

UNIVERSIDAD DON BOSCO



FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

DISEÑO Y PROGRAMACIÓN DE SOFTWARE MULTIPLATAFORMA

PRIMERA FASE DEL PROYECTO DE CÁTEDRA

ASESOR:

ING. MARIO ALVARADO - DPS G04T

ING. MIGUEL ORELLANA - DPS G03T

PRESENTADO POR:

DPS G04T

JOSÍAS ISAAC ALVARENGA ROMERO

#AR200778

LUIS ANTONIO BONILLA ELIAS

#BE202156

ALFONSO ANTONIO FERNÁNDEZ COTTO

#FC200910

KAREN SOFÍA RIVAS RIVAS

#RR201098

DPS G03T

DIEGO FERNANDO GÓMEZ CASTRO

#GC201085

DPS104 G04T Y G03T

• 03 • SEPTIEMBRE 2023 •

ANTIGUO CUSCATLÁN

ÍNDICE

- 01** Introducción
- 02** Descripción del proyecto
- 03** Metodología de trabajo
- 04** Mockups
- 05** Lógica de desarrollo
- 06** Herramientas
- 07** Presupuesto
- 08** Licencias Creative Commons
- 09** Bibliografía

INTRODUCCIÓN

En el contexto de nuestra innovadora aplicación de gestión de reclamos de seguros automovilísticos, nos centramos en mejorar la eficacia y la comunicación entre aseguradoras, talleres y proveedores, simplificando el proceso de reclamos en la industria automovilística.

Nuestra aplicación es la respuesta a esta necesidad, al proporcionar una plataforma integral que permite a las aseguradoras, talleres y proveedores colaborar de manera efectiva y ágil en el manejo de reclamos automovilísticos.

En el sector automovilístico, los activos involucrados son los vehículos y los servicios relacionados. Nuestra aplicación simplifica la gestión de reclamos al proporcionar una vía digital detallada y ordenada para documentar y procesar reclamos, permitiendo una asignación eficiente a talleres asociados y la obtención de repuestos necesarios.

El control de reclamos es de suma importancia, ya que los costos asociados a los mismos representan un aspecto crítico en la industria de seguros automovilísticos. Con nuestra aplicación, ofrecemos herramientas específicas para aseguradoras, talleres y proveedores que reducen los problemas comunes, como la falta de control detallado o las fluctuaciones bruscas en la asignación de reclamos.

Este desafío se manifiesta en todos los aspectos de la industria automovilística, donde la gestión de reclamos es fundamental. Nuestra aplicación aborda estos desafíos y promueve una comunicación fluida y una colaboración efectiva entre las partes involucradas, mejorando así la experiencia de gestión de reclamos de seguros automovilísticos en todas las etapas del proceso.

DESCRIPCIÓN DEL PROYECTO

AutoRepair Linker es una aplicación móvil de gestión de reclamos de seguros automovilísticos. Esta es una solución diseñada para agilizar el proceso de reclamos entre las aseguradoras de automóviles, los talleres asociados a las mismas y proveedores de repuestos de automóviles.

Características claves para aseguradoras:

- Gestión de talleres asociados: adición y eliminación de talleres asociados
- Gestión de tickets: permitirá abrir reclamos digitales con los detalles para la reparación de los vehículos
- Asignación de talleres: las aseguradoras podrán gestionar la asignación de reclamos a los talleres asociados

Características claves para talleres:

- Gestión de tickets: los talleres pueden rechazar, aceptar y actualizar el estado de los tickets asignados por las aseguradoras
- Obtención de materiales: los talleres pueden buscar y realizar solicitudes de materiales en el AutoParts Marketplace integrado en la plataforma

Características claves para proveedores:

- Gestión de inventario disponible: los proveedores de repuestos asociados podrán gestionar un inventario de repuestos disponibles

