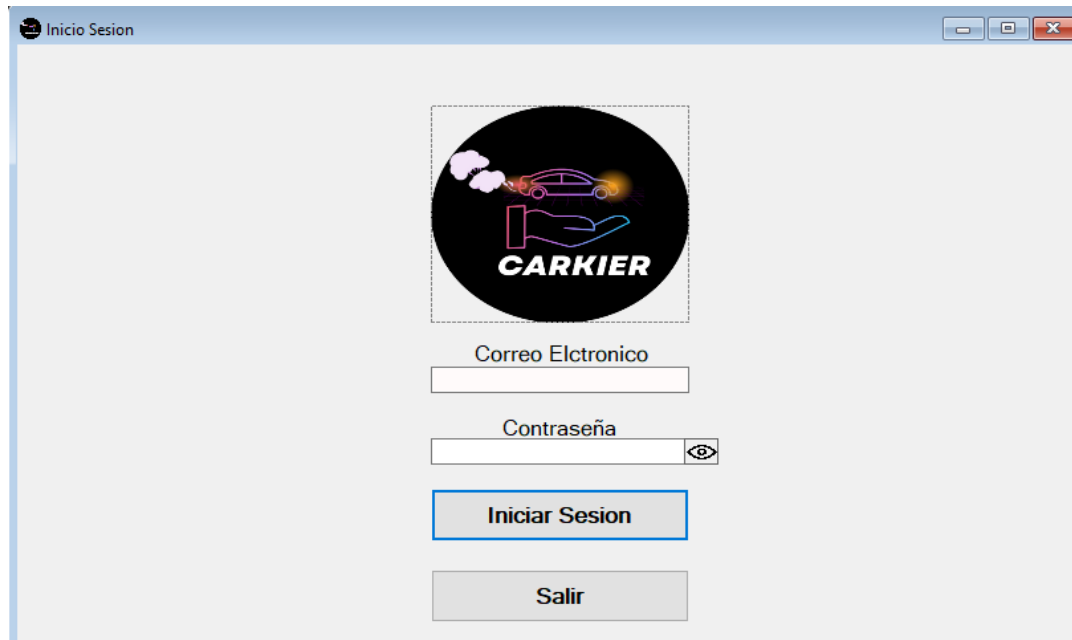
6.3-. Diagrama de Interfaces

En este apartado hablaremos de cada aplicación (Escritorio y Móvil) y mostrar un esquema resumido de cada aplicación.

ESQUEMA APLICACIÓN ESCRITORIO



APLICACIÓN DE ESCRITORIO

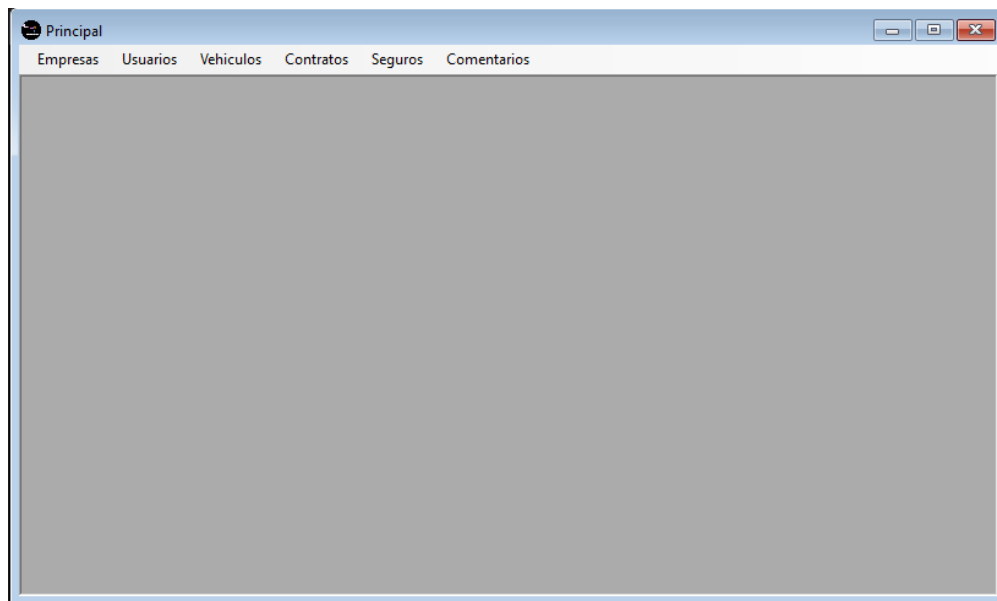


En esta ventana podremos hacer el inicio sesión, para ello deberíamos introducir el correo y contraseña y darle al inicio sesión en contraseña hay un ojo para que puedas verificar si la contraseña que has puesto está bien o está mal escrita.

¿QUÉ PASARÍA SI NO METES BIEN LOS DATOS?

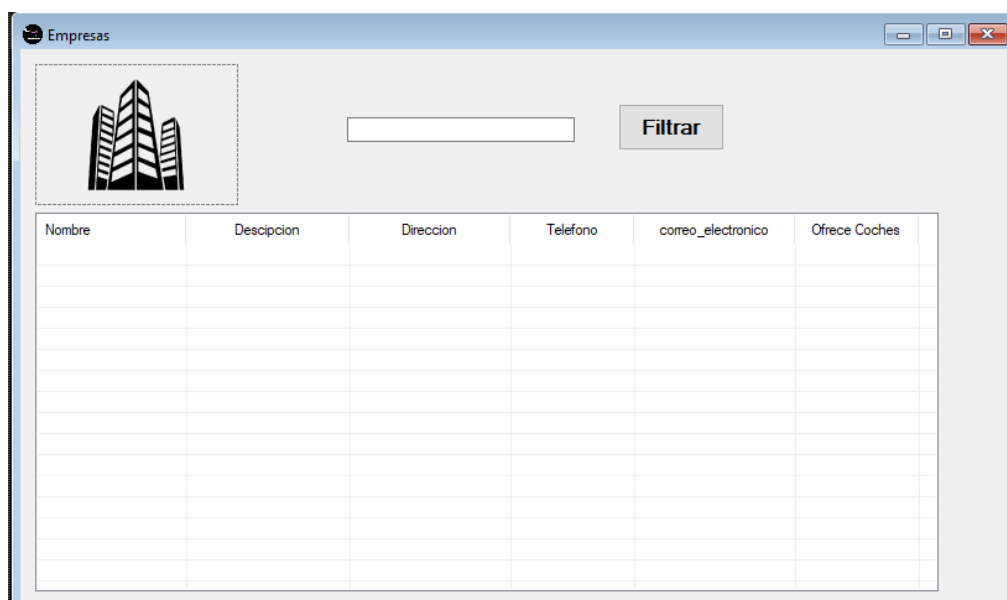
Lo que pasaría es que saldría un mensaje de error diciéndote que no es posible iniciar sesión ya que no existe ningún administrador con esas credenciales.

Si las credenciales que introdujiste y es administrador te llevaría a la siguiente ventana.



La segunda ventana que sería la principal es un mdi-container, en la parte de arriba hay unos textos donde le tendrás que ir dando y saldrán una ventana para cada una donde se mostraría dichas tablas y su información.

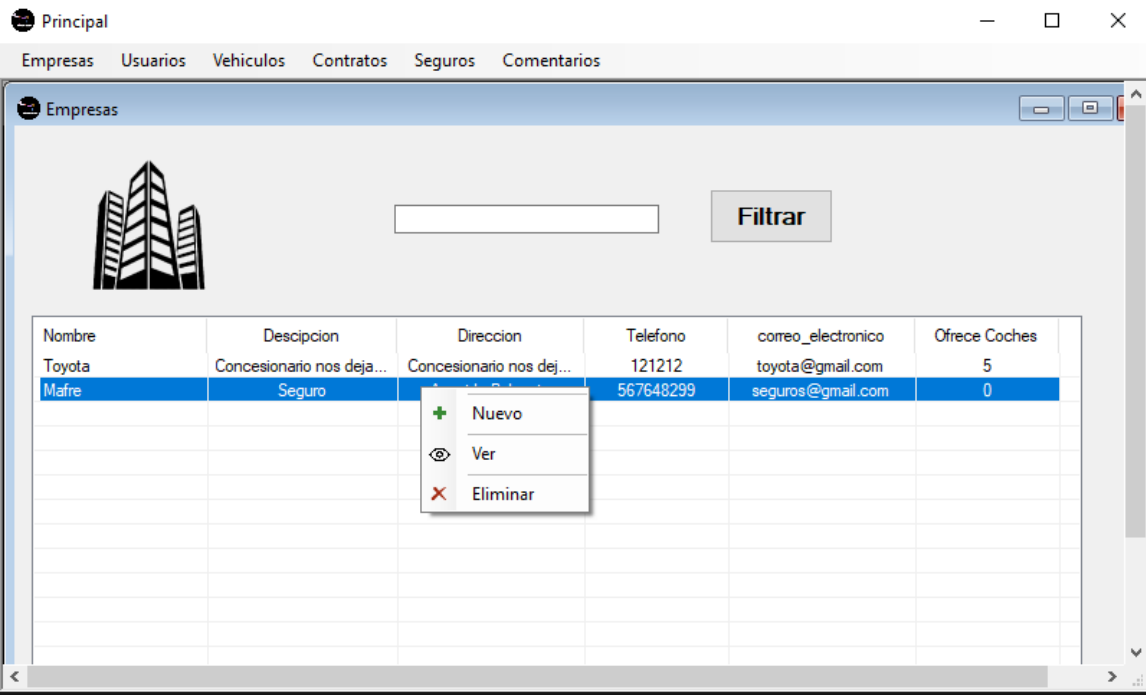
EMPRESAS



Que queremos ver las empresas que tenemos le daría a empresas y nos saldría en la parte grisácea que se ve otra ventana que sería esta.

En la tabla se vería cada empresa que hay y el filtrar este puesto para un futuro no muy lejano para mostrar la empresa con más facilidad.

