



UNIVERSIDAD
VERACRUZANA

Facultad de estadística e informática

Ingeniería de software
Desarrollo de sistemas web

Proyecto Final

Javier Alberto Calderón Blas
Negrete García Eder Iván
Ángel Eduardo Martínez Leo Lim
Raul Arturo Peredo Estudillo

Xalapa, Ver.
18 de diciembre de 2022

Contenido

Introducción	1
Sustento metodológico	2
EDT.....	3
Artefactos de Diseño.....	7
Diagrama de clases.....	7
Diagrama de CU	7
Diagrama relacional	9
Diagrama Navegacional Conductor.....	10
Diagrama Navegacional Ajustador	10
Diagrama Contexto Ajustador	11
Diagrama contexto Conductor.....	12
Descripciones de CU.....	13
CU. – Iniciar sesión.....	13
Descripción de caso de uso	13
CU. – Registrar cuenta	14
Descripción de caso de uso	14
CU. – Registrar reporte.....	15
Descripción de caso de uso	15
CU. – Modificar cuenta	17
Descripción de caso de uso	17
CU. – Comprar póliza	19
Descripción de caso de uso	19
CU. – Pagar póliza.....	20
Descripción de caso de uso	20
CU. – Consultar pólizas	21
Descripción de caso de uso	21
CU. – Consultar póliza	22
Descripción de caso de uso	22
CU. – Consultar reportes	23
Descripción de caso de uso	23
CU. – Ver detalles reporte	24
Descripción de caso de uso	24
CU.- Iniciar Sesión Ajustador	25
Descripción de caso de uso	25

CU.- Consultar reportes asignados	26
Descripción de caso de uso	26
CU.- Dictaminar Reporte.....	27
Descripción de caso de uso	27
CU.- Ver Detalles Reporte	28
Descripción de caso de uso	28
Pruebas	29
Postman.....	29
Resultado de las pruebas	29
Mocha	31
Código de las pruebas	31
Resultados de pruebas	33
Conclusión	34

Introducción

Nivel Problemática:

La aseguradora HDI Seguros, requiere implementa un sistema informático para llevar la gestión de los reportes de siniestros vehiculares de sus asegurados, esto debido a que actualmente los reportes se hacen de manera manual por un ajustador calificado que se traslada hasta el lugar del siniestro para levantar el reporte y poder dictaminar el hecho, ocasionando principalmente un mayor tiempo de atención.

En este proyecto se busca sistematizar el registro del reporte, para que el mismo conductor involucrado en el siniestro pueda levantar el reporte inicial del siniestro desde una aplicación web mejorando con esto los tiempos de atención ya que el ajustador podría llegar solo a evaluar los daños y conciliar con los involucrados, se requiere que todos los reportes se concentren en la base de datos de la aseguradora y que los reportes realizados puedan ser visualizados y dictaminados por los ajustadores quienes también tendrán acceso al sistema web.

Es necesario para que la aseguradora pueda tener un mejor control tanto con sus clientes y sus empleados de esta manera le facilita la comunicación entre estos dos de igual manera agiliza el proceso en el cual se levanta un reporte ya que muchas veces es necesario que se hagan rápido para no obstaculizar el tráfico ni retrasar a los clientes en sus actividades del día.

Nivel Proyecto:

Se busca que, mediante las técnicas, conceptos y practicas realizadas en la clase de DSR (Desarrollo de Sistemas en Red) se logre cumplir con el objetivo de solución hacia la problemática planteada al inicio de curso y con esto comprobar y poner en practica todo lo aprendido durante el semestre.

Sustento metodológico

OOHDM - Metodología de diseño hipermedia orientada a objetos

Es una metodología orientada a objetos que propone un proceso de desarrollo de cinco fases donde se combinan notaciones graficas UML con otras propias de la metodología

Inicialmente se utilizaba para el desarrollo de aplicaciones de hipermedia básicas, sin embargo, se adaptó para aplicaciones web más complejas y con mayor interacción entre usuarios.

- Utiliza modelos especializados como:
 - Conceptual
 - De navegación
 - Interfaz de usuario
- Su objetivo es hacer más eficaz el diseño de aplicaciones
- Sus etapas son:
 - Obtención de requerimientos: Su objetivo es conocer los actores y funcionalidades que debe contener la aplicación, posteriormente se deben extraer y modelar detalladamente los CASOS DE USO a desarrollar
 - Diseño conceptual: Se realiza a través del modelamiento de DIAGRAMAS DE CLASES que contengan las clases, relaciones y subsistemas que intervienen en cada funcionalidad
 - Diseño navegacional: Se utiliza para representar los diferentes caminos que puede ejecutar la aplicación a nivel general o dependiendo del tipo de usuario
 - Diseño de interfaz abstracta: Es necesario especificar las interfaces de usuario que se visualizaran en la aplicación web

Un aspecto importante que considerar en esta etapa es cuidar que el diseño de las interfaces de usuario coincida con el diseño navegacional

- Implementación: Consiste en realizar la codificación de la aplicación web ya sea de forma nativa o mediante algún gestor de contenido. Durante esta etapa se deben respetar los modelos previamente diseñados y se deben incluir etapas de pruebas unitarias, de integración o de sistema

EDT

ID	RESPONSABLES	FASE	TAREA	DESCRIPCIÓN
TAR-01	Equipo completo	Obtención de requerimientos	Selección de problemática	Se selecciona el problema que requiere el desarrollo de un sistema web
TAR-02	Equipo completo		Elaboración de perfil de proyecto	Se desarrolla un documento en el cual se explican y detallan los puntos importantes de la problemática y del sistema a desarrollar
TAR-03	Equipo completo		Elaboración de lista de requerimientos inicial (Wishlist)	Se elabora una lista de requisitos funcionales y no funcionales, iniciales para el sistema
TAR-04	Equipo completo		Encuesta	Se crea una encuesta para poder realizar un estudio de mercado
TAR-05	Equipo completo		Estudio de mercado	Se analizan los datos de la encuesta y se realizan observaciones en cuanto al diseño de otros sistemas parecidos
TAR-06	Equipo completo	Diseño conceptual	Diagrama de clases	Se realiza un diagrama de clases, en el cual se representan todas las clases, subsistemas y relaciones que intervienen en la funcionalidad del sistema web
TAR-07	Equipo completo		Diagrama de Casos de Uso	Se realiza un diagrama de casos de uso, en el cual se representan todos

				los tipos de usuarios y las acciones que pueden llevar a cabo en el sistema web
TAR-08	Equipo completo		Diagrama navegacional de clases - Administrador	Se crea el diagrama navegacional de clases que define la navegabilidad de los objetos para el Administrador
TAR-09	Equipo completo		Diagrama navegacional de contexto - Administrador	Se crea el diagrama navegacional de contexto que define cómo se agrupan y navegan los objetos para el Administrador
TAR-10	Equipo completo	Diseño navegacional	Diagrama navegacional de clases – Ejecutivo de asistencia	Se crea el diagrama navegacional de clases que define la navegabilidad de los objetos para el Ejecutivo de asistencia
TAR-11	Equipo completo		Diagrama navegacional de contexto – Ejecutivo de asistencia	Se crea el diagrama navegacional de contexto que define cómo se agrupan y navegan los objetos para el Ejecutivo de asistencia
TAR-12	Javier Alberto Calderón Blas		Diagrama navegacional de clases - Conductor	Se crea el diagrama navegacional de clases que define la navegabilidad de los objetos para el Conductor
TAR-13	Eder Iván Negrete García		Diagrama navegacional de contexto - Conductor	Se crea el diagrama navegacional de contexto que define cómo se agrupan y navegan los objetos para el Conductor
TAR-14	Ángel Eduardo Martínez Leo Lim		Diagrama navegacional de clases – Ajustador de seguros	Se crea el diagrama navegacional de clases que define la navegabilidad de los objetos para el

