

**Desarrollo de un aplicativo móvil partner-car que sirva como herramienta de comunicación  
para los conductores mediante el uso de reconocimiento de voz**

Jorge Luis Montoya Parada

Universitaria Agustiniana  
Facultad de Ingenierías  
Programa de Tecnología en Desarrollo de Software  
Bogotá D.C.  
2020

**Desarrollo de un aplicativo móvil partner-car que sirva como herramienta de comunicación  
para los conductores mediante el uso de reconocimiento de voz**

Jorge Luis Montoya Parada

Director

Mauricio Alonso Villalba

Trabajo de grado para optar al título de Tecnólogo en Desarrollo de Software

Universitaria Agustiniana

Facultad de Ingenierías

Programa de Tecnología en Desarrollo de Software

Bogotá D.C.

2020

## **Resumen**

El presente documento respeta la estructura recomendada por el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830- 1998”. Las secciones omitidas se consideran no aplicables.

La única desviación frente al estándar se produce en la sección 3, donde los requisitos se describen en formato de lista, reduciendo por ello complejidad de las plantillas recomendadas por el estándar para dicha sección.

La sección 3 enumera ciertos requisitos que no pueden calificarse como tales, ya que no describen servicios que debe proporcionar el software, sino aspectos procedimentales que deben ser ignorados. Dichos requisitos son: Req (25) y Req (34). Se incluyen tales requisitos para lograr una mayor claridad en la exposición.

## Tabla de contenidos

Introducción .....	6
1. Planeación del proyecto .....	7
1.1 Objetivos del proyecto.....	7
1.1.1 Objetivo general.....	7
1.1.2 Objetivos específicos. ....	7
1.2 Planteamiento del problema y/o necesidad .....	8
1.3 Alcance del proyecto .....	8
1.4 Metodología de desarrollo de software .....	9
1.5 Definiciones, acrónimos y abreviaturas.....	15
2.Marco teórico y estado del arte .....	17
3. Especificación de requisitos de software (ieee 830) .....	24
3.1 Perspectiva del producto.....	24
3.2 Funcionalidad del producto .....	25
3.3 Características de los Usuarios .....	25
3.4 Restricciones.....	25
3.5 Suposiciones y Dependencias.....	26
3.6 Requisitos Específicos .....	26
3.6.1 Actores y roles: .....	26
3.6.2 Requisitos Funcionales .....	26
3.6.3 Diagrama de casos de uso: .....	28
3.6.4 Especificaciones de los casos de uso .....	29
3.7 Requisitos de Rendimiento .....	29
3.7.1 Requisitos tecnológicos .....	29
3.8 Restricciones de Diseño .....	30
3.9 Atributos del Software del Sistema.....	30
4. Diseño del software (iso-12207-1) .....	31
4.1 Diseño de la arquitectura del Software.....	31
4.2.1 Diagrama de clases .....	33
4.2.2 Diagrama de paquetes .....	33
4.2.3 Diagrama de despliegue .....	34

4.3 Diseño de interfaz .....	35
4.3.1. Interfaz gráfica del usuario.....	35
4.3.2 Interfaces de entrada y salida .....	37
5. Implementación.....	38
5.1 Plataformas de desarrollo .....	38
5.2 Base de datos .....	38
5.3. Infraestructura de Hardware y redes.....	39
6. Pruebas del Software. ....	40
6.1 Pruebas del Software. ....	40
6.2 Pruebas de usabilidad. ....	42
Conclusiones y recomendaciones.....	45
Referencias .....	46

## Introducción

El potencial con el que cuentan actualmente los teléfonos celulares ha llevado a crear una necesidad de estar constantemente al pendiente de ellos ya sea por trabajo, personal o entretenimientos. Estos dispositivos permiten tener tanta información a la mano en cuestión de segundos que para muchas personas parece ser un reto mantenerse alejado de las pantallas mientras manejan. Las distracciones más comunes incluyen hablar, enviar mensajes de texto, navegar en internet y seleccionar música.

Tomarse selfies, llamar y mensajear en celulares cuando se maneja, aumenta cuatro veces la probabilidad de sufrir accidentes, de acuerdo con la Organización Mundial de la Salud (OMS). A través de una aplicación que controla diversos parámetros en el comportamiento de los conductores, la firma estadounidense *Zen Drive* analizó los datos anónimos de 1,8 millones de usuarios que recorrieron más de 7.000 millones de kilómetros en tres meses, y los completó con una encuesta sobre 500 conductores. *Zen Drive* aseguró que los conductores triplicaron el tiempo en el que usan sus teléfonos al volante y determinó que los adictos al móvil son un peligro público mayor que los conductores ebrios ya que ignoran la ruta el 28% del tiempo que están manejando y están en la carretera casi el doble del tiempo que la población general.