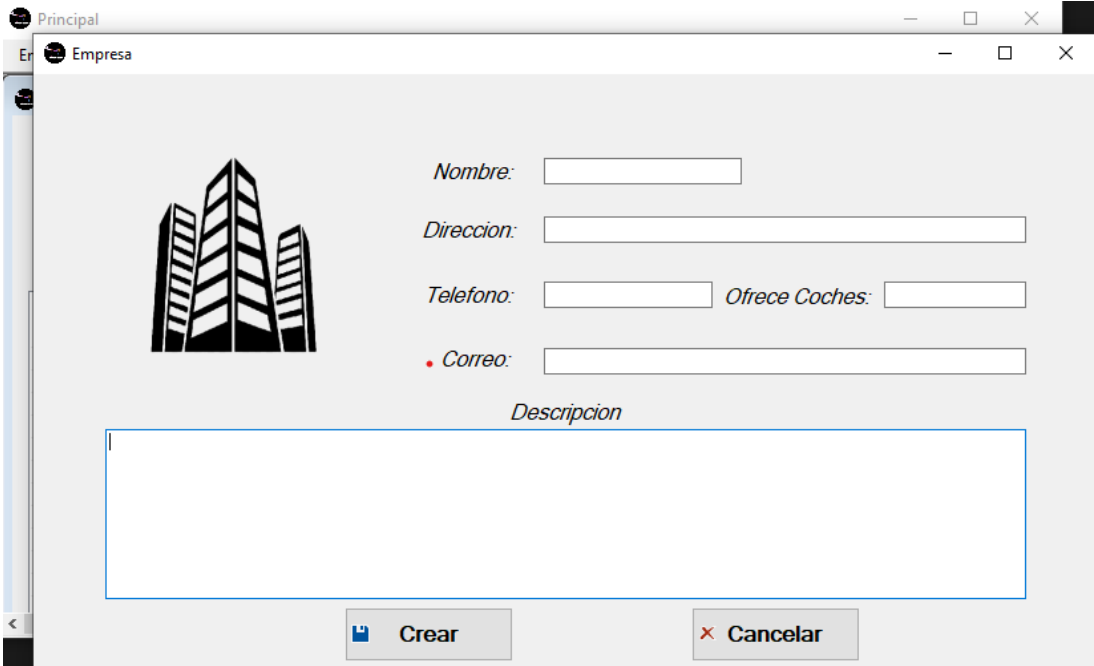
Crear, ver/modificar y eliminar



Para ello estando en la ventana de empresa como es en este caso tendremos que elegir ósea dar clic en un elemento en la tabla y como podrás ver se pondrá en azul y le daremos click izquierdo y saldrá un mini menú donde saldrá las 3 opciones que serían las siguientes: Nuevo, ver que también sirve para modificar y eliminar.

NUEVA EMPRESA

Para crear la empresa deberías dar a la opción nueva en la imagen anterior.



Principal Empresa

Nombre:

Direccion:

Telefono: Ofrece Coches:

Correo:

Descripcion

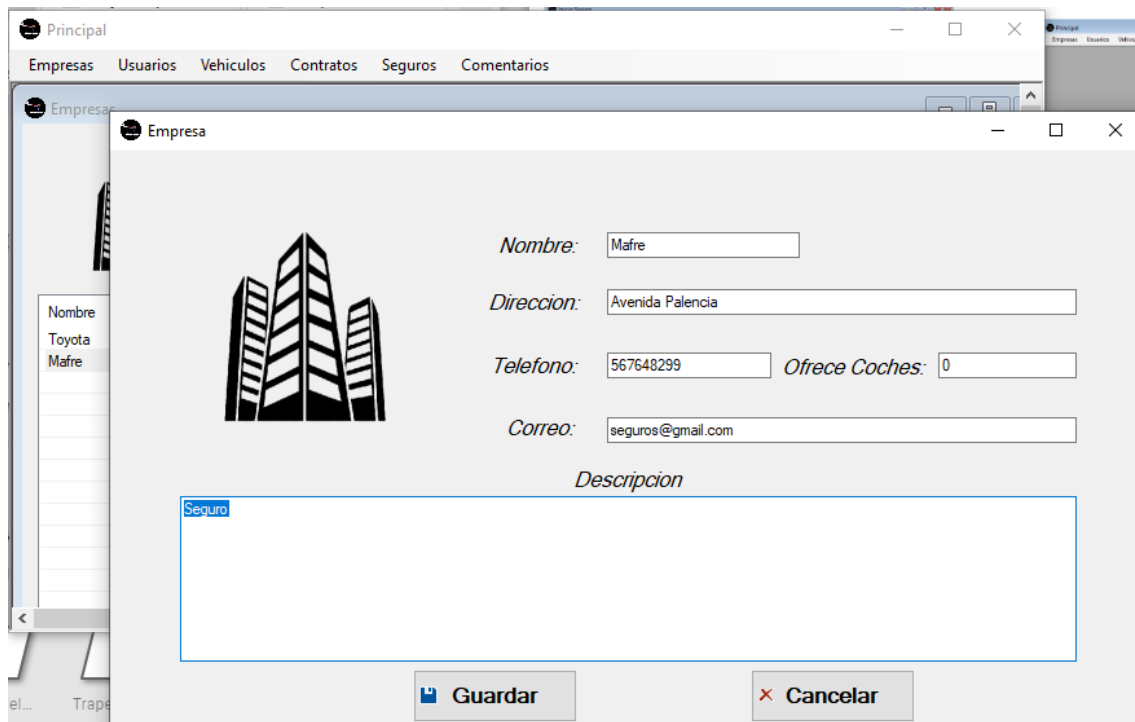
Crear Cancelar Si

En esta ventana tendremos que introducir los siguientes atributos para la hora de la creación de la empresa si por ejemplo no ofrece coches los dejamos en blanco y no pasaría nada.

Una vez escrito todos los atributos de la empresa le das a crear y se crearía y si hay algún fallo o error te daría un mensaje. Si por ejemplo te has confundido y no querías entrar en esta ventana tienes 2 opciones para salir dar a la x o dar al botón de cancelar.

MODIFICAR O VER

Para poder mostrar una empresa o modificarlo en la imagen anterior en vez de dar a nuevo le puedes dar dos veces o dándole a la opción de mostrar.



The screenshot displays a web application interface. At the top, there is a navigation bar with the following tabs: Empresas, Usuarios, Vehiculos, Contratos, Seguros, and Comentarios. Below this, a modal window titled 'Empresa' is open. On the left side of the modal, there is a sidebar with a list of companies: 'Nombre', 'Toyota', and 'Mafre'. The main area of the modal contains a form for editing a company. The form includes a logo of three stylized buildings on the left. To the right of the logo, there are several input fields: 'Nombre:' with the value 'Mafre', 'Direccion:' with the value 'Avenida Palencia', 'Telefono:' with the value '567648299', and 'Ofrece Coches:' with the value '0'. Below these fields is a 'Correo:' field with the value 'seguros@gmail.com'. At the bottom of the form is a large text area labeled 'Descripcion' with the value 'Seguro'. At the bottom right of the modal, there are two buttons: 'Guardar' (Save) and 'Cancelar' (Cancel).

Si quieres actualizar algún campo solo tendrás que modificar el campo que quieras cambiar y darle a guardar y te saldrá una ventana de si estás seguro que quieras guardar los cambios y le das a guardar.

Confirmar Guardado



¿Quieres guardar los datos de las empresas?

Sí

No

En el momento que le des a guardar se guardara en la base de datos automáticamente.

ELIMINAR

Si en vez de esos quieres eliminar le das a eliminar en vez de ver y te saldrá un mensaje que te saldrá un mensaje de confirmación.

Confirmación de eliminación



¿Estás seguro de que quieres eliminar la empresas seleccionado?

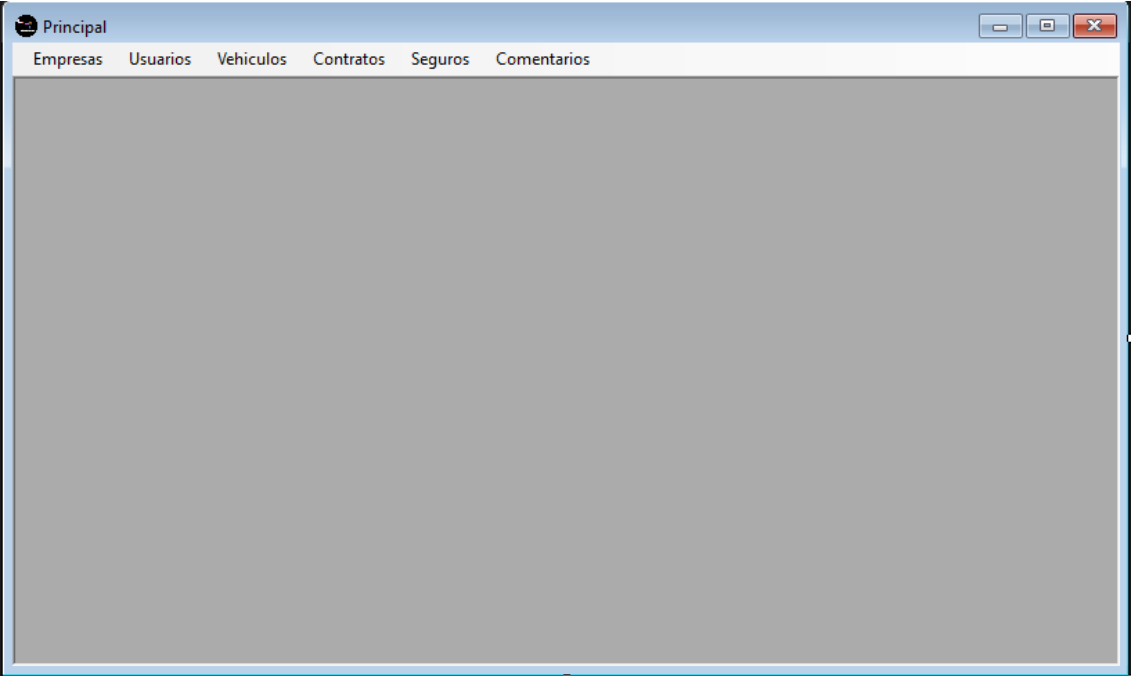
Sí

No

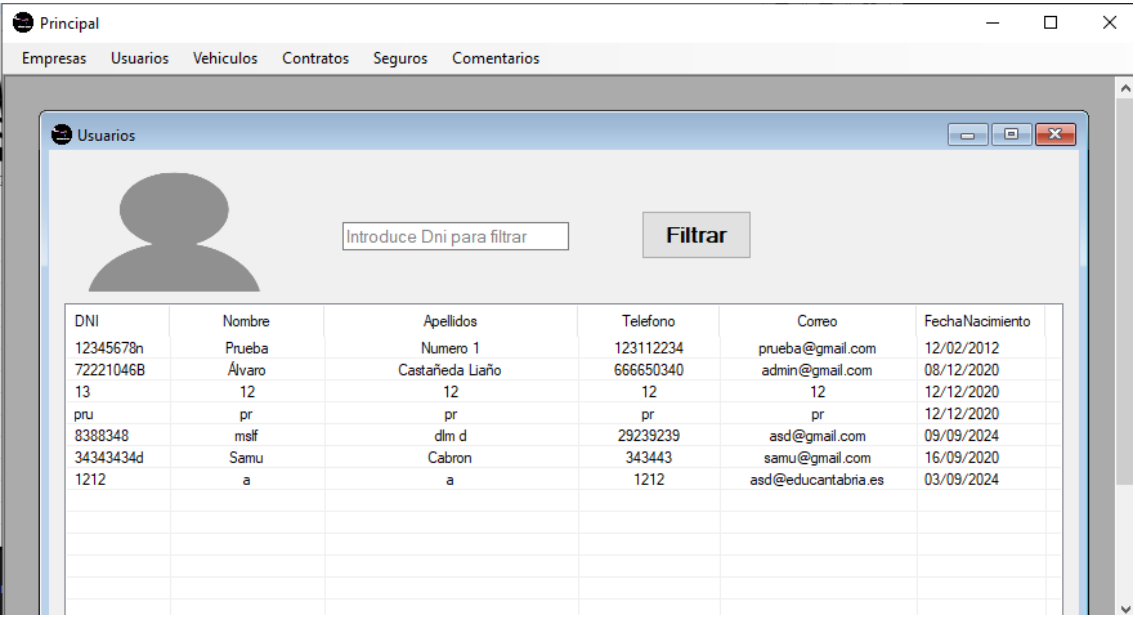
En el momento que le des que si dicha empresa que has marcado se va a eliminar automáticamente.