				Ajustador de seguros
TAR-15	Raúl Arturo Peredo Estudillo	Diseño navegacional	Diagrama navegacional de contexto – Ajustador de seguros	Se crea el diagrama navegacional de contexto que define cómo se agrupan y navegan los objetos para el Ajustador de seguros
TAR-16	Ángel Eduardo Martínez Leo Lim	Diseño de interfaz abstracta	Interfaces para Ejecutivo de asistencia	Se crean las interfaces necesarias para representar la interacción del Ejecutivo de asistencia
TAR-17	Javier Alberto Calderón Blas		Interfaces para Conductor	Se crean las interfaces necesarias para representar la interacción del Conductor
TAR-18	Eder Iván Negrete García		Interfaces para Administrador	Se crean las interfaces necesarias para representar la interacción del Administrador
TAR-19	Raúl Arturo Peredo Estudillo		Interfaces para Ajustador de seguros	Se crean las interfaces necesarias para representar la interacción del Ajustador de seguros
TAR-20	Ángel Eduardo Martínez Leo Lim y Eder Iván Negrete García	Implementación	Codificación Administrador	Se codifica todo lo necesario para el usuario Administrador
TAR-21	Raúl Arturo Peredo Estudillo y Javier Alberto Calderón Blas		Codificación Ejecutivo asistencia	Se codifica todo lo necesario para el usuario Ejecutivo asistencia
TAR-22	Raúl Arturo Peredo Estudillo y Javier Alberto Calderón Blas		Codificación Ajustador de seguros	Se codifica todo lo necesario para el usuario Ajustador de seguros

TAR-23	Ángel Eduardo Martínez Leo Lim y Eder Iván Negrete García		Codificación Conductor	Se codifica todo lo necesario para el usuario Conductor
TAR-20	Ángel Eduardo Martínez Leo Lim y Eder Iván Negrete García	Pruebas	Pruebas Administrador	Se realizan las pruebas necesarias para Administrador
TAR-21	Raúl Arturo Peredo Estudillo y Javier Alberto Calderón Blas		Pruebas Ejecutivo asistencia	Se realizan las pruebas necesarias para Ejecutivo asistencia
TAR-22	Raúl Arturo Peredo Estudillo y Javier Alberto Calderón Blas		Pruebas Ajustador de seguros	Se realizan las pruebas necesarias para Ajustador de seguros
TAR-23	Ángel Eduardo Martínez Leo Lim y Eder Iván Negrete García	Pruebas	Pruebas Conductor	Se realizan las pruebas necesarias para Conductor