Terminología

Nomofobia

## **1. Planeación del proyecto**

Actualmente existen tecnologías capaces de entrar en un “diálogo” con personas que les indican mediante algunos comandos de voz ciertas actividades específicas que son capaces de llevar a cabo sin la intervención alguna del usuario. Siguiendo una de las tendencias más marcadas actualmente en el mundo tecnológico, la interacción humano-máquina mediante el uso de la voz, se propone la integración de un sistema de infoentretenimiento para el auto capaz de entender el lenguaje natural del usuario mediante comandos de voz cotidianos, proporcionando así un sistema que brinde más seguridad y comodidad a las personas mejorando la experiencia de manejo del vehículo, con el único fin de reducir las distracciones a las personas y a su vez disminuir las cifras de accidentalidad.

Gracias a los avances tecnológicos que cada día surgen en el mundo, este tipo de sistemas pueden ser muy eficientes e indispensables a la hora de adquirir un vehículo que proporcione las medidas de seguridad suficientes para los usuarios, las cuales se deben cumplir con los estándares o normas establecidas por los entes que regulan dichas normas. Como se trata de un sistema el cual permita la interacción directa con la máquina, no tendrá que preocuparse por mirar el teléfono, simplemente lo podrá dejar a un lado, brindando mucha comodidad al usuario ya que no existirá ningún tipo de cableado que pueda estorbar a la hora de conducir. Este tipo de innovaciones puede ser muy novedosa y atractiva para los usuarios, no solo observando como un simple sistema de interacción sino como una medida de seguridad para los usuarios.

### **1.1 Objetivos del proyecto**

#### **1.1.1 Objetivo general.**

Generar un aplicativo móvil que sirva como herramienta de comunicación para los conductores, utilizando comandos de voz.

#### **1.1.2 Objetivos específicos.**

Utilizar un reconocimiento de voz que reconozca la voz del usuario y que se ejecute la función que se expresa.

Entrenar el sistema con comandos específicos que el usuario pueda utilizar y el sistema los ejecute correctamente.

Reconocer la voz del usuario que desea enviar un mensaje por la aplicación de WhatsApp, posteriormente que el sistema lo convierta a texto plano y envíe dicho mensaje.

Cuando llegue un mensaje al usuario, el sistema activará el altavoz y leerá el mensaje para el usuario.

El sistema maneja autenticación para cada usuario, permitiendo así que el sistema sea portable y el usuario pueda usarlo en cualquier automóvil.

## **1.2 Planteamiento del problema y/o necesidad**

Se estima que alrededor de un millón trescientos cincuenta mil personas, de entre 19 y 25 años de edad, al año fallecen en accidentes automovilísticos, se estima que alrededor del 50% de estos accidentes son o están relacionados con el uso del teléfono celular mientras manejan.

Hoy en día se observa que hay una fuerte problemática con el uso de los celulares, pues así como estos nos benefician en la comunicación y la información en tiempo real, los seres humanos todavía no hemos comprendido cuando es el debido uso de estos, por ello nos ha llevado a que se cometan errores que afectan contra la vida, es por ello que es urgente implementar una herramienta que no nos desconecte de nuestra vida normal, pero que tampoco se corra un riesgo con el uso de un Smartphone.

Es por ello que se ha decidido innovar con un software de tipo aplicativo móvil, en donde este a través de una de las mejores tecnologías hoy en día, como lo es la inteligencia artificial, se pueda combatir este problema de manera óptima, en donde el software no solo garantiza el buen funcionamiento de la misma, sino procurar la no distracción del usuario mientras realiza el ejercicio de la conducción, también este se encarga de monitorear la seguridad del usuario cuando este se encuentre en un accidente.

Se analiza de forma superficial la situación de la ciudad de Puebla, México donde la mayoría del transporte es el uso del automóvil propio o en su mayor defecto el uso de plataformas de transporte como lo es Uber, Didi y Cabify, se observa que la mayoría de estas personas siempre portan un teléfono inteligente, por lo cual no se demuestra una falta económica para comprar el recurso de un teléfono inteligente, pero a la vez se observa que este es usado indebidamente para fines totalmente opuestos a la conducción, se hicieron unas entrevistas en donde estos afirman que ellos usan el teléfono móvil como método de distracción al tráfico que se acumula alrededor de la ciudad y estos aprovechan para mandar un mensaje o hacer una llamada a un allegado. Por lo cual esto se ha vuelto una costumbre común, pero hay que darle una solución y que mejor que esta sea a través de la tecnología.