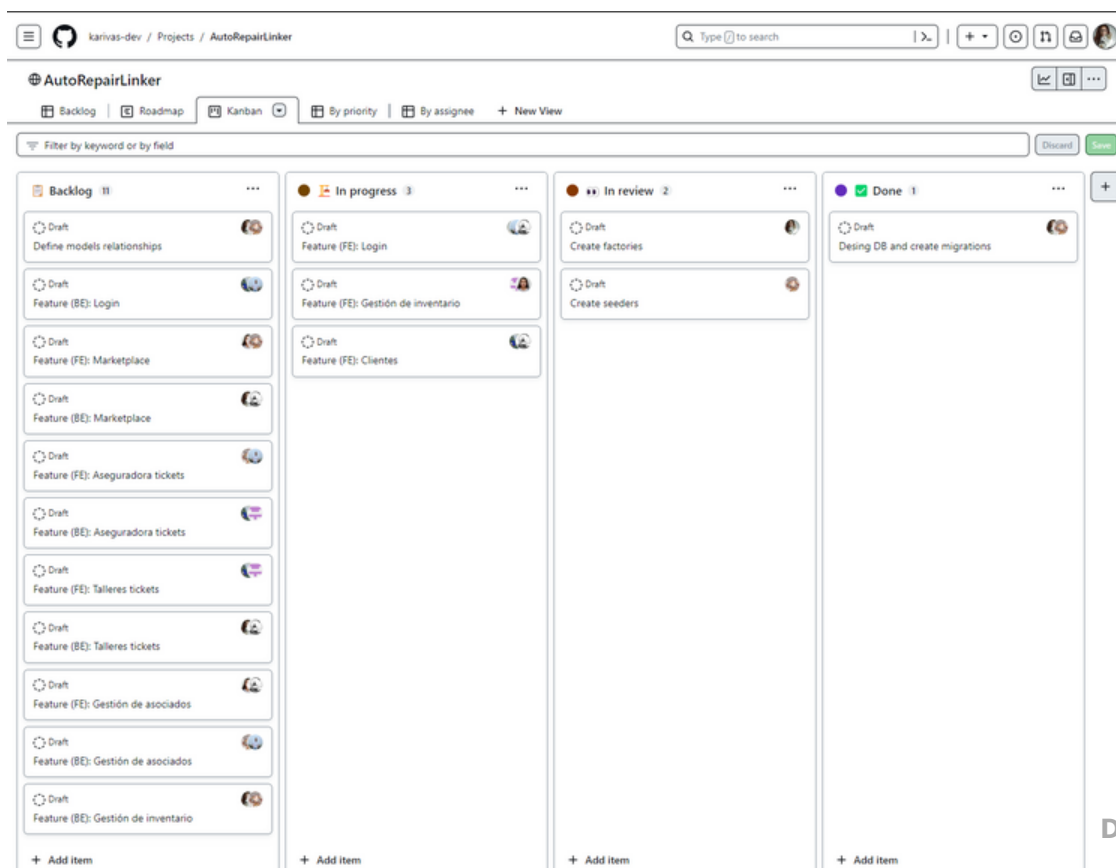
METODOLOGÍA DE TRABAJO (KANBAN)

En este proyecto se utiliza la metodología Kanban para el proceso de gestión de las tareas a desarrollar. La metodología se basa en la visualización del flujo de trabajo y el control de la capacidad de trabajo en tiempo real mediante la utilización de tarjetas o "kanban boards".

En esta metodología los equipos utilizan un tablero Kanban para visualizar este flujo de trabajo y para llevar un registro de los proyectos en curso, las tareas completadas y las tareas pendientes. Las tarjetas en el tablero Kanban representan tareas específicas y se mueven a través de diferentes columnas que representan etapas del proceso, como "Por hacer", "En progreso" y "Completado".

El enfoque en la visualización del trabajo y la capacidad real permite a los equipos tomar decisiones informadas sobre la priorización y la planificación del trabajo. Además, la metodología Kanban fomenta la entrega continua y la mejora continua mediante la retroalimentación constante y el ajuste del proceso en función de los resultados.