USUARIOS

VER

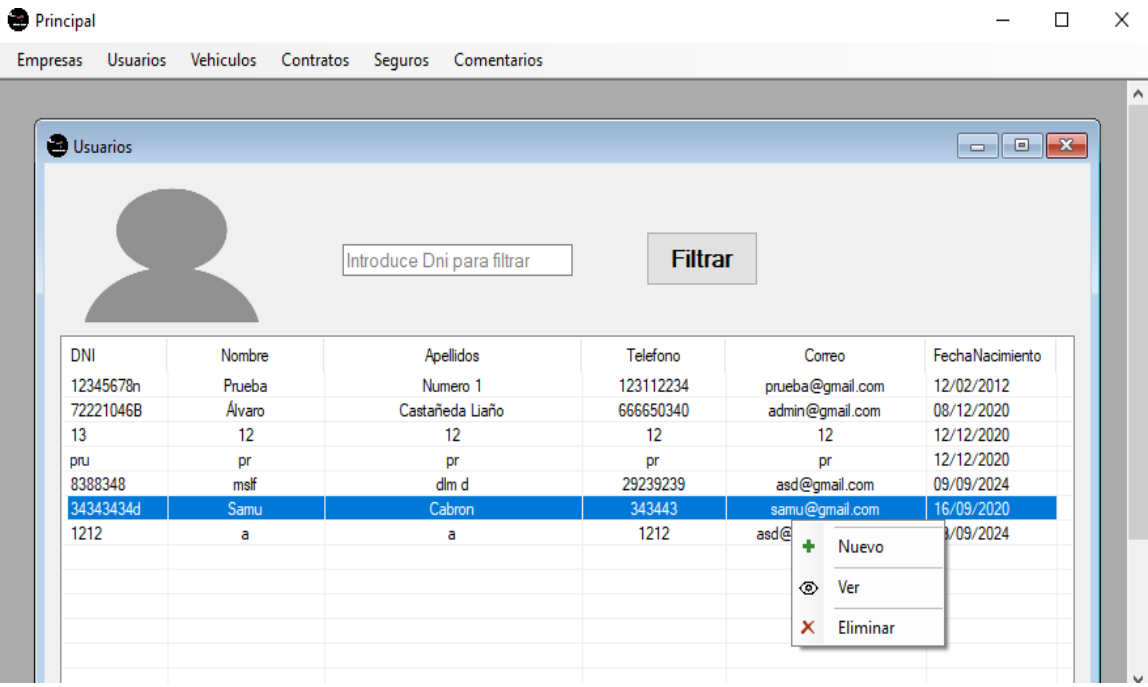


Como hago para ver los usuarios los datos de cada uno volvamos a la venta principal y nos sale usuarios le daremos ahí y se abrirá una nueva ventana.



Como vemos en la imagen hay un filtrar que actualmente no funciona, pero en un futuro se implementara la funcionalidad, y saldría una tabla con las columnas de los atributos de cada usuario.

CREAR, VER O MOSTRAR Y ELIMINAR



Para que nos salga las opciones estas es fácil es marcar un usuario en la tabla y clic derecho y nos muestra las opciones.

CREAR

Para crear un nuevo usuario le damos a nuevo y nos abrirá la siguiente ventana:

Principal

Usuario

DNI: *Nombre:*

Apellidos: *Telefono:*

Fecha Nacimiento: *Correo:*

Contraseña:

Especificaciones **Contratos**

Carnets de conducir

Tipo	Fecha Expedicion	Fecha Caducidad	

Crear **Cancelar**

Aquí escribirías los campos desde DNI hasta la contraseña y le das a crear los botones de especificaciones y contratos más la tabla de carnets de momento no te haría falta agregar nada ya que solo quieres crear un usuario. Una vez que agregues los atributos le das a crear te saldrá un mensaje de estas seguro.

VER O MODIFICAR

Si quieres ver o modificar un usuario en específico en vez de darle a nuevo le das a ver.

Principal

Empresas Usuarios Vehiculos Contratos Seguros Comentarios

Usuario

DNI: 13 Nombre: 12

Apellidos: 12 Telefono: 12

Fecha Nacimiento: 12-12-2020 Correo: 12

Especificaciones Contratos

Carnets de conducir

Tipo	Fecha Expedicion	Fecha Caducidad	

Guardar Cancelar

Si quieres modificar algún dato clicas el campo le modificas y le das a guardar te saldrá un mensaje de si estás seguro que quieres guardar los cambios y le das que si como harías para crear un carnet pues en la tabla que se muestra le darás Click derecho y nuevo y si el usuario tiene algún carnet y necesitas modificar o eliminar clic derecho ver o eliminar como en los demás.

ELIMINAR

Si quieres eliminar un usuario le daremos clic y le damos a eliminar nos saldrá un mensaje que te mostrara que en unos minutos se eliminara ya que este eliminar se marca la casilla de eliminar en la tabla datos del usuario y eliminado se ponga true y cuando pase 1 mes se eliminan los usuarios que estén a true en ese campo.

DATOS DEL USUARIO

En estos datos son más específicos los que hay se crea solo cuando se registra el usuario.

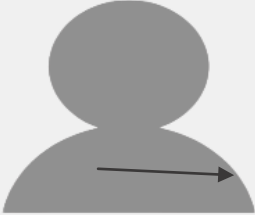
VER O MODIFICAR

Como hago para ver los datos del usuario de un usuario para ello tendremos que ver específicamente los datos de un usuario es decir ir a la ventana de ver un usuario y dar al botón de especificaciones

Principal

EmpresasUsuariosVehiculosContratosSegurosComentarios

Usuario



DNI:13

Nombre:12

Apellidos:12

Telefono:12

Fecha Nacimiento:12-12-2020

Correo:12

Especificaciones

Contratos

Carnets de conducir

Tipo	Fecha Expedicion	Fecha Caducidad	

Guardar

Cancelar

Principal

Empresas Usuarios Vehiculos Contratos Seguros Comentarios

Usuario

Datos Usuario

Usuario: Álvaro Castañeda Liaño

Administrador: SI Puntos: 4000

BANEOS

Fecha Inico: Sin banear Fecha Final: Sin banear

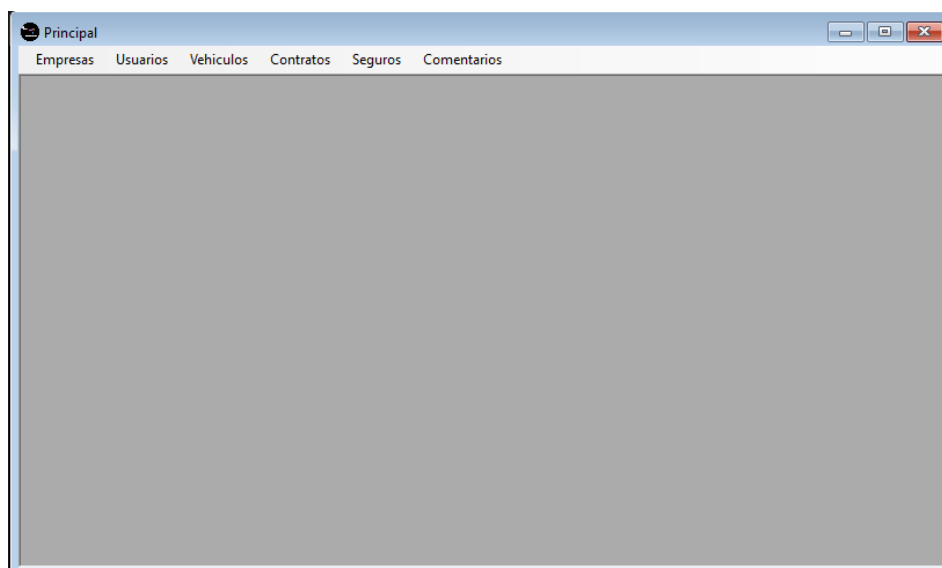
Cantidad baneado: 0 Eliminar: ☐

Guardar Cancelar

Una vez dada nos abriría esta ventana y en esta ventana podríamos ver y modificar los datos para salir si has modificado le das a guardar y te saldrá un mensaje de seguro que quieres guardar los cambios le das a que si y se guardara.

VEHICULOS

Para poder mostrar los vehículos hay que ir a la ventana principal y en las opciones de arriba dar a Vehículos



CREAR

[illegible]



Empresa:

Usuario Pertenece DNI:

Matricula:

Estado:

DISPONIBLE

Marca:


Modelo:

Año:

Kilometro:

Precio Venta:


Precio Dia:

 Crear

 Cancelar

En esta ventana rellenas los campos y le das a crear y si haya algún error de que te falta campos te saldrá un mensaje diciéndote que hay algo mal puesto.

VER O MODIFICAR



Empresa:

No pertenece

Usuario Pertenece:

Prueba Numero 1

Matricula:

280 DFH

Estado:

MANTENIMIENTO

Marca:

pesado

Modelo:

pequeño

Año:

2026

Kilometro:

87

Precio Venta:

12312

Precio Dia:

187

Guardar


Cancelar

Si lo que quieres es ver o modificar en vez de elegir nuevo le das a ver y saldría la ventana con los datos especificados. Si quieres modificar algún dato modificas el campo y le das a guardar te saldrá un mensaje de confirmación para proteger en caso de que te has confundido.

ELIMINAR

Si lo que quieres es eliminar el vehículo en vez de dar a nuevo o ver le das a eliminar. Te saldrá un mensaje de confirmación y le das a si.