Artefactos de Diseño

Diagrama de clases

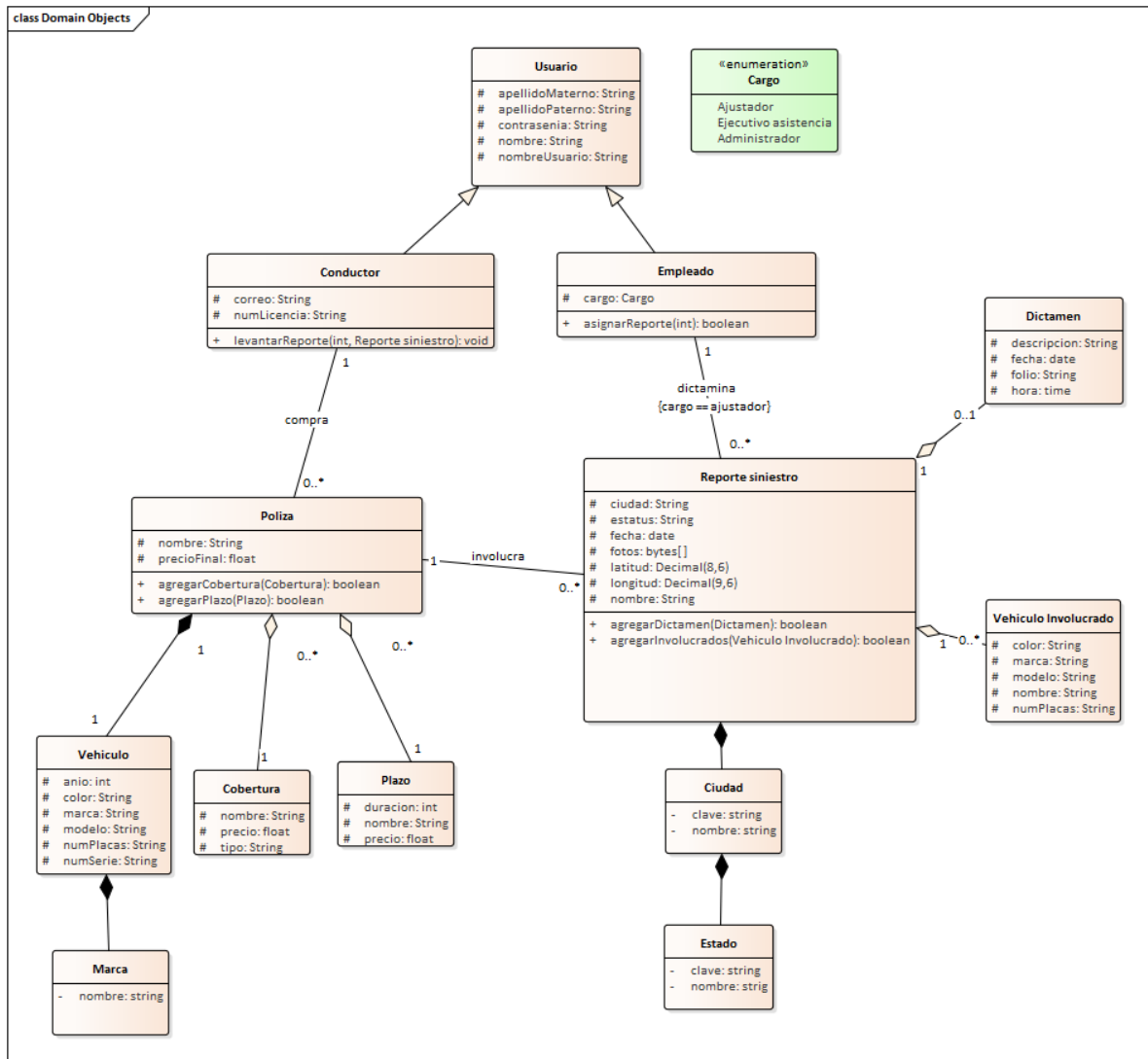


Diagrama de CU

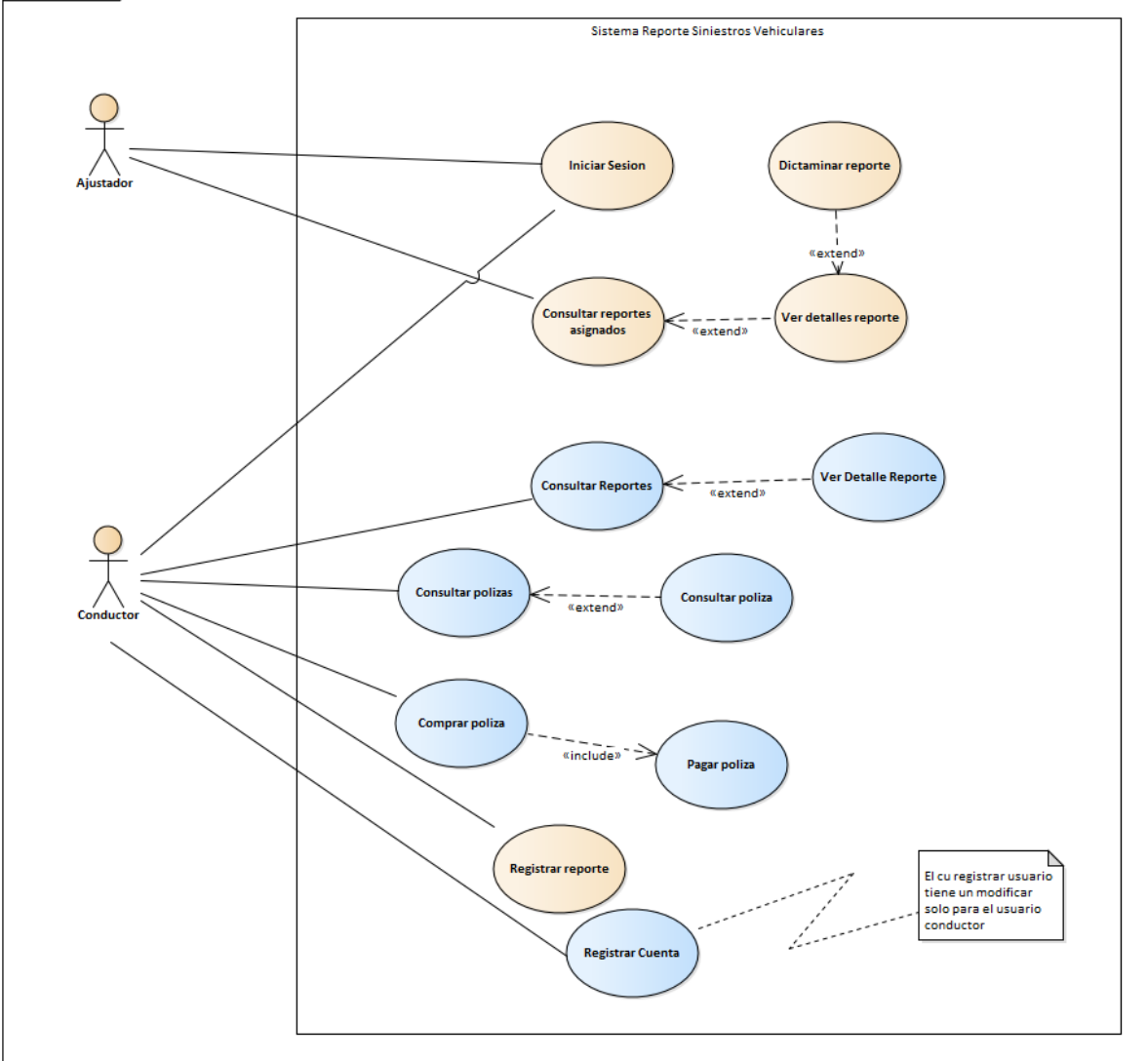


Diagrama relacional

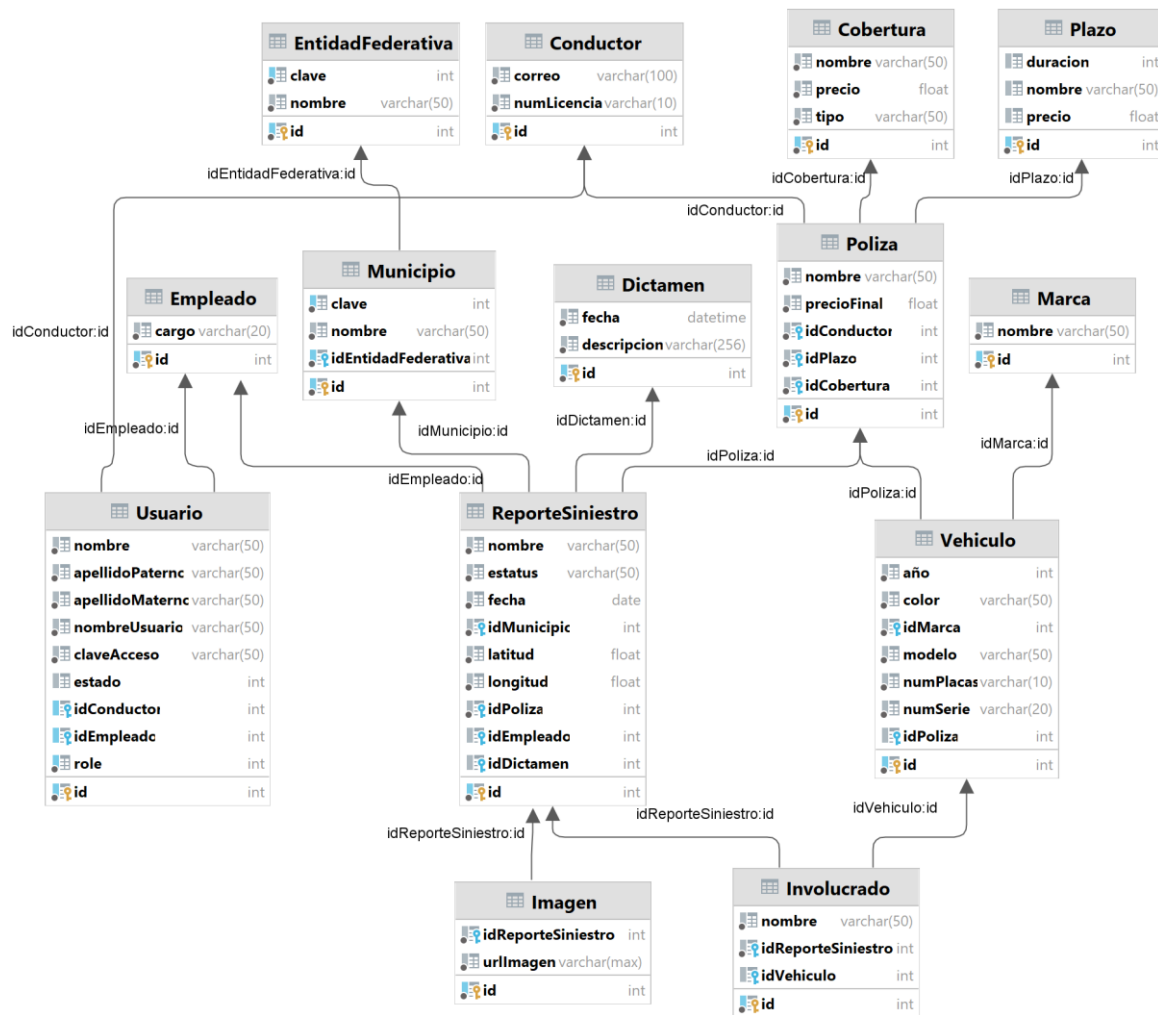


Diagrama Navegacional Conductor