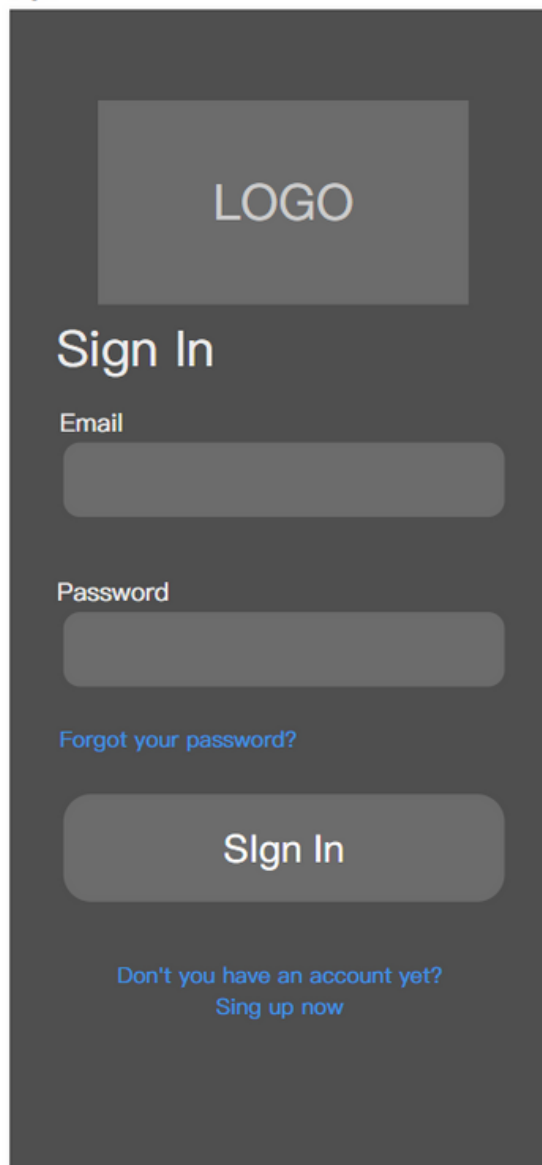
Nuestra implementación realizará las tarjetas o tablero Kanban en GitHub projects, donde podemos agregar tareas, asignarlas, e incluso marcarlas según la etapa de desarrollo en la que se encuentren. Se puede encontrar en el repositorio del proyecto en la pestaña Projects.



MOCKUPS

Para proporcionar una comprensión más detallada de los mockups, hemos definido varios tipos generales que son comunes en diversos módulos o áreas. Estas plantillas nos permiten abordar respuestas similares con ajustes específicos según el contexto. A continuación, se presentan algunas de las pantallas clave que hemos diseñado:

Login

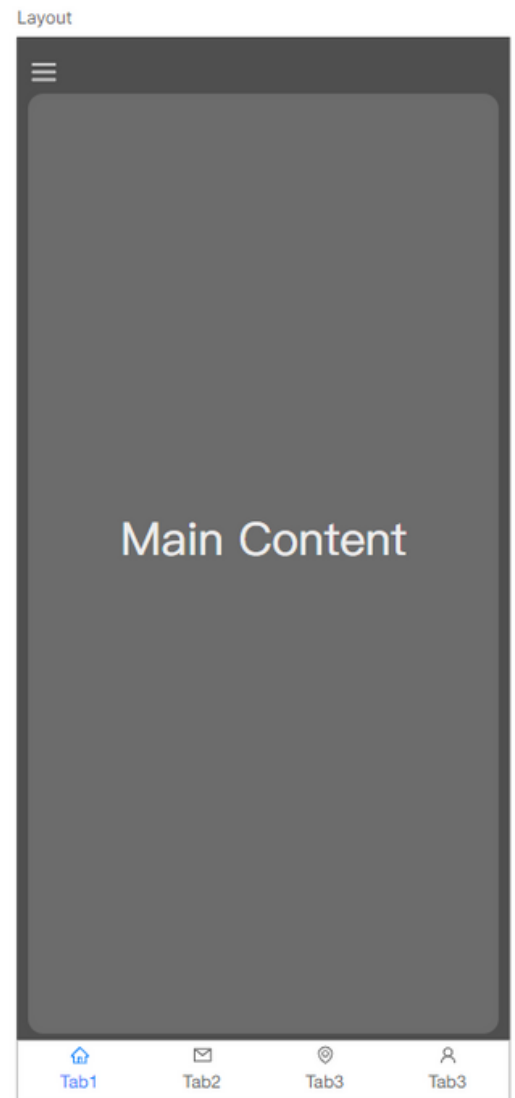
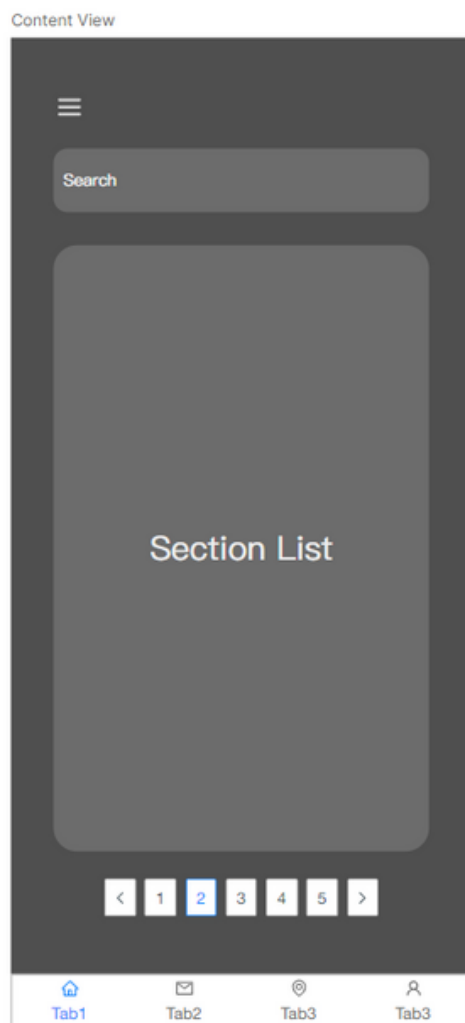
A dark gray login form mockup. At the top is a light gray rectangular box containing the word "LOGO" in white capital letters. Below the logo is the text "Sign In" in a white sans-serif font. Underneath are two input fields: the first is labeled "Email" in a small white font, and the second is labeled "Password" in a small white font. Below the password field is a link "Forgot your password?" in a light blue font. At the bottom of the form is a large, rounded rectangular button with the text "Sign In" in white. Below the button is another link "Don't you have an account yet? Sing up now" in a light blue font.