Confirmación de eliminación

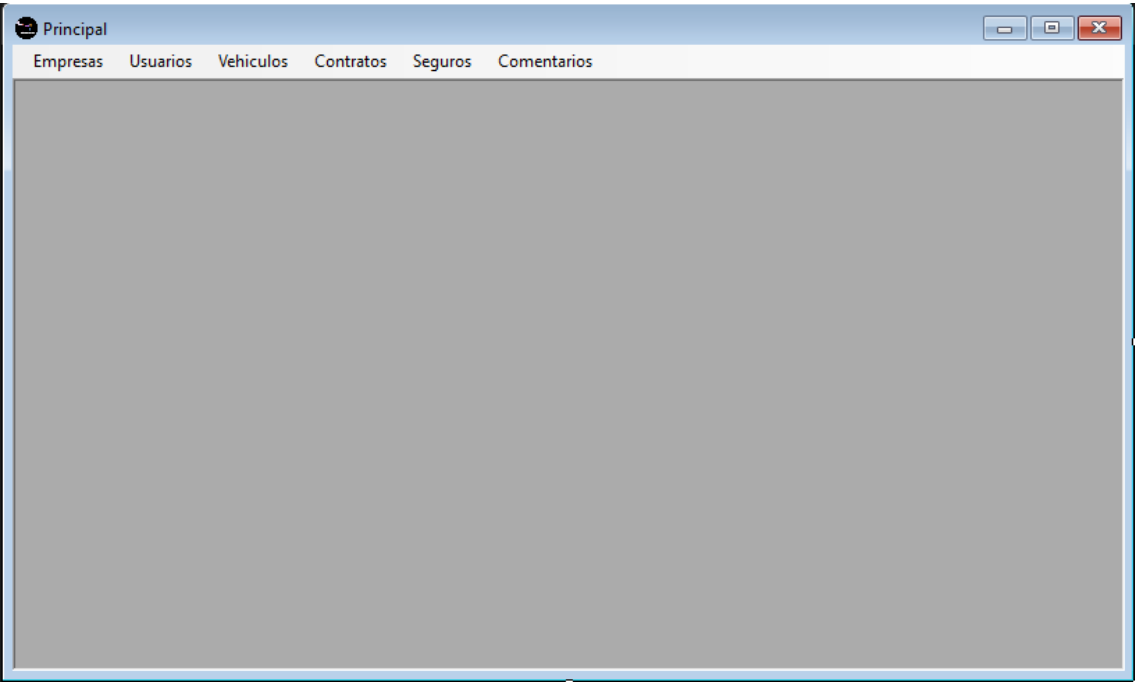


¿Estás seguro de que quieres eliminar el vehiculo seleccionado?

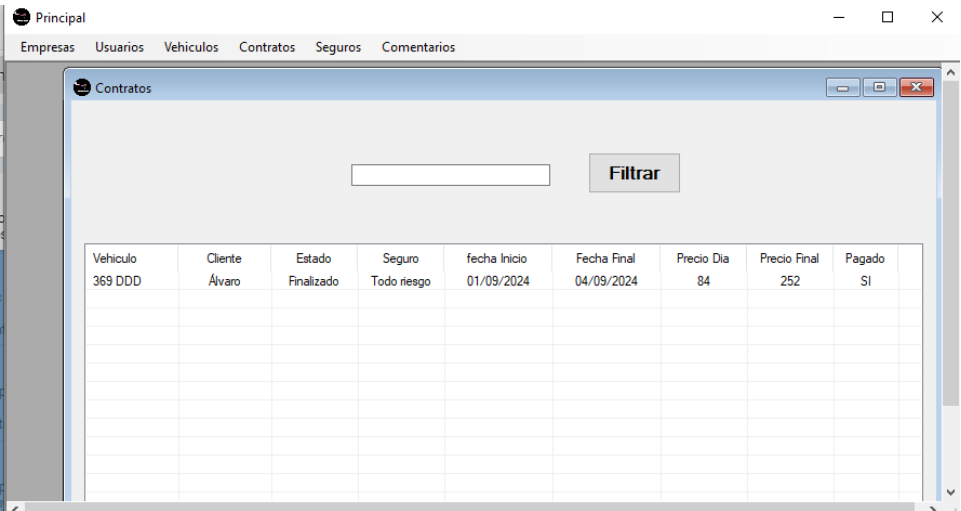
Sí

No

CONTRATOS



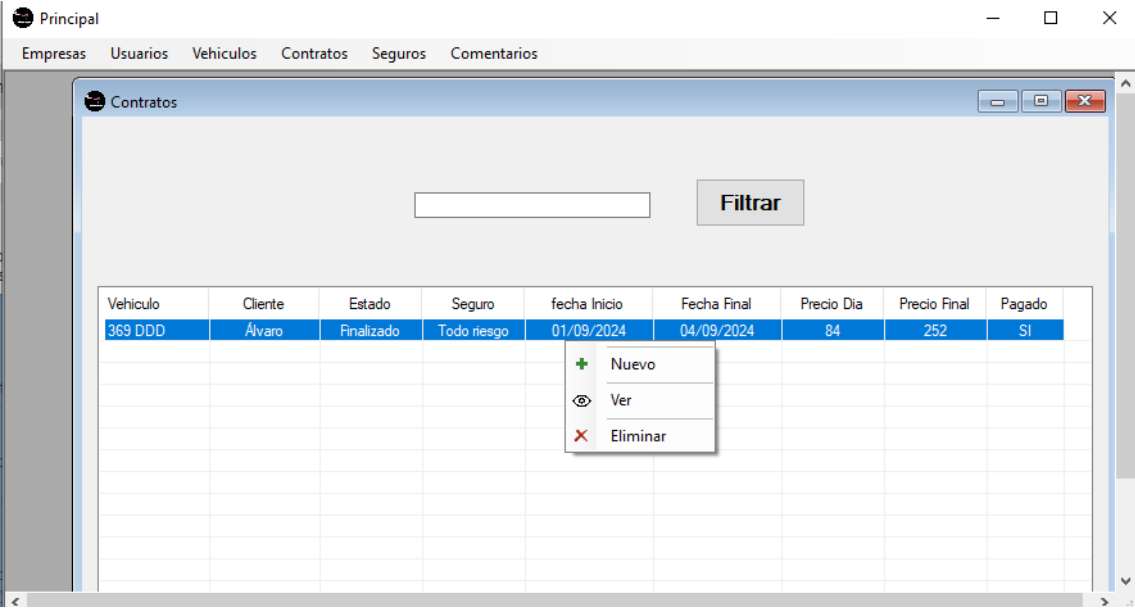
Y le damos a contratos y se mostrara una ventana muy parecida a la que hemos estado viendo seria la siguiente ventana:



En esta ventana se mostrará todos los contratos que ha habido con las especificaciones.

CREAR, VER O MODIFICAR Y ELIMINAR

Para poder conseguir mostrar las 3 opciones estas solo tendrás que elegir y dar clic derecho



CREAR

The screenshot shows a form titled 'Contrato' for creating a new contract. The form contains the following fields and controls:

- Matricula Vehiculo:** A text input field.
- Cliente:** A text input field.
- Estado:** A dropdown menu with 'Iniciado' selected.
- Seguro:** A dropdown menu with 'Todo riesgo' selected.
- Pagado:** A dropdown menu.
- Fecha Inicio:** A text input field.
- Fecha Final:** A text input field.
- Buttons:** 'Crear' (with a plus icon) and 'Cancelar' (with a red X icon).

Al elegir la opción de nuevo se abrirá la ventana, para crear dicho contrato tendrás que rellenar los campos y darle a crear

VER O MODIFICAR

Hay dos formas para ello si le das doble clic al contrato que quieres ver automáticamente se abrirá la ventana con los datos específicos sino le das clic derecho y ver.

Contrato

Matricula Vehiculo:

369 DDD

Cliente:

Alvaro Castañeda Liaño

Estado:

Iniciado

Seguro:

Todo riesgo

Pagado:

Si

Fecha Inicio:

01/09/2024

Fecha Final:

04/09/2024

Precio Dia:

84

Precio Total:

252

Guardar

Cancelar

Aquí como podrás ver podemos ver los campos rellenos de sus atributos y podemos modificar casi todo menos la matricula y el cliente y si queremos guardar le das a guardar

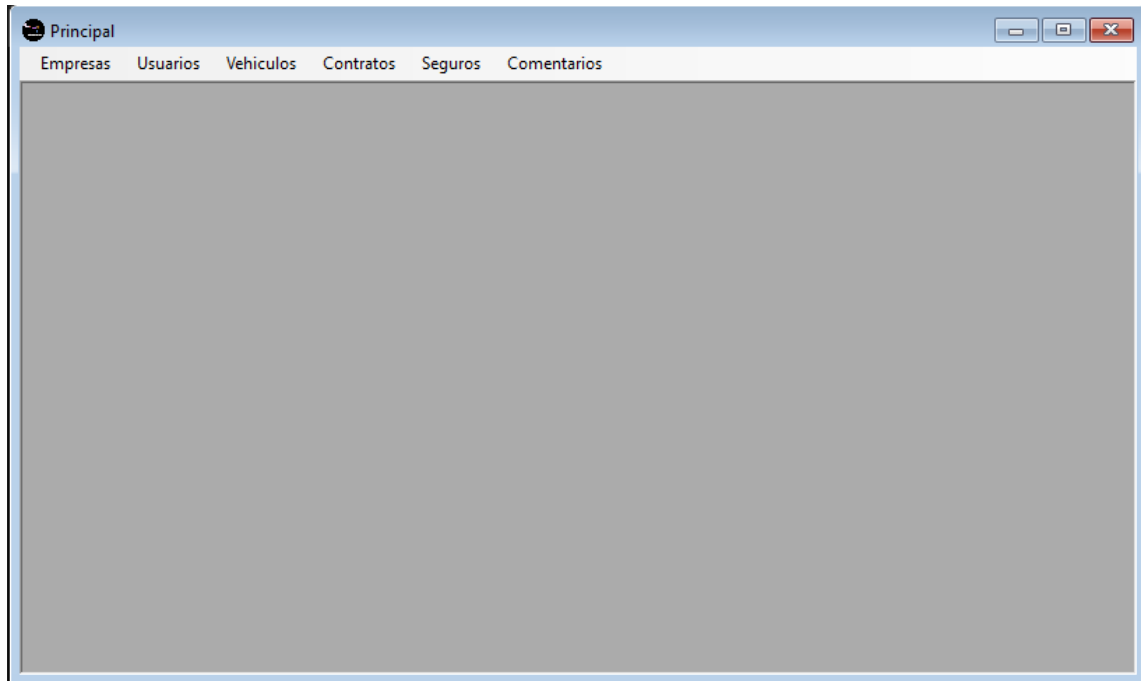
ELIMINAR

[illegible]

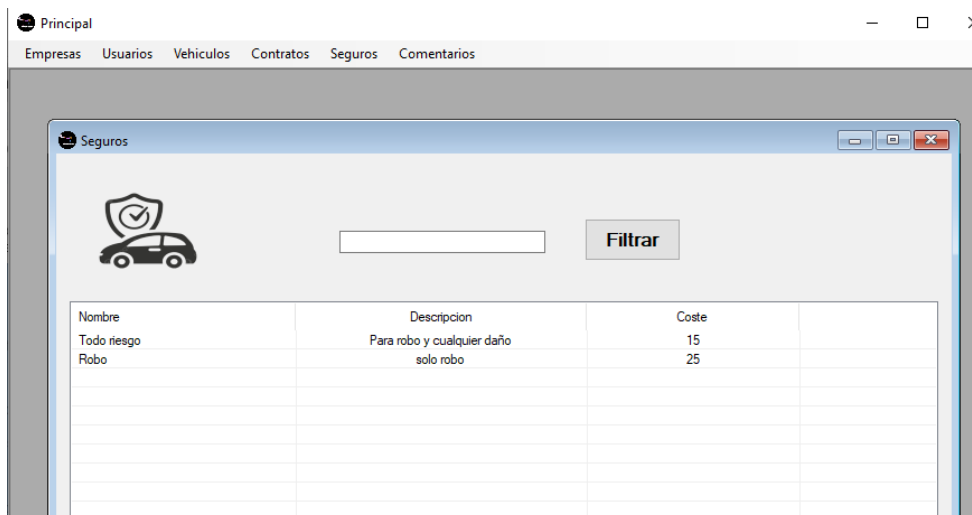
Al seleccionar un contrato y darle clic derecho le daremos a la opción de eliminar nos saldrá un mensaje de confirmación y le daremos a que si.