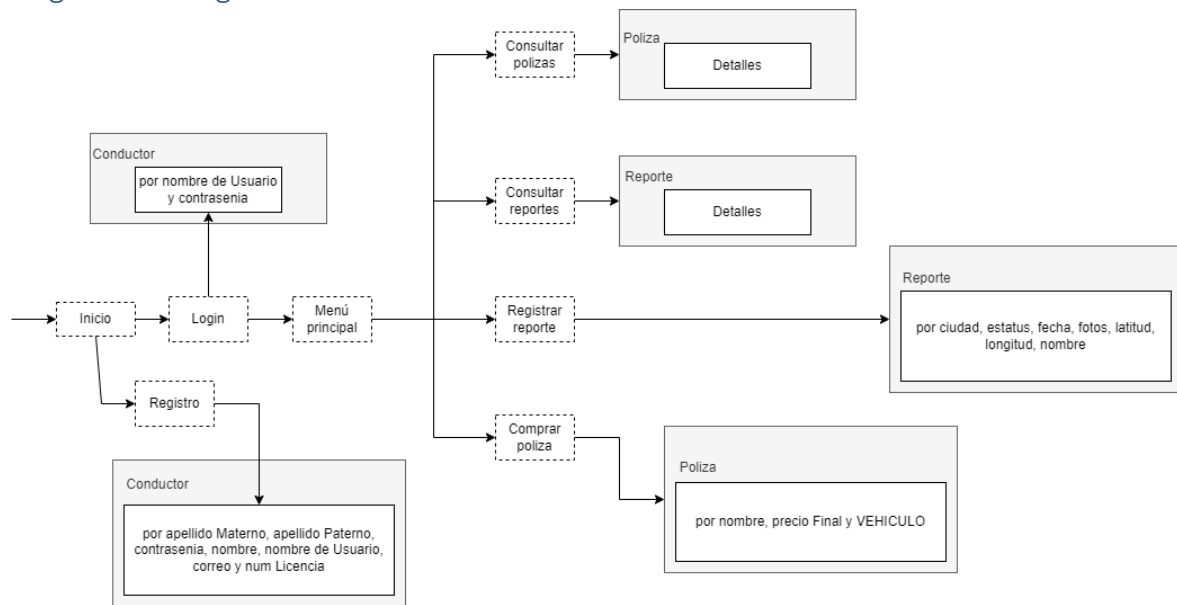


Diagrama Navegacional Ajustador

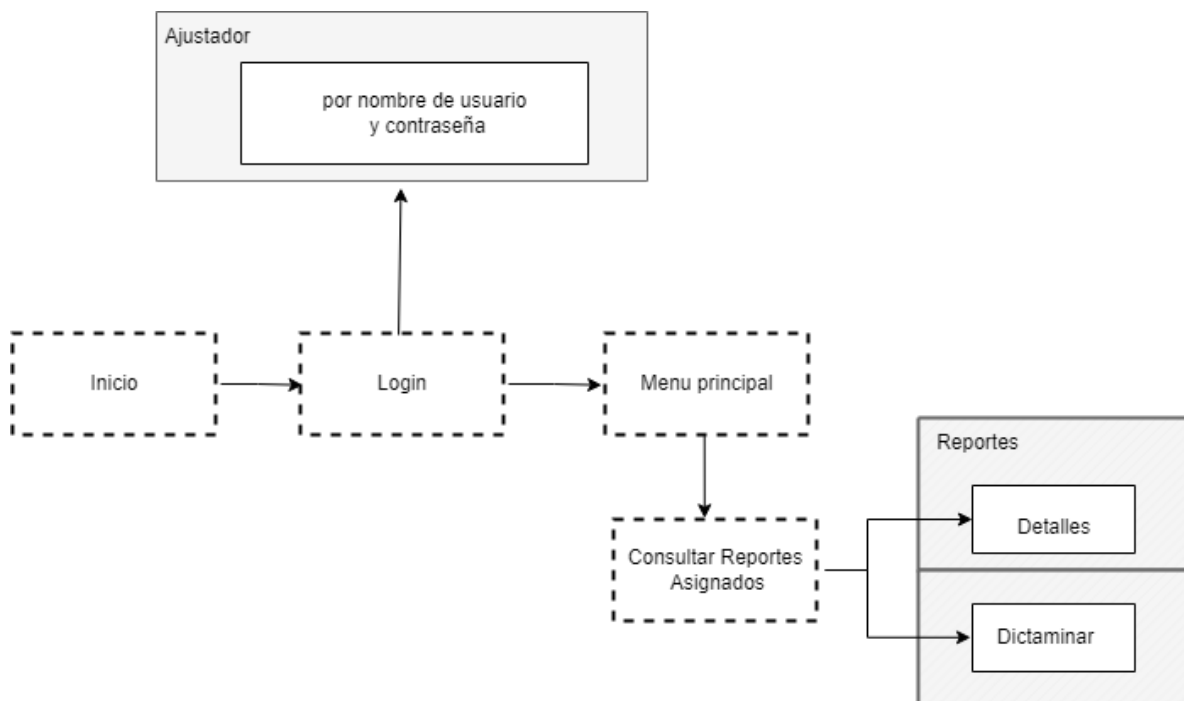


Diagrama Contexto Ajustador

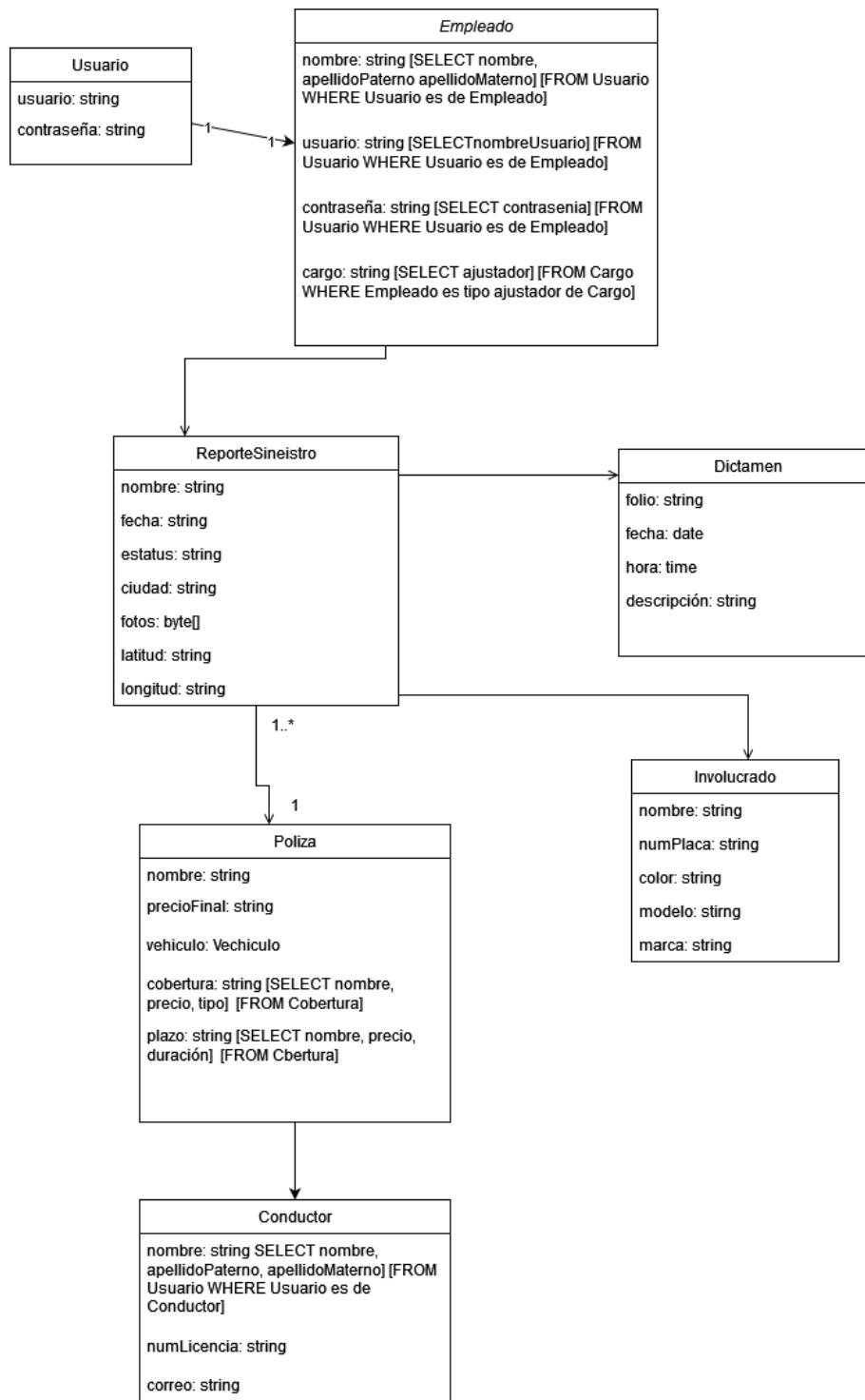
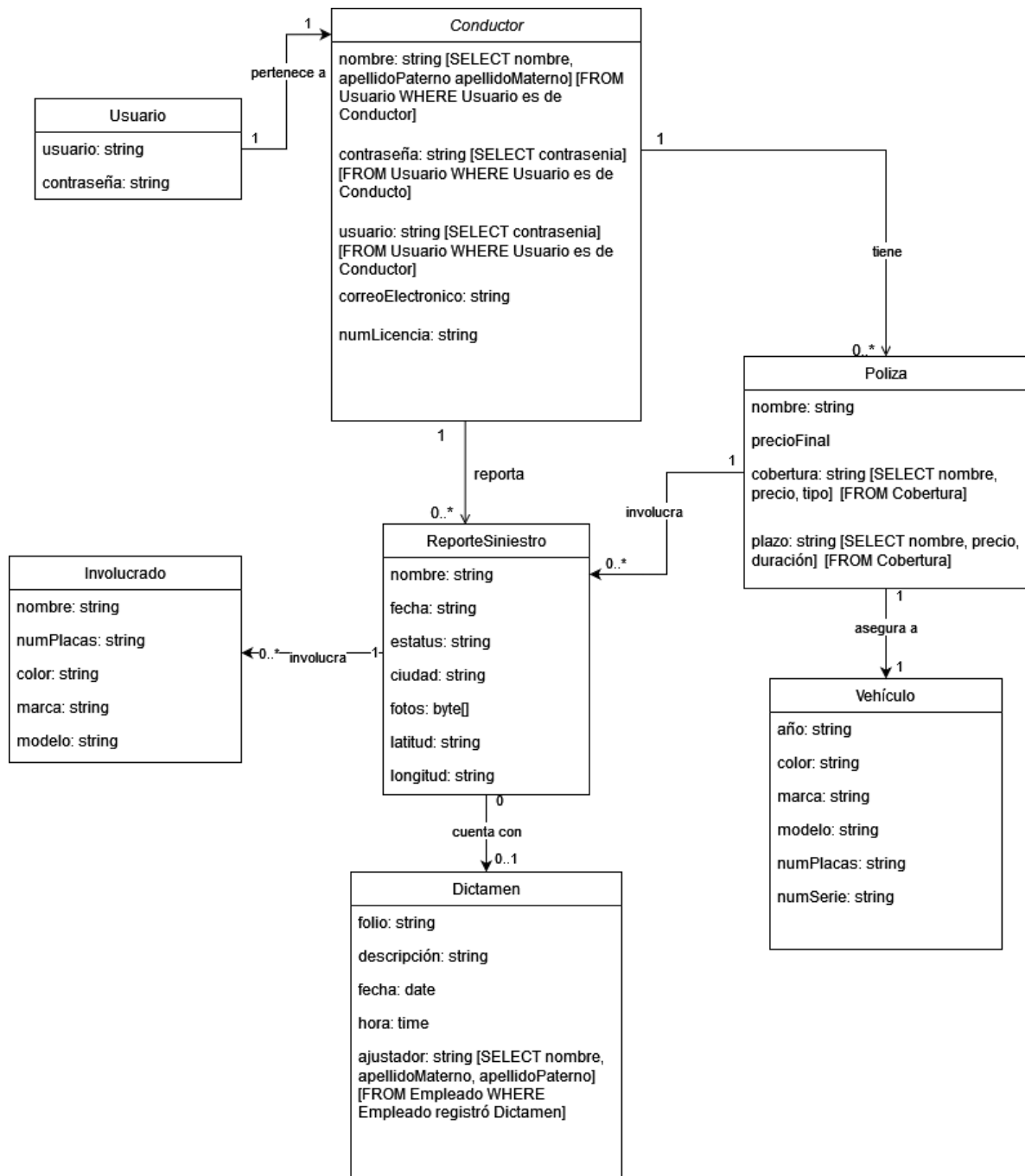