Este mockup representa la página inicial de autenticación. Aquí se exhibe la disposición de los campos de entrada para el nombre de usuario y la contraseña, así como el botón para acceder.

Si bien los elementos visuales como colores y logotipos pueden variar según el proyecto, esta plantilla nos ayuda a establecer la estructura básica de la página de inicio de sesión.

El mockup de diseño general establece la estructura visual básica de una página o pantalla. Incluye áreas para encabezados, menús de navegación, contenido principal y pies de página.

Aunque los elementos de diseño y la paleta de colores pueden cambiar según la marca o el contexto, esta plantilla nos permite comprender cómo se organizarán los diferentes componentes en la interfaz.



El mockup de la pantalla de lista muestra cómo se presentará un conjunto de elementos en una vista de lista. Esto puede aplicarse a talleres, aseguradoras, proveedores u otros elementos similares.

Aquí definimos cómo se mostrará cada entrada de la lista, qué información se incluirá y cómo se organizará. Esto nos ayuda a crear una experiencia coherente en la presentación de múltiples elementos.

En esta plantilla, delineamos la disposición de un formulario, ya sea de edición o de creación. Esto implica definir los campos necesarios, como dependiendo del contexto que este trate, para así tener un correcto funcionamiento en actualizaciones o nuevas inserciones en el sistema.

La estructura general del formulario, los botones de envío y los mensajes de validación son elementos clave que podemos establecer mediante este mockup.

Form

The mockup shows a mobile app screen titled 'Create / Edit'. At the top left is a back arrow. Below the title are four labeled input fields: 'Field 1' (a simple text box), 'Field 2' (a simple text box), 'Field 3' (a dropdown menu with '1' selected), and 'Field 4' (a larger text area). At the bottom right is a 'Save' button.

Item View

The mockup shows a mobile app screen titled 'Item View'. At the top left is a back arrow. Below it are two buttons: 'photo' (circular) and 'Item' (rounded rectangle). Below these are two more buttons: 'Edit' (rounded rectangle) and 'Item Status' (rounded rectangle). At the bottom is a large rectangular area labeled 'Item Details'.