SEGUROS

Para ello volvemos a la ventana Principal

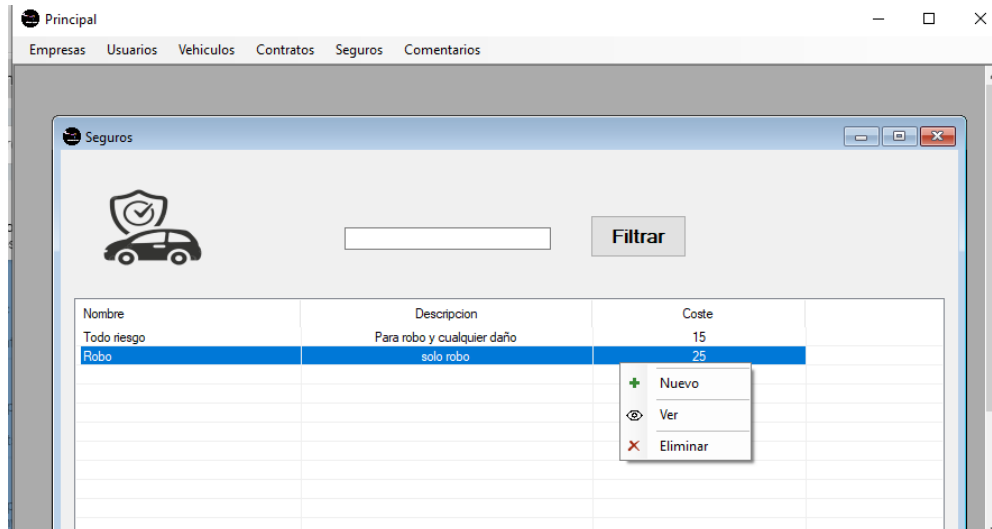


Le daremos a seguros y se abrirá una ventana con los tipos de seguros que tenemos



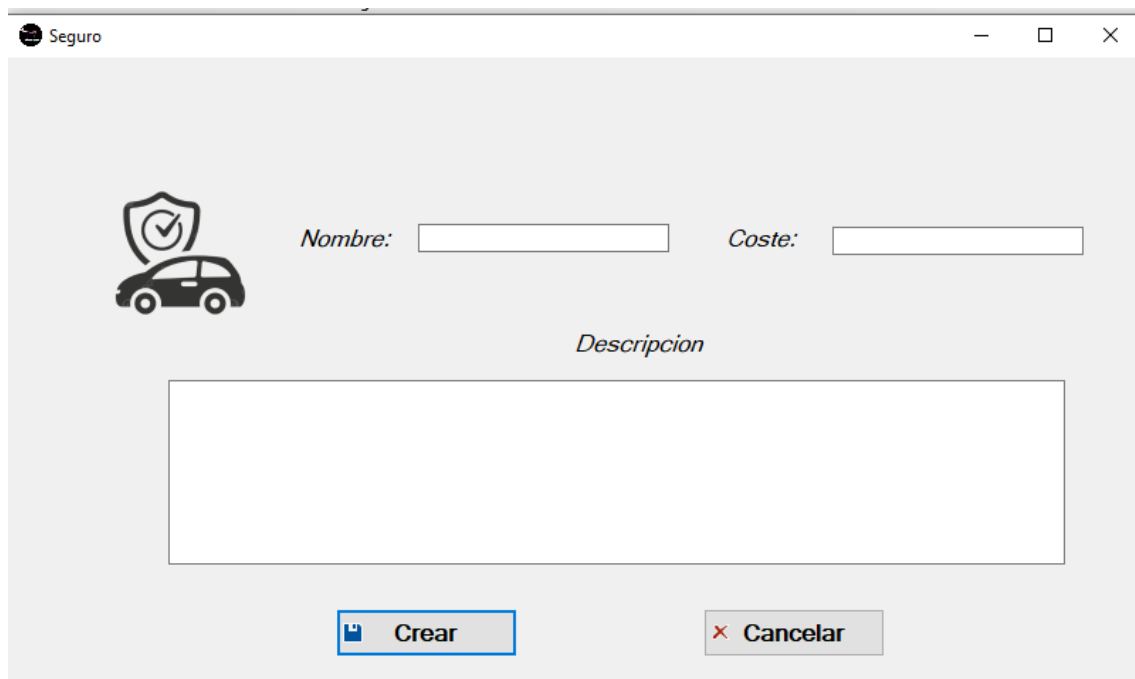
CREAR, VER O MODIFICAR Y ELIMINAR

Para que nos salga estas opciones tendremos que dar clic derecho y es cuando se nos abrirá un pequeño menú con las 3 opciones



CREAR


Para crear un nuevo seguro le daremos a nuevo esto abrirá una nueva ventana donde tendrás que introducir los datos



Esta es la ventana que se abrirá nada más darle a nuevo aquí introducimos esos 3 atributos que son nombre descripción y coste y le damos a crear si todo va bien no nos saldrá ningún mensaje de error.

VER O MODIFICAR

Lo mismo que para crear, pero en vez de dar a nuevo le daremos a ver y mostrara una ventana como esta



Seguro

Nombre: Robo Coste: 25

Descripcion

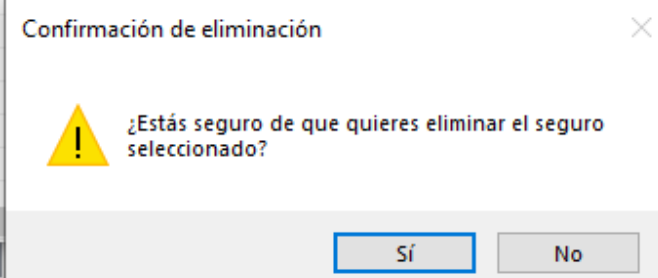
solo robo

Guardar Cancelar

Como podemos ver si le damos a ver los campos se verían rellenos y si quieres modificar algo solo tendrías que dar clic en el campo y escribir lo que quieras y una vez escrito darle a guardar y automáticamente se te guardaría.

ELIMINAR

Si quieres eliminar el seguro tendrás que dar clic derecho obviamente seleccionando un seguro y en el mini menú darle a eliminar



Confirmación de eliminación

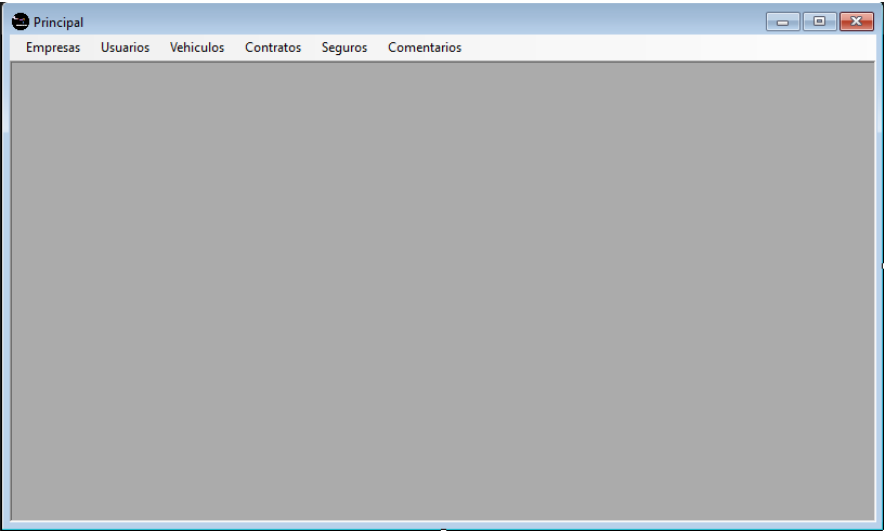
¿Estás seguro de que quieres eliminar el seguro seleccionado?