Diagrama contexto Conductor



Descripciones de CU

CU. – Iniciar sesión

Descripción de caso de uso

Nombre:	CU- Iniciar sesion
Descripción:	Permite al conductor iniciar sesión
Precondición(es):	Sin precondiciones
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none">1. El sistema muestra la ventana “Iniciar sesión” con 2 campos para número de celular (nombre de usuario) y contraseña los cuales pertenecen a USUARIO. Además, muestra un botón “Iniciar sesión” y un label para registrarse (Ver EX1)2. El conductor rellena los campos y da clic en “Iniciar sesión” (Ver FA1)3. El sistema valida las credenciales y despliega el menú (Ver FA2, FA3 y EX1)4. Termina CU
Flujos Alternos	<p>FA1. Clic en “Registrarse”</p> <ol style="list-style-type: none">1. El sistema cierra la ventana y despliega la página “Crear cuenta iCrash”. Termina CU2. Ver CU “Registrar cuenta” <p>FA2. Campos faltantes</p> <ol style="list-style-type: none">1. El sistema muestra el mensaje “Faltan campos” y un botón de “Aceptar”2. El conductor da clic en el botón “Aceptar”3. El sistema regresa al FN 1 <p>FA3. Usuario no encontrado</p> <ol style="list-style-type: none">1. El sistema muestra el mensaje “Datos incorrectos” y un botón de “Aceptar”2. El conductor da clic en el botón “Aceptar”3. El sistema regresa al FN 1
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none">1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”.2. El conductor da clic en el botón “Aceptar”

	3. El sistema cierra la ventana y termina CU.
Extiende	NA
Postcondición(es):	POST-1 Se inicia sesión correctamente

CU. – Registrar cuenta

Descripción de caso de uso

Nombre:	CU- Registrar cuenta
Descripción:	Permite a un conductor registrarse en el sistema
Precondición(es):	Sin precondiciones
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Crear cuenta iCrash” donde se muestran los campos nombre, apellido paterno, apellido materno, numero de celular (nombre de usuario) y contraseña los cuales pertenecen a USUARIO. Además, los propios de CONDUCTOR (correo electrónico y numero de licencia), por último, muestra 2 botones “Registrarse” y “Regresar” (Ver EX1) 2. El conductor rellena los campos y da clic en “Registrarse” (Ver FA1) 3. El sistema valida los campos, registra al nuevo CONDUCTOR en el sistema y muestra un mensaje “Se registro la cuenta” (Ver FA2, FA3, FA4, FA5 y EX1) 4. El conductor da clic en “Aceptar” en el mensaje 5. El sistema cierra la ventana y regresa a la ventana “Iniciar sesión” 6. Termina CU
Flujos Alternos	<p>FA1. Clic en “Regresar”</p> <ol style="list-style-type: none"> 1. El sistema cierra la ventana y despliega la página “Iniciar sesión”. Termina CU 2. Ver CU “Registrar cuenta” <p>FA2. Campos faltantes</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Faltan campos” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1

	<p>FA3. Correo no valido</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Correo no valido” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1 <p>FA4. Contraseña no segura</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Su contraseña no es segura” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1 <p>FA5. Contraseñas no coinciden</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Las contraseñas no coinciden” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1 <p>FA6. Datos ya registrados</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Datos ya registrados” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El profesor/estudiante da clic en el botón “Aceptar” 3. El sistema cierra la ventana y termina CU.
Extiende	CU – Iniciar sesión
Postcondición(es):	POST-1 Se registra con éxito la cuenta en el sistema

CU. – Registrar reporte

Descripción de caso de uso

Nombre:	CU- Registrar reporte
Descripción:	Permite a un conductor registrar un reporte de un siniestro

Precondición(es):	PRE01- Existen pólizas compradas
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Registro de reporte” donde se recuperan y muestran las siguientes secciones del reporte (Ver EX1): <ul style="list-style-type: none"> - Combobox donde se recuperan y muestran las POLIZAS que tiene el conductor - 2 campos para la latitud y longitud (REPORTE), junto con un botón “Calcular ubicación” - Botón “Elegir archivos” (imágenes de REPORTE) - Campos nombre, marca, modelo, color y numero de placas pertenecientes a INVOLUCRADO. Además, un botón “Agregar”, una tabla con los INVOLUCRADO agregados y un botón “Remover involucrado” - 2 botones “Registrar” y “Regresar” 2. El conductor selecciona la POLIZA, da clic en “Calcular ubicación” (se rellena latitud y longitud), selecciona las imágenes, agrega los datos de los INVOLUCRADO y da clic en “Registrar” (Ver FA1, FA2, FA3 y FA4) 3. El sistema valida la selección y los campos, guarda la información en el sistema y muestra el mensaje “Se registró el reporte” (Ver FA5, FA6 y EX1) 4. El conductor da clic en “Aceptar” en el mensaje 5. El sistema cierra la ventana y regresa a la ventana “Menú” (Ver EX1) 6. Termina CU
Flujos Alternos	<p>FA1. Clic en “Elegir archivos”</p> <ol style="list-style-type: none"> 1. El sistema despliega un explorador de archivos 2. El conductor busca y selecciona la imagen que desea y da clic en “Seleccionar” 3. El sistema muestra en la ventana el nombre del archivo y regresa al FN 1 <p>FA2. Clic en “Agregar”</p> <ol style="list-style-type: none"> 1. El sistema valida los campos y en caso de tener campos faltantes los rellena con “Anónimo”, por último, regresa al FN1 <p>FA3. Clic en “Remover involucrado”</p> <ol style="list-style-type: none"> 1. El sistema valida la selección y remueve al INVOLUCRADO de la lista, por último, regresa al FN 1

	<p>FA4. Clic en “Regresar”</p> <ol style="list-style-type: none"> 1. El sistema cierra la ventana y regresa a la ventana “Menú” 2. Termina CU <p>FA5. Datos faltantes</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Datos faltantes” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1 <p>FA6. Imágenes faltantes</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Debe subir un mínimo de 4 y un máximo de 8 imágenes” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El conductor da clic en el botón “Aceptar” 3. El sistema cierra la ventana y termina CU.
Extiende	NA
Postcondición(es):	POST-1 Se registra con éxito el reporte en el sistema

CU. – Modificar cuenta

Descripción de caso de uso

Nombre:	CU- Modificar cuenta
Descripción:	Permite al conductor modificar los datos de su cuenta
Precondición(es):	Sin precondiciones
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Modificar cuenta iCrash” donde se recuperan y muestran los campos nombre, apellido paterno, apellido materno, numero de celular (nombre de usuario) y contraseña los cuales pertenecen a USUARIO. Además, los propios de

	<p>CONDUCTOR (correo electrónico y numero de licencia), por último, muestra 2 botones “Modificar” y “Regresar” (Ver EX1)</p> <ol style="list-style-type: none"> 2. El conductor modifica los campos y da clic en “Modificar” (Ver FA1) 3. El sistema valida los campos, modifica los datos del CONDUCTOR en el sistema y muestra un mensaje “Se modificaron los datos” (Ver FA2, FA3, FA4, FA5 y EX1) 4. El conductor da clic en “Aceptar” en el mensaje 5. El sistema cierra la ventana y regresa a la ventana “Menú” 6. Termina CU
Flujos Alternos	<p>FA1. Clic en Regresar</p> <ol style="list-style-type: none"> 1. El sistema cierra la ventana y regresa a la anterior 2. Termina CU <p>FA2. Campos incompletos</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Campos incompletos” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1 <p>FA3. Campos inválidos</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Campos inválidos” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El conductor da clic en el botón “Aceptar” 3. El sistema cierra la ventana y termina CU.
Extiende	
Postcondición(es):	POST-1 Se modifican los datos del conductor con éxito en el sistema