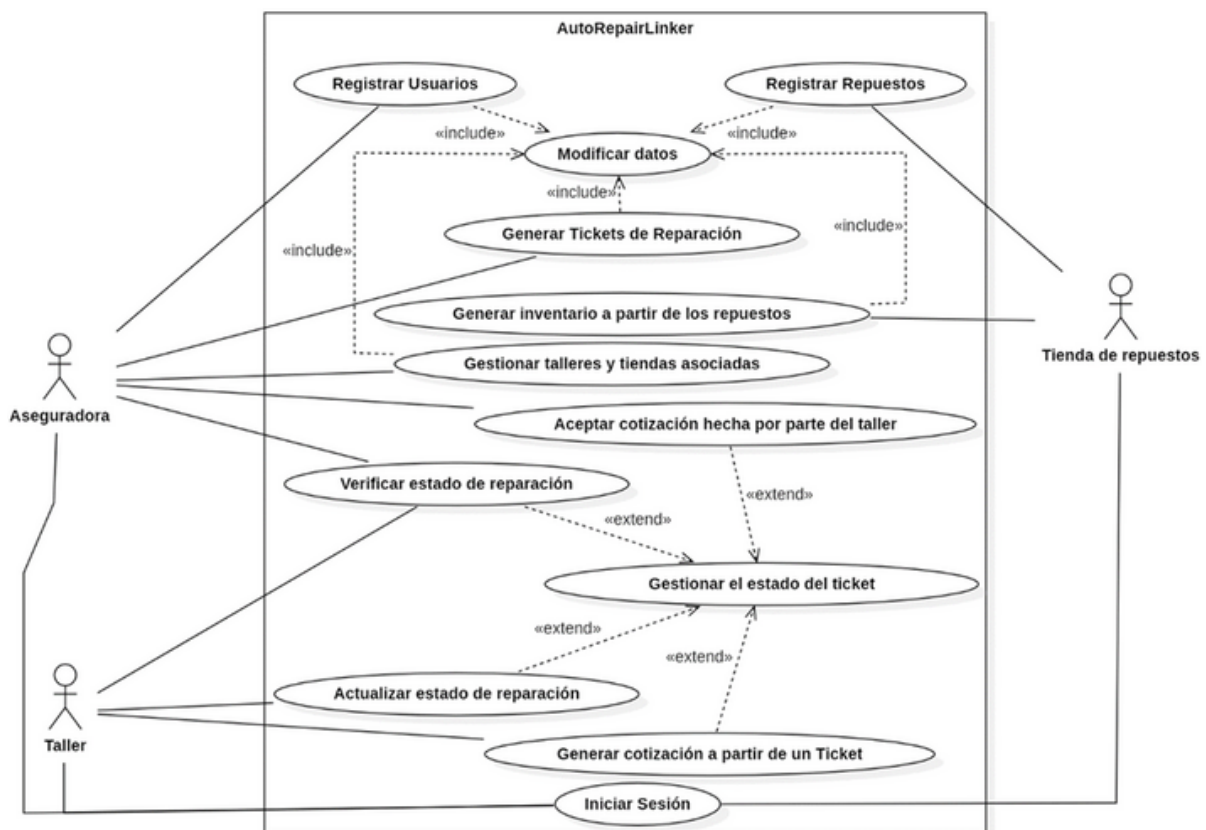
En este mockup, representamos cómo se presentarán los detalles específicos de un ítem o registro especificado.

Aquí se incluyen elementos como imágenes, descripciones detalladas y botones para acciones relacionadas, además de incluir información sobre las relaciones que tenga dicho elemento.

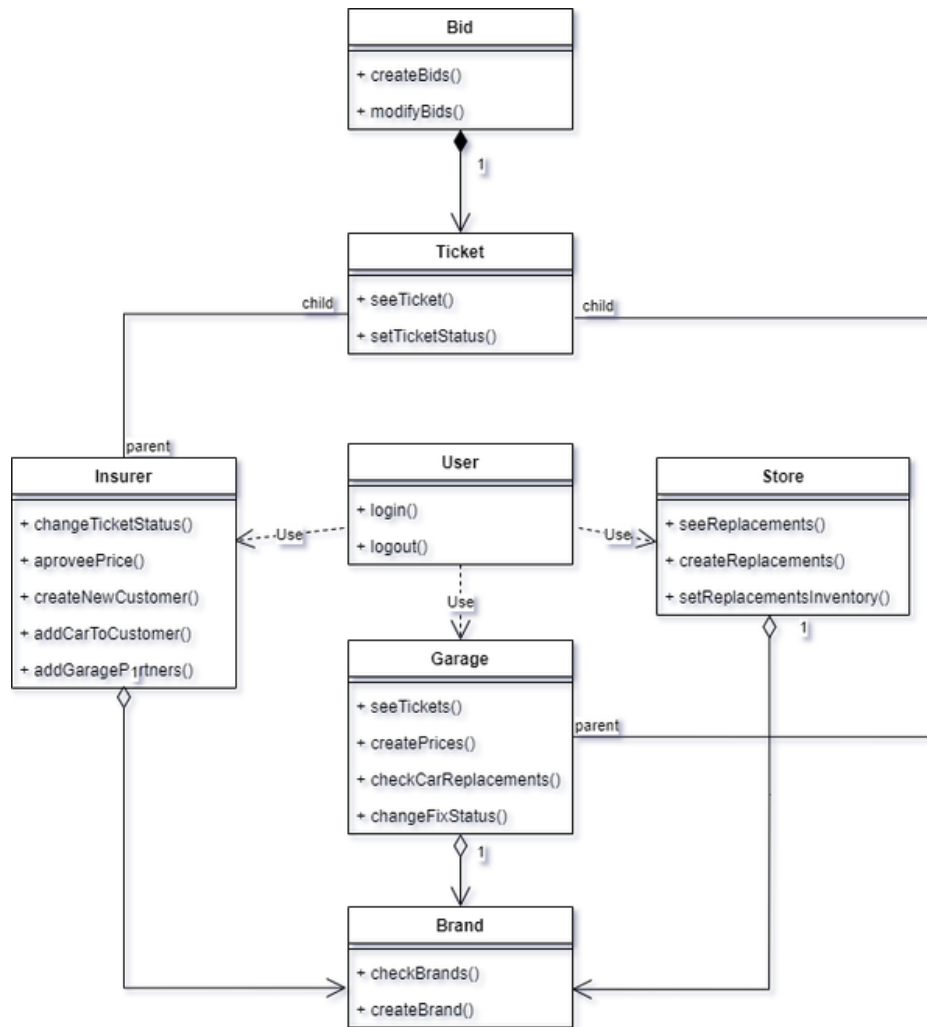
Aunque los atributos y el contenido del ítem pueden variar, este mockup nos brinda una estructura para resaltar características clave.