Sí No

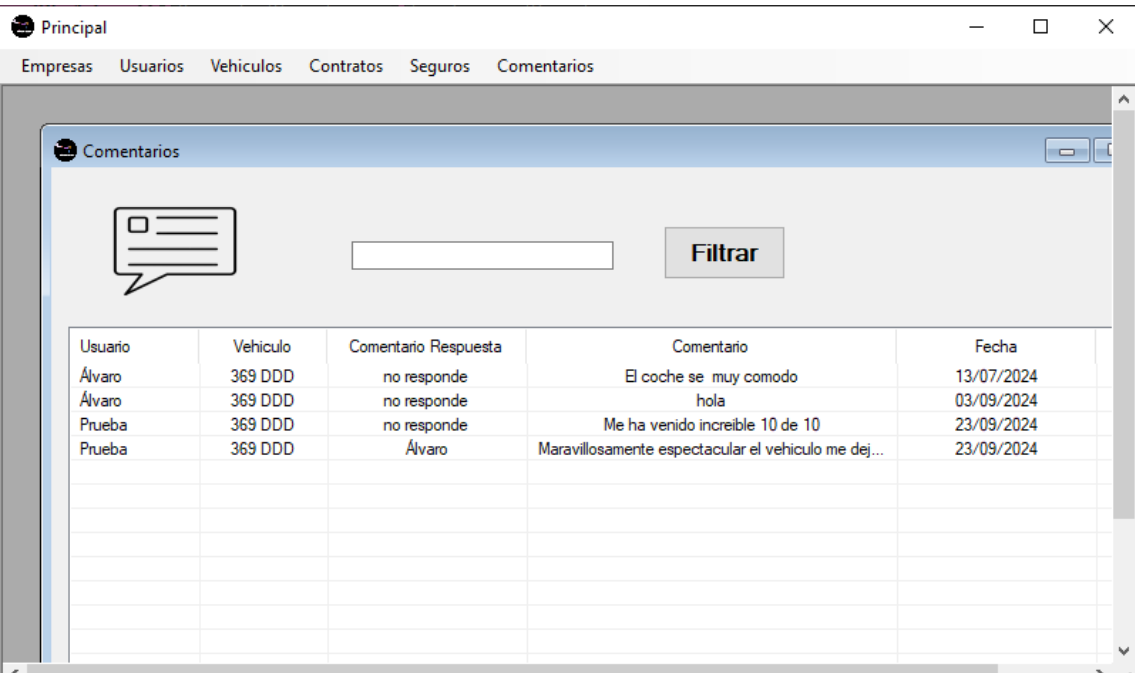
Te saldrá un mensaje de confirmación y le darás que si para eliminar

COMENTARIOS

Para ello volvemos a la ventana Principal y le daremos a Comentarios una vez dado se abrirá una ventana

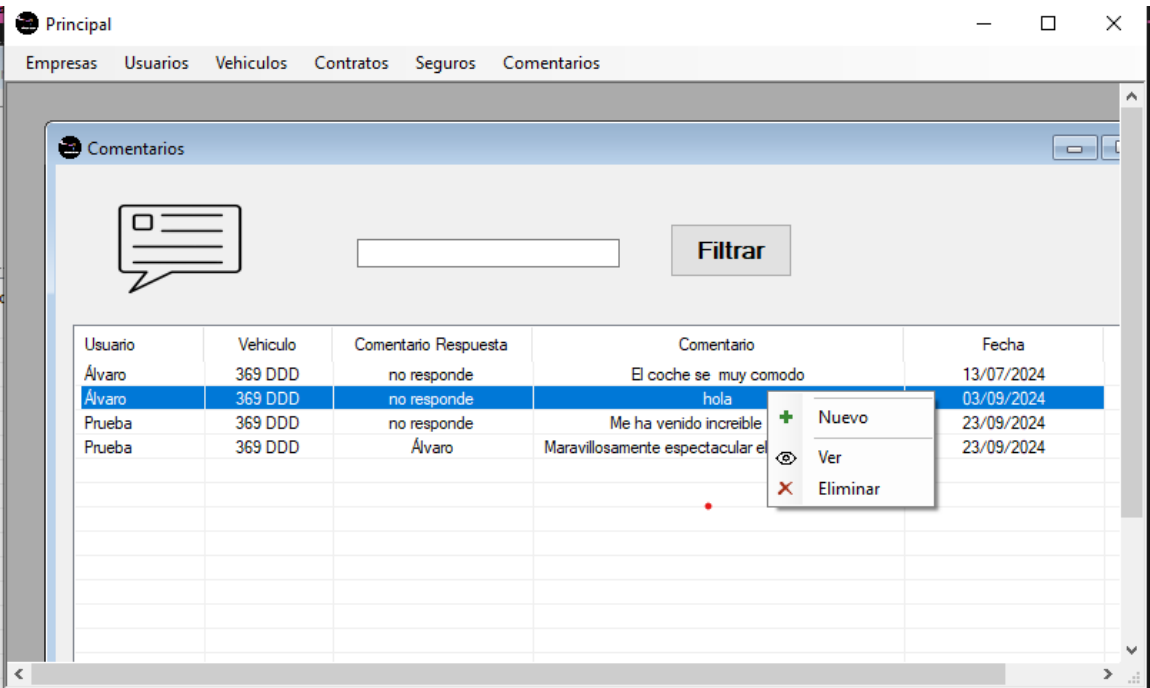


Aquí se mostrará todos los comentarios que hay guardadas en la base de datos, también hay un filtrar para un futuro actualmente no funciona.



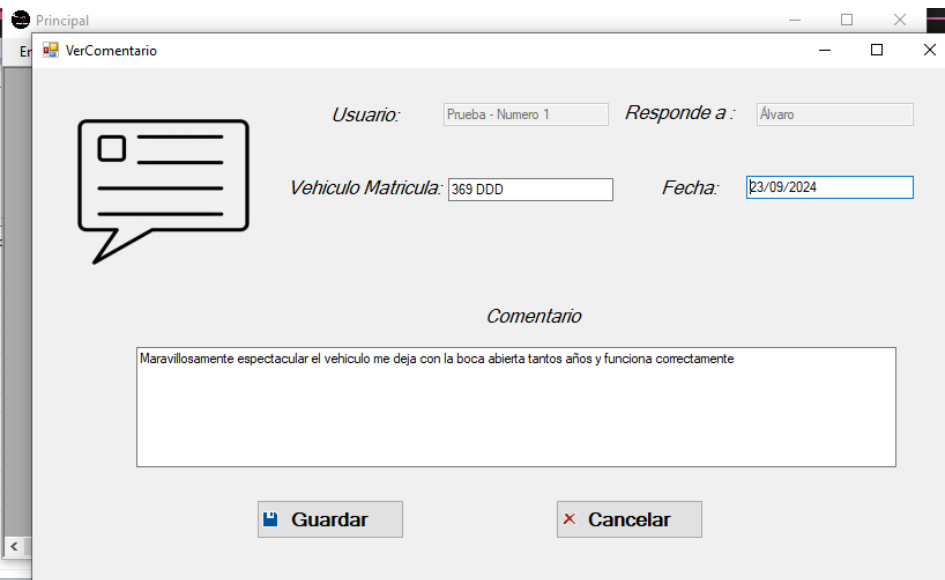
CREAR, VER O MODIFICAR Y ELIMINAR

Para que nos salga las opciones tendremos que darle clic derecho



CREAR

En las 3 opciones le daremos a Nuevo




Quando le demos se abrirá una nueva ventana donde saldrá los datos del comentario y si quieres modificar le modificas y le guardas sin problemas

ELIMINAR

Si lo que deseas es eliminar le das a eliminar aceptas la confirmación que saldrá y listo. La confirmación que saldría seria esta

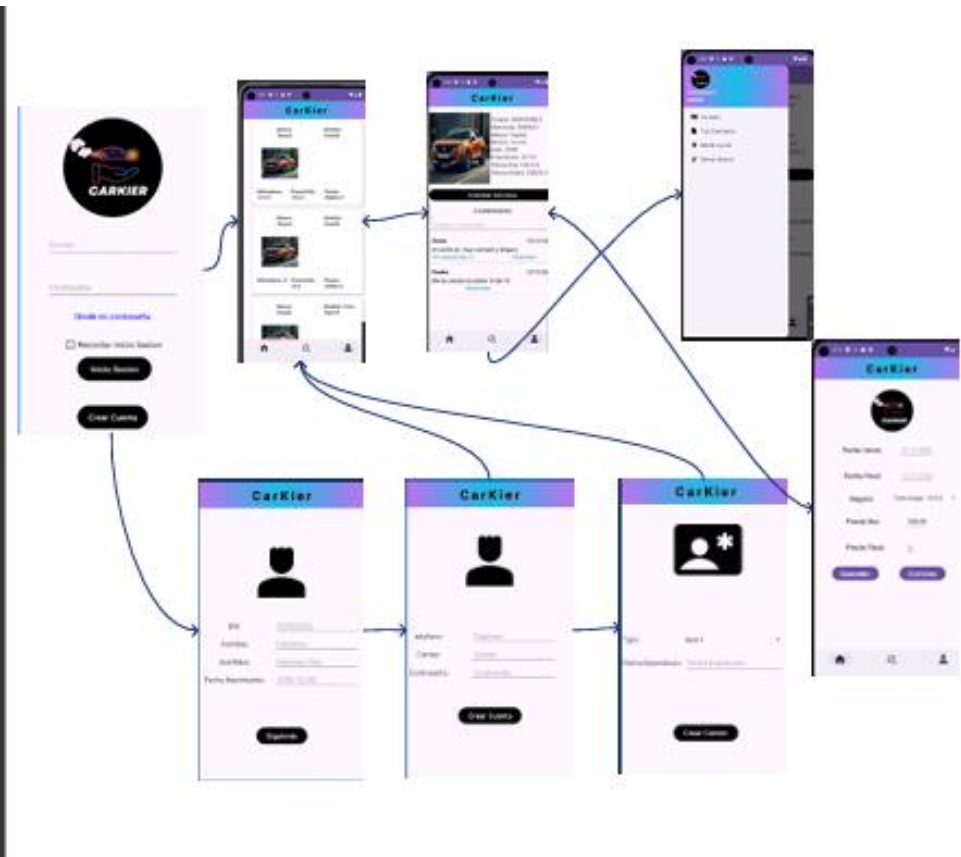
Confirmación de eliminación

 ¿Estás seguro de que quieres eliminar el comentario seleccionado?

Sí

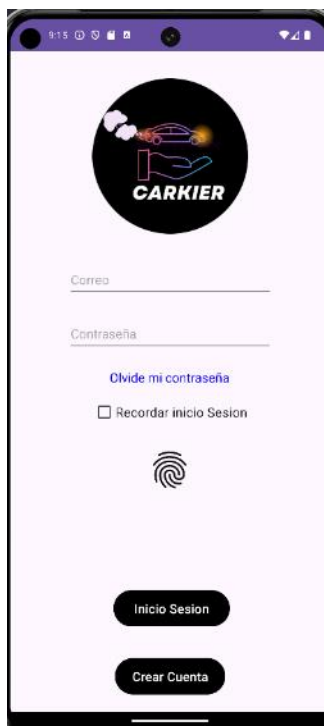
No

ESQUEMA APLICACIÓN MÓVIL



APLICACIÓN MÓVIL

En esta aplicación lo utilizaran cualquier usuario ya que esta aplicación sirve para alquilar nuestro servicio tendría las siguientes pantallas La principal sería un video de la intro y después llevaría al inicio sesión que sería esta foto:



Aquí podrás ver el inicio sesión que necesitarías introducir tu correo y contraseña puedes recordar el inicio sesión para que no estes todo el rato iniciando sesión a no ser que le des a cerrar sesión. Si se te ha olvidado la contraseña le das olvide mi contraseña que te abrirá una nueva ventana en el que tendrás que poner un correo y te llegara un correo esto es para un futuro.

Si tienes cuentas pues introduce el correo y contraseña y le das a iniciar sesión una vez dado te llevara a la principal.