CU. – Comprar póliza

Descripción de caso de uso

Nombre:	CU- Comprar póliza
Descripción:	Permite al conductor comprar una póliza
Precondición(es):	Sin precondiciones
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none">1. El sistema muestra la ventana “Comprar póliza” donde se recuperan y muestran las siguientes secciones de la póliza (Ver EX1):<ul style="list-style-type: none">- 2 campos para el nombre y precio final de POLIZA- Combobox donde se recuperan nombre, precio y tipo pertenecientes a COBERTURA- Combobox donde se recuperan duración, nombre y precio pertenecientes a PLAZO- Campos para año, color, marca, modelo, numero placas y numero serie pertenecientes a VEHÍCULO- 2 botones “Comprar” y “Regresar”2. El conductor selecciona y rellena los campos y da clic en “Comprar” (Ver FA1)3. El sistema valida los campos y las selecciones, por último, despliega la ventana emergente “Pagar póliza” (Ver FA2, FA3 y EX1)4. Ver CU “Pagar póliza”.
Flujos Alternos	<p>FA1. Clic en Regresar</p> <ol style="list-style-type: none">1. El sistema cierra la ventana y regresa a la anterior2. Termina CU <p>FA2. Campos incompletos</p> <ol style="list-style-type: none">1. El sistema muestra el mensaje “Campos incompletos” y un botón de “Aceptar”2. El conductor da clic en el botón “Aceptar”3. El sistema regresa al FN 1 <p>FA3. Datos ya registrados</p> <ol style="list-style-type: none">1. El sistema muestra el mensaje “Datos ya registrados” y un botón de “Aceptar”2. El conductor da clic en el botón “Aceptar”

	3. El sistema regresa al FN 1
Excepción(es):	EX1. Error en la conexión de base de datos. 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El conductor da clic en el botón “Aceptar” 3. El sistema cierra la ventana y termina CU.
Incluye	CU. Pagar póliza
Postcondición(es):	POST-1 Se validan los datos para la compra de la póliza con éxito

CU. – Pagar póliza

Descripción de caso de uso

Nombre:	CU- Pagar póliza
Descripción:	Permite al conductor pagar la póliza
Precondición(es):	PRE01 – Se tienen los datos correctos de la póliza a pagar
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Pagar póliza” donde se muestran los campos numero de tarjeta, nombre, fecha de expiración y CCV, por último, muestra 2 botones “Pagar” y “Regresar” (Ver EX1) 2. El conductor selecciona y rellena los campos y da clic en “Pagar” (Ver FA1) 3. El sistema valida los campos y las selecciones, por último, despliega el mensaje “Se ha comprado la póliza con éxito” junto con un botón de aceptar (Ver FA2 y EX1) 4. El conductor da clic en “Aceptar” 5. El sistema regresa a la ventana “Menú” (Ver EX1) 6. Termina CU
Flujos Alternos	<p>FA1. Clic en Regresar</p> <ol style="list-style-type: none"> 1. El sistema cierra la ventana y regresa a la anterior 2. Termina CU <p>FA2. Campos incompletos</p>

	<ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Campos incompletos” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El conductor da clic en el botón “Aceptar” 3. El sistema cierra la ventana y termina CU.
Incluye	
Postcondición(es):	POST-1 Se guarda con éxito la información de la póliza

CU. – Consultar pólizas

Descripción de caso de uso

Nombre:	CU- Consultar pólizas
Descripción:	Permite al conductor consultar todas las pólizas que tiene
Precondición(es):	PRE01- Existen pólizas registradas en el sistema
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Consultar pólizas” donde se recuperan y muestran en una tabla las POLIZA (nombre y precio final) con su VEHICULO (modelo y placas), por último, muestra 2 botones “Ver detalles póliza” y “Regresar” (Ver EX1) 2. El conductor selecciona la póliza y da clic en “Ver detalles póliza” (Ver FA1) 3. El sistema valida la selección y despliega la ventana “Consultar póliza” (Ver FA2 y EX1) 4. Ver CU “Consultar póliza”
Flujos Alternos	<p>FA1. Clic en Regresar</p> <ol style="list-style-type: none"> 1. El sistema cierra la ventana y regresa a la anterior 2. Termina CU <p>FA2. Selección faltante</p>

	<ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Seleccione una póliza” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El conductor da clic en el botón “Aceptar” 3. El sistema cierra la ventana y termina CU.
Extiende	
Postcondición(es):	POST-1 Se valida la selección de la póliza y se despliega la siguiente ventana

CU. – Consultar póliza

Descripción de caso de uso

Nombre:	CU- Consultar póliza
Descripción:	Permite al conductor consultar todos los detalles de una póliza
Precondición(es):	PRE01- Existen pólizas registradas en el sistema
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Consultar póliza” donde se recuperan y muestran las siguientes secciones de la póliza (Ver EX1): <ul style="list-style-type: none"> - Campos para nombre póliza y precio final (POLIZA), nombre cobertura, precio cobertura y tipo cobertura (COBERTURA), y duración plazo, nombre plazo y precio plazo (PLAZO) - Campos para numero placas, numero serie, marca vehículo, modelo vehículo, año y color (VEHICULO) - Botón “Regresar” 2. El conductor visualiza los datos, por último, da clic en “Regresar” 3. El sistema regresa a la ventana “Consultar pólizas” (Ver EX1) 4. Termina CU
Flujos Alternos	

Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El conductor da clic en el botón “Aceptar” 3. El sistema cierra la ventana y termina CU.
Extiende	CU. Consultar pólizas
Postcondición(es):	POST-1 Se visualiza con éxito la información de la póliza

CU. – Consultar reportes

Descripción de caso de uso

Nombre:	CU- Consultar reportes
Descripción:	Permite al conductor consultar todos los reportes que ha realizado
Precondición(es):	PRE01- Existen reportes registrados en el sistema
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Consultar reportes” donde se recuperan y muestran en una tabla el nombre (POLIZA), nombre, ciudad, latitud, longitud y estatus (REPORTE SINIESTRO), por último, muestra 2 botones “Ver detalles póliza” y “Regresar” (Ver EX1) 2. El conductor selecciona el reporte y da clic en “Ver detalles reporte” (Ver FA1) 3. El sistema valida la selección y despliega la ventana “Detalles reporte siniestro” (Ver FA2 y EX1) 4. Ver CU “Ver detalles reporte”
Flujos Alternos	<p>FA1. Clic en Regresar</p> <ol style="list-style-type: none"> 1. El sistema cierra la ventana y regresa a la anterior 2. Termina CU <p>FA2. Selección faltante</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Seleccione un reporte” y un botón de “Aceptar” 2. El conductor da clic en el botón “Aceptar” 3. El sistema regresa al FN 1

Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 4. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 5. El conductor da clic en el botón “Aceptar” 6. El sistema cierra la ventana y termina CU.
Extiende	
Postcondición(es):	POST-1 Se valida la selección del reporte y se despliega la siguiente ventana

CU. – Ver detalles reporte

Descripción de caso de uso

Nombre:	CU- Ver detalles reporte
Descripción:	Permite al conductor consultar todos los detalles de una póliza
Precondición(es):	PRE01- Existen pólizas registradas en el sistema
Actor(es):	Conductor
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Detalles reporte siniestro” donde se recuperan y muestran las siguientes secciones de la póliza (Ver EX1): <ul style="list-style-type: none"> - Campos para nombre (POLIZA), nombre, ciudad, latitud, longitud y estatus (REPORTE SINIESTRO) - Campos para número placas, numero serie, marca vehículo, modelo vehículo, año y color (VEHICULO) - Imágenes del REPORTE SINIESTRO - Tabla con nombre, marca, modelo, color y numero placas de los INVOLUCRADO - Campos para nombre, apellido paterno y apellido materno de USUARIO (Ajustador) - Tabla con el folio, fecha y descripción del DICTAMEN (En caso de que hubiera) - Botón “Regresar” 2. El conductor visualiza los datos, por último, da clic en “Regresar” 3. El sistema regresa a la ventana “Consultar reportes” (Ver EX1) 4. Termina CU