LÓGICA DE DESARROLLO

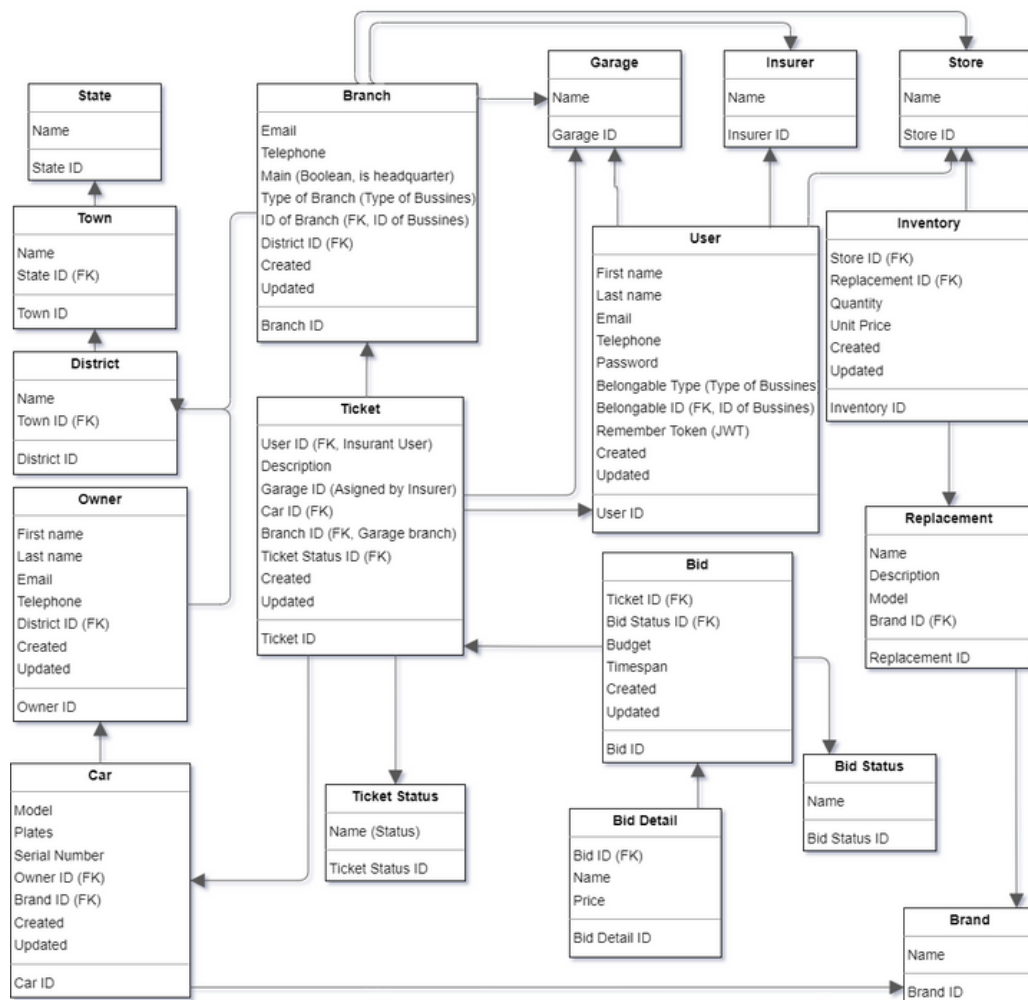
En el desarrollo de este proyecto se ha planteado el siguiente diagrama de caso de uso, el cual describe un poco la funcionalidad del mismo, los aspectos que hay que tener en cuenta y sobretodo los actores principales del mismo.