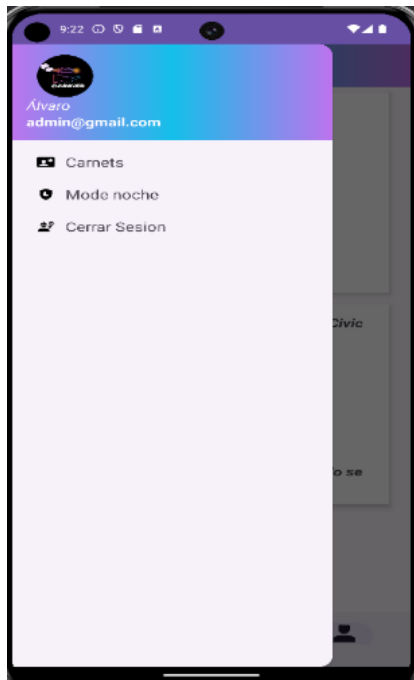


Esta es la ventana principal te saldrán los vehículos que hay a la disposición para vosotros.

Podrás ver que abajo hay una navegación que la casa es la ventana principal que es donde estamos actualmente.

La del medio es un filtrar pero que ahora mismo no está en funcionamiento ya que es para un futuro.

Y el personaje que es para ver los ajustes que te abrirá una ventana lateral como en la siguiente imagen:



En el menú lateral saldrá las siguientes opciones:

Un logo que por defecto saldrá el logo de la aplicación pero en un futuro se pondría la imagen que quiera poner el usuario.

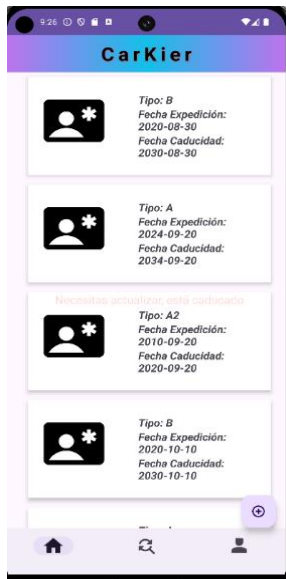
Se mostrará tu nombre y correo y luego varias opciones:

Carnets donde ahí podrás darle y te llevara a una ventana donde te muestran tus carnets.

Modo noche para poner la aplicación en modo noche y cerrar sesion para que a la hora de iniciar sesion debas introducir tu nombre y contraseña.

CARNETS

Pues una vez dado nos cargara una ventana donde se verán los carnet que tienes actualmente.



Como podrás ver se verán todos los carnets de conducir y si hay algún carnet de conducir caducado te avisar con un texto en la imagen no se puede apreciar, pero el 3 carnet hay una frase rojo escrita diciendo que esta caducado el texto se ve intermitente es decir se muestra se quita se muestra y así todo el rato.

NUEVO CARNET

Si queremos agregar un carnet nuevo le daremos al botón que hay encima del menú de abajo que hay un más y se abrirá la siguiente ventana:

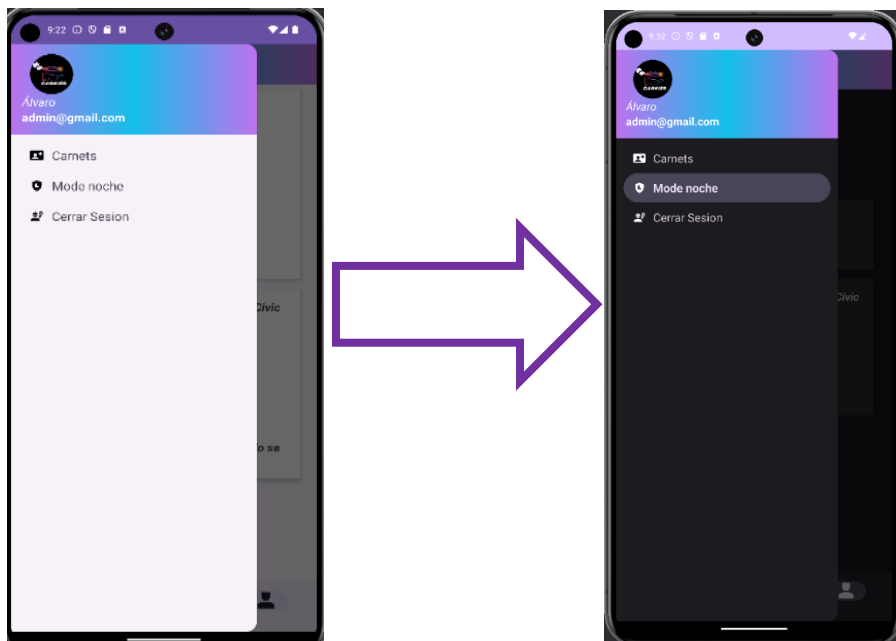


Aquí tendrás que elegir el tipo de carnet y la fecha en la que lo has sacado y solo con eso crearías el carnet.

Si queremos volver para atrás solo hay que darle a la casita de menú de abajo y volveríamos a la ventana principal.

MODO NOCHE

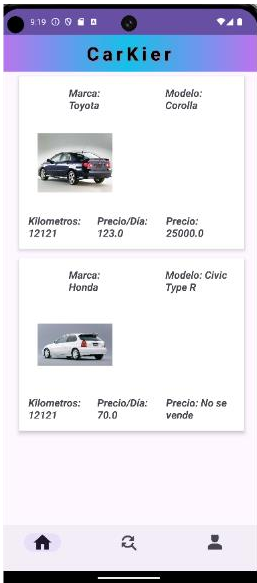
Esta función la he puesto sobre todo para la gente que tiene el móvil en modo nocturno es una función en que las ventanas en vez de ser blancas pasan a ser oscuras y los textos de otro color es decir pasamos.



Y Si le damos a cerrar sesión volveríamos a la ventana del inicio sesión y te tocaría volver a iniciar sesión.

Como hacemos para querer contratar un vehículo para ello iremos a la ventana principal, le daremos clic al coche que queremos elegir y se abriría lo siguiente.

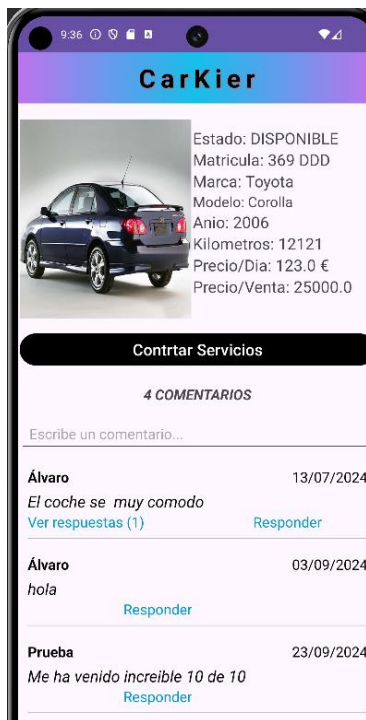
VER VEHÍCULO



Quando estemos en la ventana principal y hay algún vehículo que te interesa tendrás que darle en el y nos abrirá una ventana nueva con las especificaciones al vehículo, el botón de contratar dicho vehículo y los comentarios



CONTRATAR SERVICIOS



Estando en la ventana del vehículo le daremos el botón de contratar servicio esto nos abrirá una nueva ventana donde tendrás que rellenar algunos campos

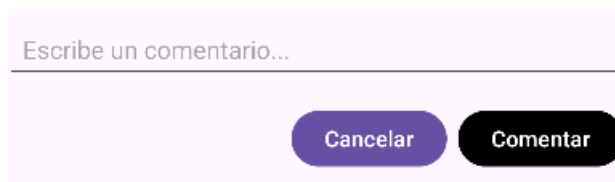


Para contratar tendrás que darle al botón de contratar y rellenar estos datos una vez rellenados le das a contratar y listo ya tendrías contratado el vehículo.

Cuando introduces las fechas saldrá un calendario y ahí puedes elegir desde que fecha a que fecha.



Como hago para escribir un comentario del vehículo para ello estando en esta navegación de la imagen le das a escribir comentario.



Cuando le des se aumentará y se verá dos botones para comentar y cancelar escribes y si le das a comentar se creará el comentario y lo podrá ver todo el mundo.

Para ver las respuestas del comentario si hay respuesta se verá en el comentario ver respuesta como lo ves en el primer comentario.

7-. PLANIFICACIÓN

DIAGRAMA DE GANTT

[illegible]

a) Definición de recursos y logística necesarios

Los materiales que necesitamos para el diseño son:

- 1) MySQL
- 2) Android Studio
- 3) IntelliJ Spring Boot
- 4) Swagger
- 5) Microsoft office

Par poder utilizar las aplicaciones necesitaremos

- 1) Móvil Android
- 2) Ordenador
- 3) Tablet

Para los recursos humanos

Necesitaríamos gente con un técnico superior de aplicaciones multiplataforma

8-. IMPLEMENTACIÓN/ DESARROLLO

8.1 IDE DESARROLLO

Para la creación del proyecto se ha desarrollado diferentes (IDE) para cada parte del proyecto.

IDE PARA EL DESARROLLO

Para la creación del proyecto, se seleccionaron diferentes entornos de desarrollo integrados (IDEs), cada uno adecuado para el tipo de plataforma que se iba a implementar.

8.1.1 API

El desarrollo de la API se realizó utilizando Spring Boot, una herramienta poderosa que facilita la creación de aplicaciones backend. Este framework es totalmente compatible con IntelliJ IDEA Ultimate, lo que proporciona un entorno de desarrollo robusto y eficiente.

8.1.2 APLICACIÓN MÓVIL

Para el desarrollo de la aplicación móvil, se eligió Android Studio, ya que ofrece un conjunto de herramientas especializadas para el desarrollo en plataformas Android, optimizando tanto el diseño como la implementación.

8.1.3 APLICACIÓN DE ESCRITORIO

La aplicación de escritorio fue desarrollada en Visual Studio, un IDE específico de Microsoft que ofrece un entorno optimizado para la creación de aplicaciones de escritorio, asegurando alto rendimiento y una integración perfecta con las herramientas de desarrollo de Windows.

8.2 LENGUAJES DE PROGRAMACIÓN

Dependiendo de que parte del desarrollo se ha implementado diferentes tipos de lenguajes.

8.2 LENGUAJES DE PROGRAMACIÓN

El desarrollo del proyecto ha requerido el uso de distintos lenguajes de programación, cada uno seleccionado en función de las necesidades específicas de cada parte del sistema.

8.2.1 BASE DE DATOS

La base de datos se gestionó utilizando MySQL, un sistema de gestión de bases de datos relacional que emplea SQL (Structured Query Language), el cual es adecuado para la consulta y manipulación eficiente de datos.

8.2.2 API

La API fue desarrollada en Java, un lenguaje ampliamente compatible con el entorno de Spring Boot y conocido por su fiabilidad, seguridad y capacidad de escalabilidad, ideal para construir servicios web.