Flujos Alternos	
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El conductor da clic en el botón “Aceptar” 3. El sistema cierra la ventana y termina CU.
Extiende	CU. Consultar reportes
Postcondición(es):	POST-1 Se visualiza con éxito la información del reporte de siniestro

CU.- Iniciar Sesión Ajustador

Descripción de caso de uso

Nombre:	CU- Iniciar sesión Ajustador
Descripción:	Permite al ajustador iniciar sesión en el sistema
Precondición(es):	Sin precondiciones
Actor(es):	Ajustador
Flujos normales	<ol style="list-style-type: none"> 5. El sistema muestra la ventana “Iniciar sesión” con 2 campos para número de celular (nombre de usuario) y contraseña los cuales pertenecen a USUARIO, muestra un botón “Iniciar sesión” 6. El ajustador rellena los campos y da clic en “Iniciar sesión” 7. El sistema valida las credenciales y despliega el menú (Ver FA1, FA2 y EX1) 8. Termina CU
Flujos Alternos	<p>FA1. Campos faltantes</p> <ol style="list-style-type: none"> 4. El sistema muestra el mensaje “Faltan campos” y un botón de “Aceptar” 5. El ajustador da clic en el botón “Aceptar” 6. El sistema regresa al FN 1 <p>FA2. Usuario no encontrado</p>

	<ol style="list-style-type: none"> 4. El sistema muestra el mensaje "Datos incorrectos" y un botón de "Aceptar" 5. El ajustador da clic en el botón "Aceptar" 6. El sistema regresa al FN 1
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 4. El sistema muestra el mensaje "Error, no se pudo conectar con la base de datos" y el botón "Aceptar". 5. El ajustador da clic en el botón "Aceptar" 6. El sistema cierra la ventana y termina CU.
Extiende	NA
Postcondición(es):	POST-1 Se inicia sesión correctamente

CU.- Consultar reportes asignados

Descripción de caso de uso

Nombre:	CU- Consultar reportes
Descripción:	Permite al ajustador consultar los reportes que están asignados a su usuario.
Precondición(es):	PRE1. Estar logueado en el sistema
Actor(es):	Ajustador
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana "Menú Principal" con los botones "consultar reportes" y "salir". 2. El ajustador da clic en el botón "consultar reportes". 3. El sistema muestra la ventana "Consultar Reportes" con una tabla con los reportes que tiene asignados, y tres botones "ver detalles reporte", Dictaminar Reporte" y "Salir". Ver EX1 4. Termina CU.
Flujos Alternos	
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje "Error, no se pudo conectar con la base de datos" y el botón "Aceptar". 2. El ajustador da clic en el botón "Aceptar". 3. El sistema cierra la ventana y termina CU.

Extiende	CU. Dictaminar Reporte CU. Ver Detalles Reporte
Postcondición(es):	

CU.- Dictaminar Reporte

Descripción de caso de uso

Nombre:	CU- Dictaminar Reporte
Descripción:	Permite al ajustador dictaminar un accidente en específico.
Precondición(es):	PRE1. Estar logueado en el sistema PRE2. Haber seleccionado un reporte con anterioridad.
Actor(es):	Ajustador
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana “Dictaminar Reporte” con atributo de DICTAMEN (descripción, fecha, folio y hora) así como un imput para cada atributo y un botón “Dictaminar”. 2. El ajustador ingresa la información en cada uno de los inputs y da clic en “Dictaminar”. 3. El sistema valida la información y guarda el dictamen en la base de datos del sistema. (Ver FA1, FA2 y EX1). 4. Termina CU.
Flujos Alternos	<p>FA1. Campos faltantes</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Faltan campos” y un botón de “Aceptar” 2. El ajustador da clic en el botón “Aceptar” 3. El sistema regresa al FN 1 <p>FA2. Formato invalido</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Formato invalido para este campo” y un botón de “Aceptar” 2. El ajustador da clic en el botón “Aceptar” 3. El sistema regresa al FN 1
Excepción(es):	<p>EX1. Error en la conexión de base de datos.</p> <ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Error, no se pudo conectar con la base de datos” y el botón “Aceptar”. 2. El ajustador da clic en el botón “Aceptar”. 3. El sistema cierra la ventana y termina CU.

Extiende	NA
Postcondición(es):	EL dictamen queda almacenado en la base de datos.

CU.- Ver Detalles Reporte

Descripción de caso de uso

Nombre:	CU- Ver detalles Reporte.
Descripción:	Permite al ajustador ver los detalles de un reporte en específico.
Precondición(es):	PRE1. Estar logueado en el sistema PRE2. Haber seleccionado un reporte con anterioridad.
Actor(es):	Ajustador
Flujos normales	<ol style="list-style-type: none"> 1. El sistema muestra la ventana "Detalles Reporte" con los datos de REPORTE (ciudad, estatus, fecha, fotos, latitud, longitud, nombre) con la información del reporte seleccionado, y un botón de "Volver." (Ver EX1) 2. El ajustador consulta la información y da clic en volver. 3. Termina CU.
Flujos Alternos	
Excepción(es):	EX1. Error en la conexión de base de datos. <ol style="list-style-type: none"> 4. El sistema muestra el mensaje "Error, no se pudo conectar con la base de datos" y el botón "Aceptar". 5. El ajustador da clic en el botón "Aceptar". 6. El sistema cierra la ventana y termina CU.
Extiende	NA
Postcondición(es):	EL dictamen queda almacenado en la base de datos.

Pruebas

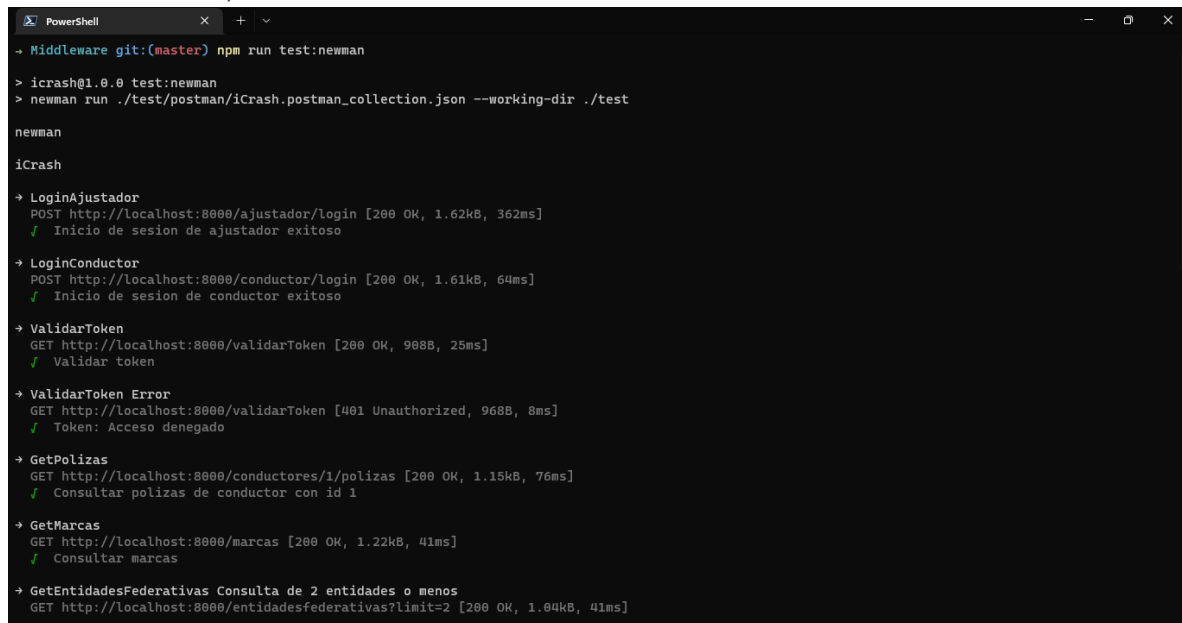
Se realizaron pruebas tanto con Postman, como con Mocha.

Postman: Postman es una plataforma gratuita si trabajas solo o de pago si quieres trabajar de manera colaborativa con tu equipo. Esta plataforma posibilita y facilita la creación y el uso de APIs. Se puede usar, por ejemplo, para obtener información sobre las respuestas HTTP, en diferentes métodos, que realicemos a APIs de diferentes temáticas.