## **1.3 Alcance del proyecto**

Actualmente existen tecnologías capaces de entrar en un “diálogo” con personas que les indican mediante algunos comandos de voz ciertas actividades específicas que son capaces de llevar a cabo



sin la intervención alguna del usuario. Siguiendo una de las tendencias más marcadas actualmente en el mundo tecnológico, la interacción humano-máquina mediante el uso de la voz, se propone la integración de un sistema de infotainment para el auto capaz de entender el lenguaje natural del usuario mediante comandos de voz cotidianos, proporcionando así un sistema que brinde más seguridad y comodidad a las personas mejorando la experiencia de manejo del vehículo, con el único fin de reducir las distracciones a las personas y a su vez disminuir las cifras de accidentalidad.

Actualmente se cuentan con herramientas que cumplan la tarea que se propone en este proyecto, sin embargo muchas de estas no son altamente llamativas para el usuario, debido a que no son altamente intuitivas, por lo cual el usuario simplemente, no les presta atención, muchas veces es debido a que se niegan al cambio, porque piensan que el usuario requiere de una gran experiencia para manejarlas, pero en realidad es que la población ni siquiera le echa un vistazo para enterarse de cómo es que en realidad funcionan las herramientas, es por esto que con este proyecto se incluirá un plus y es de brindarle el menor tiempo de respuesta al usuario y también realizando una interfaz intuitiva para que el usuario solo controle la aplicación con su voz, a su vez reduciendo el consumo de batería del teléfono, lo cual hoy en día es muy valorado por el usuario.

Gracias a los avances tecnológicos que cada día surgen en el mundo, este tipo de sistemas pueden ser muy eficientes e indispensables a la hora de adquirir un vehículo que proporcione las medidas de seguridad suficientes para los usuarios, las cuales se deben cumplir con los estándares o normas establecidas por los entes que regulan dichas normas. Como se trata de un sistema el cual permita la interacción directa con la máquina, no tendrá que preocuparse por mirar el teléfono, simplemente lo podrá dejar a un lado, brindando mucha comodidad al usuario ya que no existirá ningún tipo de cableado que pueda estorbar a la hora de conducir. Este tipo de innovaciones puede ser muy novedosa y atractiva para los usuarios, no solo observando como un simple sistema de interacción sino como una medida de seguridad para los usuarios.

#### **1.4 Metodología de desarrollo de software**

Se planea utilizar la metodología de desarrollo rápido de aplicaciones, ya que es una aplicación a corto plazo y esta permitirá en un futuro tener más escalabilidad, con nuevas funciones. La metodología de desarrollo rápido de aplicaciones fue desarrollada inicialmente por James Martin en 1980. El método de esta se basa en el desarrollo iterativo por medio de la construcción de prototipos y el uso de utilidades de caso, el objetivo principal de esta metodología es la usabilidad, utilidad y la rapidez de ejecución.

La metodología se describe en los siguientes procesos:

#### Modelado de gestión:

En este proceso, se plantea como reunir la información que necesitara el sistema, como la procesara y que información nueva va a generar el sistema; para este caso, se requiere información personal del usuario, la cual se procesara y se generara una salida.

#### Modelado de datos:

En esta fase, se perfilarán los datos, dándole un uso en específico, con información concisa y precisa, la cual permita al sistema tener unas respuestas más acertadas y como conectar estos datos con otros, para que con estos el sistema tome decisiones.

#### Modelado de proceso:

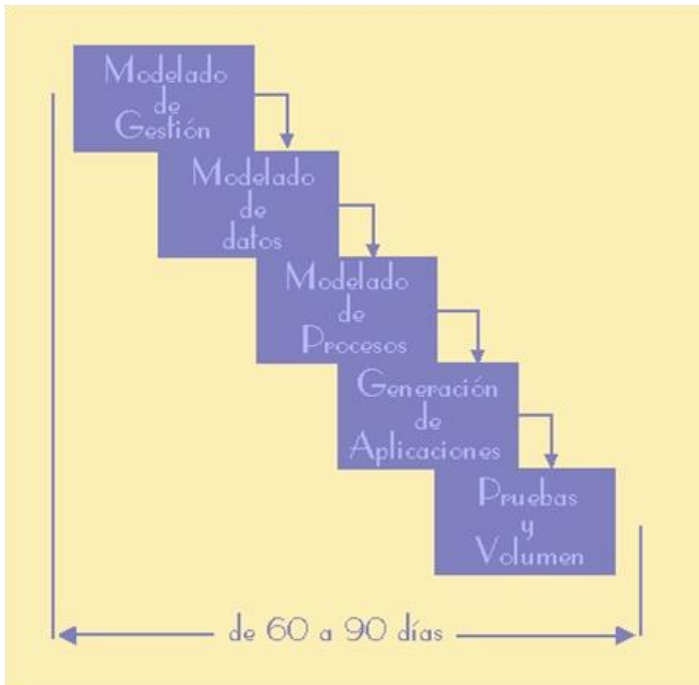
En esta fase, el sistema lo que pretende es procesar los datos que han sido recolectados inicialmente, como la información personal del usuario, la que pasará a ser guardada en una base de datos, que constantemente estará siendo consultado por el sistema y siendo actualizada.