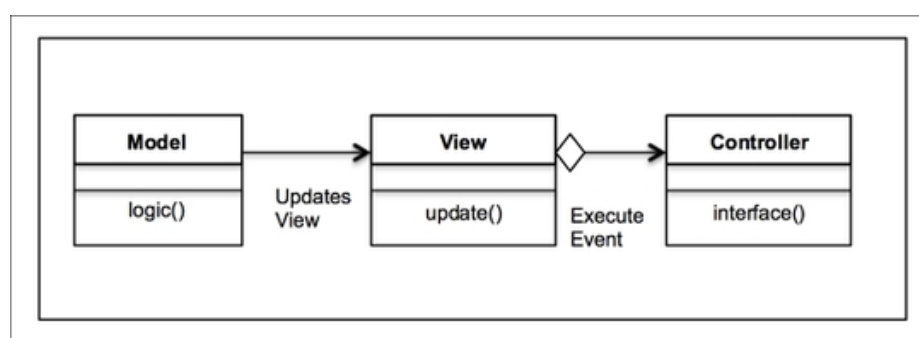
A continuación, se presenta el diagrama de clases. En el diagrama de clases se encuentra el usuario quien puede ingresar y salir de su cuenta. Por otra parte, se especifican las aseguradoras (insurer), talleres (garage), y tiendas distribuidoras (store). Se contempla la creación de marcas para identificar a los vehículos y los repuestos distribuidos.



También podemos destacar que para cumplir con el diagrama anterior, tenemos que plantear una correcta estructuración de los datos que se manejarán en el sistema, es por ello que hemos planeado la siguiente estructuración de los datos, donde se pueden identificar las tablas, campos, claves primarias y foráneas, y las relaciones que existen entre ellas, lo que facilita el diseño, la optimización y el mantenimiento de bases de datos eficientes y coherentes.



De forma general se ha trabajado con el modelo MVC o modelo vista controlador, que nos ha permitido obtener todos los datos en los modelos, generar controladores para el manejo de los modelos y las vistas.



HERRAMIENTAS

Para el desarrollo de este proyecto se han optado por frameworks y librerías de gran demanda en el desarrollo en la actualidad. Entre las herramientas a utilizar encontramos:

React Native



React Native es un marco de trabajo de código abierto que combina las mejores partes del desarrollo nativo con React, una biblioteca de JavaScript de primera clase para construir interfaces de usuario. React Native le permite crear aplicaciones verdaderamente nativas y no compromete las experiencias de sus usuarios. Contiene recursos como componentes preconstruidos, bibliotecas y material de referencia .

NativeWind



NativeWind utiliza Tailwind CSS como lenguaje de scripting para crear un sistema de estilo universal para React Native. Los componentes de NativeWind se pueden compartir entre plataformas y generarán sus estilos como CSS StyleSheet en la web y StyleSheet.create para nativo. Le permite utilizar los mismos componentes con estilos ricos en todas las plataformas React Native.

Laravel



Laravel es un marco PHP muy popular que ofrece una gran cantidad de características y herramientas para ayudar a los desarrolladores de software a crear aplicaciones web y APIs de alta calidad. Algunas de las ventajas por las cuales se eligió la implementación de esta tecnología son:

- Facilidad de uso
- Seguridad
- Escalabilidad
- Flexibilidad
- Productividad

PRESUPUESTO

Para el desarrollo de este proyecto se debe tomar en consideración aspectos importantes como los siguiente:

Presupuesto de Desarrollo

Para desarrollo del proyecto se ha estimado un periodo de tiempo de 3 meses (175 horas/hombre de trabajo efectivo). En el equipo de desarrollo es de cinco desarrolladores, los cuales se ha valorado un coste de hora/hombre en US\$ 6.00 con un 5% de imprevistos, además se menciona que en esta fase debe considerarse todas las herramientas a utilizar con sus licencias, las cuales no generan un gasto en el presente proyecto (tanto en el desarrollo del sistema como la gestión de la base de datos).