Mocha: A mocha se le puede definir como un framework de pruebas rico en características que puede ser implementado en nodejs o por medio de un navegador, mocha ejecuta las pruebas en serie permitiendo reportes flexibles y exactos, haciendo que el código pasado pruebas sea óptimo al momento de estar en producción. Pero se puede resumir la definición de mocha en tres palabras “simple”, “flexible” y “divertido”

Postman

Resultado de las pruebas



```
PowerShell
Middleware git:(master) npm run test:newman
> icrash@1.0.0 test:newman
> newman run ./test/postman/iCrash.postman_collection.json --working-dir ./test

newman
iCrash

→ LoginAjustador
POST http://localhost:8000/ajustador/login [200 OK, 1.62kB, 362ms]
✓ Inicio de sesion de ajustador exitoso

→ LoginConductor
POST http://localhost:8000/conductor/login [200 OK, 1.61kB, 64ms]
✓ Inicio de sesion de conductor exitoso

→ ValidarToken
GET http://localhost:8000/validarToken [200 OK, 908B, 25ms]
✓ Validar token

→ ValidarToken Error
GET http://localhost:8000/validarToken [401 Unauthorized, 968B, 8ms]
✓ Token: Acceso denegado

→ GetPolizas
GET http://localhost:8000/conductores/1/polizas [200 OK, 1.15kB, 76ms]
✓ Consultar polizas de conductor con id 1

→ GetMarcas
GET http://localhost:8000/marcas [200 OK, 1.22kB, 41ms]
✓ Consultar marcas

→ GetEntidadesFederativas Consulta de 2 entidades o menos
GET http://localhost:8000/entidadesfederativas?limit=2 [200 OK, 1.04kB, 41ms]
```



```
PowerShell
+ GetEntidadesFederativas Consulta de 2 entidades o menos
GET http://localhost:8000/entidadesfederativas?limit=2 [200 OK, 1.04kB, 41ms]
✓ Consulta de 2 entidades federativas o menos

+ GetEntidadesFederativas
GET http://localhost:8000/entidadesfederativas [200 OK, 2.35kB, 43ms]
✓ Consultar entidades federativas

+ GetMunicipios
GET http://localhost:8000/entidadesfederativas/1/municipios [200 OK, 951B, 58ms]
✓ Consultar municipios

+ GetReporteNoEncontrado
GET http://localhost:8000/reportesSiniestro/1 [404 Not Found, 1.02kB, 51ms]
✓ Reporte no encontrado

+ GetReportesAjustador
GET http://localhost:8000/ajustadores/1/reportesSiniestro [200 OK, 2.43kB, 50ms]
✓ Consultar reportes de ajustador con id

+ GetReporte
GET http://localhost:8000/reportesSiniestro/2 [200 OK, 1.62kB, 101ms]
✓ Consultar detalles de reporte con id 2

+ GetImagen
GET http://localhost:8000/reportesSiniestro/imagenes/2-1.jpg [200 OK, 880.25kB, 21ms]
✓ Consultar imagen

+ PostReporteSineistro
POST http://localhost:8000/conductores/reportesSiniestro [201 Created, 981B, 145ms]
✓ Registrar reporte

+ PostDictaminar
POST http://localhost:8000/ajustador/10/dictaminar [400 Bad Request, 1.02kB, 38ms]

PowerShell
+ GetReporte
GET http://localhost:8000/reportesSiniestro/2 [200 OK, 1.62kB, 101ms]
✓ Consultar detalles de reporte con id 2

+ GetImagen
GET http://localhost:8000/reportesSiniestro/imagenes/2-1.jpg [200 OK, 880.25kB, 21ms]
✓ Consultar imagen

+ PostReporteSineistro
POST http://localhost:8000/conductores/reportesSiniestro [201 Created, 981B, 145ms]
✓ Registrar reporte

+ PostDictaminar
POST http://localhost:8000/ajustador/10/dictaminar [400 Bad Request, 1.02kB, 38ms]



|                                                                | executed | failed |
|----------------------------------------------------------------|----------|--------|
| iterations                                                     | 1        | 0      |
| requests                                                       | 15       | 0      |
| test-scripts                                                   | 29       | 0      |
| prerequisite-scripts                                           | 20       | 0      |
| assertions                                                     | 14       | 0      |
| total run duration: 2.9s                                       |          |        |
| total data received: 883.86kB (approx)                         |          |        |
| average response time: 74ms [min: 8ms, max: 362ms, s.d.: 83ms] |          |        |


+ Middleware git:(master) |
```

Mocha

Código de las pruebas

```
//TESTS
describe("Reportes siniestros", function() {
  describe('#Consultar reportes', function () {
    it('Debe retornar un arreglo con los reportes asignados para el idEmpleado 1 ', function() {
      return new Promise((resolve, reject) => {
        consultarReportesSiniestroAjustador(1)
          .then((reportes) => {
            assert.ok(Array.isArray(reportes), "Es un arreglo")
            resolve(reportes)
          })
          .catch(error => {
            reject(error)
          })
      })
    });
  });
});
```

```
//TESTS
describe("Dictámenes", function() {
  describe('#Registrar dictamen', function () {
    it('Debe un 400, ya que el reporte ya se encuentra dictaminado ', function() {
      return new Promise((resolve, reject) => {
        var date = Date.now()
        registrarDictamenReporteController("Se realizo el dictamen correspondiente", date, 2)
          .then((dictamenStatus) => {
            assert.equal(dictamenStatus, 400)
            reject()
          })
          .catch(error => {
            resolve(error)
          })
      })
    });
  });
});
```

```
// TESTS
describe("Usuarios", function () {
  describe('#Consultar Usuarios', function () {
    it('Debe retornar un arreglo', function () {
      return new Promise((resolve, reject) => {
        consultarUsuarios()
          .then((usuarios) => {
            assert.ok(Array.isArray(usuarios), "No es un arreglo")
            resolve(usuarios)
          })
          .catch(error => {
            reject(error)
          })
      })
    });

    it("Debe retornar un usuario con id 2", function () {
      return new Promise((resolve, reject) => {
        consultarUsuarioPorId(2)
          .then((usuario) => {
            assert.equal(usuario.id, 2)
            resolve()
          })
          .catch(error => {
            reject(error)
          })
      })
    })
  })
})
```

```

describe('#Autenticación', function () {
  it('Inicio de sesion con usuario 2282144903 y contrasena 123', function () {
    return new Promise((resolve, reject) => {
      autentificacionConductor("2282144903", "123")
        .then((response) => {
          assert.equal(response.nombreUsuario, "2282144903");
          assert.equal(response.claveAcceso, "123");
          resolve();
        })
        .catch(error => {
          reject(error);
        })
    });
  });
});

describe('#Autenticación', function () {
  it('Inicio de sesion con usuario 2282144903 y contrasena 123', function () {
    return new Promise((resolve, reject) => {
      autentificacionAjustador("2282144903", "123")
        .then((response) => {
          assert.equal(response.nombreUsuario, "2282144903");
          assert.equal(response.claveAcceso, "123");
          resolve();
        })
        .catch(error => {
          reject(error);
        })
    });
  });
});

```

Resultados de pruebas

```

9 filas
  ✓ Debe retornar un arreglo (616ms)
1 filas
  ✓ Debe retornar un usuario con id 2 (616ms)
  #Autenticación
1 filas
  ✓ Inicio de sesion con usuario 2282144903 y contrasena 123 (616ms)
  #Autenticación
1 filas
  ✓ Inicio de sesion con usuario 2282144903 y contrasena 123 (618ms)

6 passing (4s)

```

Conclusión

Para el desarrollo de software de cualquier tipo es necesario tener un amplio conocimiento no solo de la tecnología o de las herramientas que se usaran sino de la problemática que el cliente esta presentando, es por eso que en esta materia se supo en práctica todas las técnicas de desarrollo que se han aprendido durante lo que llevamos de carrera.

Algo importante a desatacar de este desarrollo fue que utilizamos tecnología web, esto fue algo importante ya que hoy en día la mayoría de las empresas opta por un desarrollo de este estilo por su bajo costo en comparación con las aplicaciones de escritorio, también por la forma en la que se accede a este tipo de sistemas, ya que es mas sencillo acceder desde un navegador que estar instalando aplicaciones en cada dispositivo en el que necesites hacer uso de la aplicación.

Por último, la metodología que usamos nos ayudo mucho ya que consideramos que fue la que mejor se adaptaba por la forma en la que trabajaba.