8.2.3 APLICACIÓN MÓVIL

Para la aplicación móvil, se optó por Kotlin, un lenguaje que se integra perfectamente con Android Studio, ofreciendo un desarrollo más limpio, conciso y eficiente, con soporte completo para la plataforma Android.

8.2.4 APLICACIÓN DE ESCRITORIO

La aplicación de escritorio se desarrolló utilizando C#, un lenguaje de programación desarrollado por Microsoft que es altamente eficiente para la creación de aplicaciones de escritorio, especialmente en el entorno de Visual Studio.

Este enfoque proporciona una visión clara de las decisiones tecnológicas tomadas, destacando la compatibilidad y los beneficios de cada herramienta y lenguaje utilizado en el desarrollo del proyecto.

8.3 Librerías Utilizadas

8.3.1 MYSQL

Para la gestión y desarrollo de la base de datos, se ha elegido MySQL Workbench, una herramienta integral que facilita la administración de bases de datos MySQL, ofreciendo una interfaz gráfica para el diseño, desarrollo y administración de las bases de datos.

8.3.2 API

- Hibernate: Se utiliza para el manejo de la persistencia de datos a través de JPA (Java Persistence API). Hibernate proporciona una forma eficiente de interactuar con la base de datos, simplificando la implementación de la capa de acceso a datos.

- argon2-jvm: Se emplea para la encriptación de contraseñas al momento de crearlas. Este algoritmo es robusto y seguro, lo que garantiza que las contraseñas se almacenen de manera segura.

8.3.3 APLICACIÓN DE ESCRITORIO / GESTIÓN

- NET Framework: Es la tecnología utilizada para el desarrollo de la aplicación de escritorio, ofreciendo un entorno de desarrollo sólido y eficiente para aplicaciones Windows.
- Newtonsoft.Json: Se utiliza para la serialización y deserialización de objetos JSON. Esta biblioteca facilita la conversión entre objetos y formatos JSON, lo que es esencial para la interacción con servicios web.

8.4 CONTROL DE VERSIONES

Para el control de versiones, se ha elegido GitHub como plataforma para el alojamiento de los repositorios. GitHub facilita el manejo del código fuente y la colaboración entre los miembros del equipo.

El repositorio de GitHub se organiza en carpetas de manera estructurada, lo que facilita la navegación y gestión de los diferentes elementos del proyecto, asegurando una mejor organización y acceso rápido a los archivos.

Este enfoque mejora la claridad y coherencia, explicando detalladamente el propósito de cada herramienta y su contribución al proyecto.

8.5 CONVENCIONES DE CODIFICACIÓN Y ESTÁNDARES SEGUIDOS

Estándar de Nomenclatura

La nomenclatura utilizada en el código sigue un enfoque claro y organizado, lo que facilita la comprensión y mantenimiento del proyecto. Las siguientes pautas se han seguido:

- **Clases:** Los nombres de las clases reflejan claramente su propósito o funcionalidad. Esto permite identificar de manera rápida la responsabilidad de cada clase dentro del sistema.
- **Métodos:** Los nombres de los métodos describen explícitamente la acción que realizan. Esto mejora la legibilidad y hace que el propósito de cada función sea fácilmente identificable.
- **Variables:** Se eligen nombres descriptivos que indican claramente la función de cada variable, evitando ambigüedades y mejorando la comprensión del código.

Este enfoque mejora significativamente la lectura y mantenimiento del código, ya que facilita la comprensión inmediata de su propósito, incluso para otros desarrolladores que trabajen en el proyecto.

Uso de Formateadores de Código

Para asegurar que el código siga un estilo coherente y elegante, se han utilizado herramientas de formateo integradas en los IDEs, como el formateador de código de **IntelliJ IDEA** y **Android Studio**. Estas herramientas ayudan a mantener un formato consistente a lo largo del código, lo que mejora la legibilidad y facilita su mantenimiento.

Esta versión mejora la claridad y la estructura, haciendo que las convenciones de codificación y el uso de herramientas sean más comprensibles y detallados.

9-. DESPLIEGUE Y MANTENIMIENTO

Buscaremos un servidor para que este en continuamente en funcionamiento, la que utilizaríamos seria Amazon Web Services ya que permite tener un entorno seguro y eficiente para la aplicación.

9.1-. APLICACIÓN MÓVIL

9.1.1 DESARROLLO

Cuando se lance la aplicación se llevaría pruebas unitarias para el correcto funcionamiento y estabilidad, el proyecto se compilará y empaquetará en un archivo Android Package (APK) o Android App Bundle (AAB).

Para la distribución en dispositivos Android, estas medidas garantizan no tenga errores y está lista para los usuarios finales.

9.1.2 DISTRIBUCIÓN

La aplicación saldría al público y se descargaría en la aplicación de Google Play Store, para facilitar al usuario la instalación y actualización directamente desde esta plataforma.

Asegura mayor accesibilidad y comodidad para los usuarios que quieran instalar la aplicación en sus dispositivos Android.

9.2 APLICACIÓN ESCRITORIO

9.2.1 DESARROLLO

Para el Sistema Operativo Windows, se genera un .exe que es el formato estándar para la instalación de aplicación de este entorno.

En un futuro se aplicará también al Sistema Operativo MacOS, en este se deberá empaquetar en un archivo .dmg.

9.2.2 DISTRIBUCIÓN

La aplicación se publicará en la tienda de a Microsoft Store y en un futuro se creará una pagina web que contará con un enlace para la descarga de la aplicación.

9.3 ACTUALIZACIONES

Para mantener a los usuarios satisfechos, podrán contactar con nosotros para mandarnos errores o bugs que detecten. Ya que la aplicación contara con actualizaciones para solucionar los bugs y agregar más funcionalidades.

9.3.1 CICLO DE ACTUALIZACIONES

La aplicación tendrá un ciclo regular de actualizaciones que están categorizadas por:

- ➔ Mayores: Estas incorporan nuevas funcionalidades a la aplicación para mejorar la experiencia del usuario.
- ➔ Menores: Estas incorporaran soluciones para los errores.

Cada actualización vendrá con un registro de cambios accesible desde la aplicación, donde se detallarán específicamente los cambios que han introducido

10-. PRUEBAS

Para asegurar el correcto funcionamiento de las aplicaciones, para que cumpla con los requisitos esperados se define un plan de pruebas detallado.

Utilizaremos diferentes tipos de pruebas:

→ Revisión de métodos:

Cada método individualmente se probará para la comprobación de su funcionamiento y devuelva los resultados específicos.

Esto permite identificar el error en etapas de desarrollo.

→ Pruebas unitarias:

Estas pruebas se enfocan en verificar cada unidad de código (como clases o funciones) de manera independiente. Se diseñarán y ejecutarán casos de prueba específicos para comprobar que cada parte del sistema funciona según lo esperado, aislado del resto del programa.

→ Prueba de integración: Este tipo de pruebas consisten en combinar unidades de código y verificar que funciona correctamente. Estas pruebas se hacen después de cada ciclo de desarrollo.

→ Pruebas de sistema: Se realizarán pruebas del sistema completo para garantizar que cumpla los requisitos funcionales y no funcionales. Para

asegurar que el sistema funcione correctamente. Estas pruebas se hacen antes del de cada lanzamiento.

→ Pruebas de Seguridad: Se harán pruebas para identificar vulnerabilidades de seguridad, para asegurar que la aplicación sea segura y no hallan amenazas ni ataques.

→ **Diseño de Pruebas**

Las pruebas se diseñarán durante las siguientes fases del desarrollo:

- Fase de Requisitos: Se identificarán los criterios de aceptación y se definirán los casos de prueba iniciales basados en los requisitos del cliente.
- Fase de Diseño: Se detallarán los casos de prueba específicos para cada unidad y módulo. También se definirán las condiciones exactas de las pruebas, como los datos de entrada, los resultados esperados y los pasos a seguir.
- Fase de Implementación: Se desarrollarán y documentarán los casos de prueba.

10-. GESTIÓN ECONÓMICA

En este apartado hablaremos de cómo será la gestión de la economía de un proyecto profesional.

10.1-. DENOMINACIÓN Y CATEGORÍA

- Nombre de la empresa: CarKier
- Categoría: Servicio de alquiler de vehículos a corto, mediano o largo plazo.
- Misión: Proveer un servicio accesible y confiable de alquiler de vehículos.
- Visión: Darnos a conocernos en el mundo y ser los mejores.

10.3-. INSTALACIÓN

Vamos a hablar de cómo será la instalación donde estará ubicada.

Está ubicada en Santander y la instalación contará con:

- ➔ Oficina Administrativa, recepción y área del trabajo
- ➔ Aparcamientos para 30 vehículos incluida puestos de carga para coches eléctricos.
- ➔ Zona de limpieza de vehículos (En un futuro se querrá implementar)

10.4-. SERVICIOS

- Ofreceremos servicios de Alquileres de vehículos para uso:
 - Diario
 - Semanal

- Mensual.
- Para los clientes que lleven tiempo conseguirán puntos con los que podrán reducir el coste del servicio

10.5-. TIPOS DE CLIENTES

Cualquier usuario podría adquirir nuestro servicio:

- Particulares: Cualquier turista o cualquier residente local que obviamente tengan carnet de conducir.
- Empresas: Si requieren algún vehículo durante x tiempo también serán bienvenidos.

10.5-. NIVEL DE OCUPACIÓN

- Flota Inicial: 20 vehículos (10 coches normales, 5 coches eléctricos, 5 furgonetas).
- Tasa de ocupación provista: Será del 50% de vehículos mensual mínimo.

10.5-. ESTRUCTURA ORGANIZATIVA

- Gerente: Gestión en general
- Atención al cliente: Gestiona las reservas, soporte y mantenimiento.
- Plantilla inicial: 1

10.6-. ANÁLISIS ECONÓMICO

Financiación	
Aportación Propia	60.000 €
Préstamo bancario	280.000€

El prestamos se devolverá a 5 años con un interés del 5 %.

Concepto	Coste €
Flota	300.000 €
Publicidad	5.000 €
Otros gastos (agua, luz, etc.)	5.000 €
Alquiler y acondicionamiento	20.000 €
Software y hardware	10.000 €
TOTAL	340.000 €

Ingresos	
Categoría	€
Ingresos Servicios	23.000€
Anuncios en la Aplicación	Depende de los usuarios diremos que 1000 € al mes
Gastos	
Alquiler local	1500 €
Sueldos	1500 €
Mantenimiento	8000 €
Seguros	2500 €
Publicidad	1000 €

Gastos extras (Mobiliarios, etc.)	3500 €
Total, Gastos	16.000€
beneficio	8.000€
Beneficio Anual	94.000 €

11-. CONCLUSIÓN

El proyecto es financieramente sostenible, con una inversión inicial que se puede recuperar en menos de 2 años. Asimismo, dispone de un amplio potencial de expansión, lo que facilita incrementar la flota y los servicios a medida que crezca la demanda.