Presupuesto de Desarrollo			
Rubro	Valor Unitario	Cantidad	Total
Software	--	--	
Visual Studio Code	--	--	
MySQL	--	--	
Android Studio	--	--	
Capital Humano (honorarios profesionales)			
5 Desarrolladores	350	5	1750
Sub Total Desarrollo (US\$)			1750
Imprevistos (%5)			87.50
Sub Total (US\$)			1837.50
Renta por Servicios Profesionales (10%)			175
Total, General (US\$)			2012.50

Presupuesto de Implementación

El presupuesto de implementación es un aspecto clave del éxito de la implementación de un software o sistema, ya que permite a los responsables de la implementación planificar y controlar los costos de manera efectiva y asegurar que se cumplan los objetivos de la implementación dentro del presupuesto asignado, además destacar que para el desarrollo de capacitaciones se ha considerado un periodo de tiempo de 20 horas/hombre de trabajo efectivo, con un coste de hora-hombre en US\$ 3 (No se ha incluido un presupuesto de mantenimiento, ni de imprevistos).

Es fundamental considerar el costo de mantenimiento de la aplicación, ya que es necesario realizar actualizaciones periódicas con el fin de mejorar continuamente el servicio ofrecido a los usuarios. En este sentido, se estima que el costo anual de mantenimiento equivale al 20% del costo inicial de desarrollo de la aplicación.

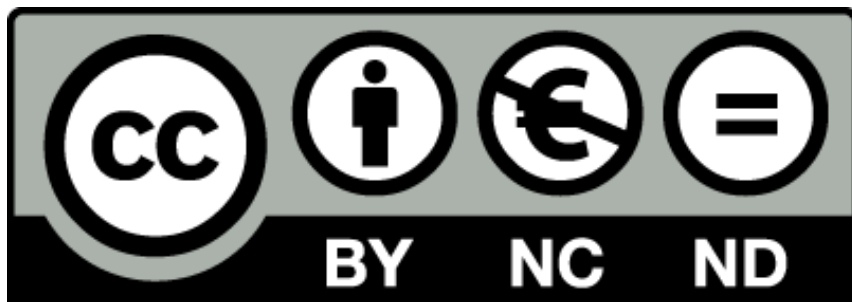
Dado que estamos optando por implementar una API como parte integral del backend de la aplicación, resulta imperativo considerar y presupuestar los costos asociados al hosting de la API y al registro del dominio correspondiente.

Presupuesto de Implementación			
Rubro	Valor Unitario	Cantidad	Total
Software	--	--	
Visual Studio Code	--	--	
MySQL	--	--	
Android Studio	--	--	
Capacitación	20	3	60
Mantenimiento (un año)	1	1	402.50
Dominio (API)	1	1	25
Hosting API (un año)	1	1	180
Total, General (US\$)			667.50

LICENCIA CREATIVE COMMONS

Para este proyecto se utilizan las Licencias de Creative Commons, concretamente:

Reconocimiento - NoComercial - SinObraDerivada (by-nc-nd). Esta licencia nos permitirá que el material creado exclusivamente en el sistema AutoRepairLinker no sea comercializado por terceros ni les permita hacer paginas derivadas de ella, ya que no se quiere que la competencia se aproveche de dicha sistema por lo cual esa es la mejor licencia que cumple con todas las expectativas.



ENLACES

- Repositorio de AutoRepair Linker API (Frontend) <https://github.com/karivas-dev/AutoRepairLinker>
- Repositorio de AutoRepair Linker API (Backend) <https://github.com/karivas-dev/AutoRepairLinker-API>
- Planificación de proyecto <https://github.com/users/karivas-dev/projects/4>
- Mockups https://mockitt.wondershare.com/proto/WVoB5eNgrzrg9ia4rWNhXS/sharing?view_mode=read_only

BIBLIOGRAFÍA

- Licenciatura en Sistemas y Computación, T. (n.d.). Metodologías de desarrollo de software. Edu.Ar. Retrieved February 4, 2023, from <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Gilibets, L. (2023). Qué es la metodología Kanban y cómo utilizarla. Thinking for Innovation. <https://www.iebschool.com/blog/metodologia-kanban-agile-scrum/>
- Installation. (n.d.). Laravel.com. Retrieved February 4, 2023, from <https://laravel.com/docs/9.x>
- Home | NativeWind. (n.d.). <https://www.nativewind.dev/>
- Setting up the development environment · React Native. (2023, August 29). <https://reactnative.dev/docs/environment-setup>
- Sobre las licencias. (s/f). Creativecommons.org. Recuperado el 2 de septiembre de 2023, de <https://creativecommons.org/licenses/?lang=es>