#### Generación de aplicaciones:

En esta sección, el DRA tiene como propósito actualizar técnicas de desarrollo de tercera generación, que pasaran a cuarta generación, la actualización que se llevara a cabo en el sistema es, que, por medio de reconocimiento de voz, se puedan acceder a las funcionalidades de la aplicación WhatsApp que normalmente hoy en día se utilizan, como lo es el envío y recepción de mensajes.

#### Pruebas de entrega:

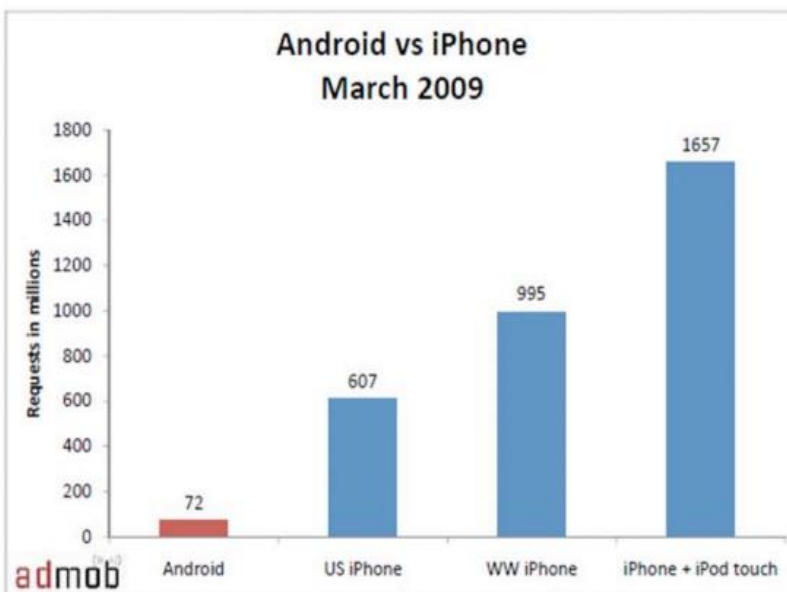
Para las pruebas, se utilizan la recursividad, por lo cual varios componentes del sistema ya están netamente probados, lo que reduce el tiempo en la fase de pruebas, sin embargo, se procederá a probar todos los componentes nuevos y estos se deben emplear en todas las interfaces a fondo.



**Figura 1.** Modelo de desarrollo rápido de aplicaciones. Anónimo (2012)

Metodología Mobile -d

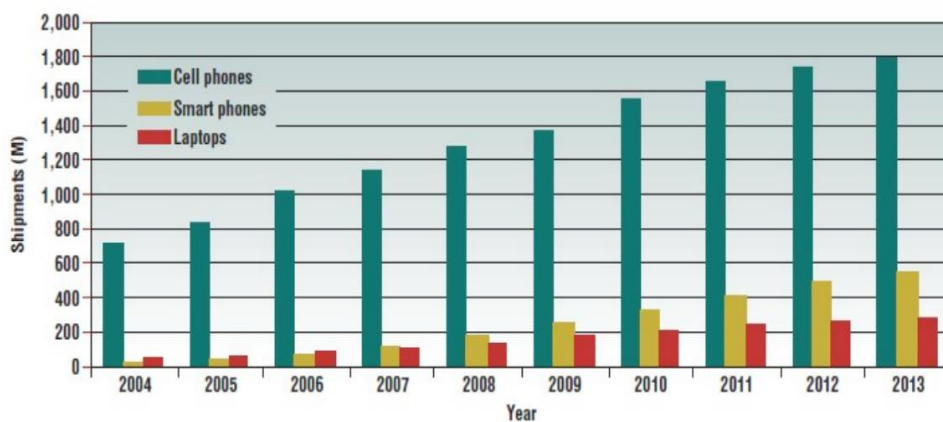
Hoy en día la mayoría de dispositivos móviles son relativamente vacíos, si estos no cuentan con aplicaciones para dar el uso óptimo a estos en donde estas ofrezcan servicios como (Pagos en línea, redes sociales, Juegos o entretenimiento) según la guía oficial del Mobile -d.



**Figura 2.** Android VS iPhone. Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, p., (2009)

Hay cierta competencia a nivel de empresas que ofrecen diferentes tipos de dispositivos móviles, en donde estos se clasifican por gamas siendo la gama alta la más destacable, ya que un usuario promedio gasta un valor cercano a los 500USD para adquirir un dispositivo móvil de alta calidad el cual en promedio le dure 3 años o más, dependiendo del uso, pero además de la gama de un dispositivo móvil también influye en su precio y rendimiento la capacidad que este ofrezca.

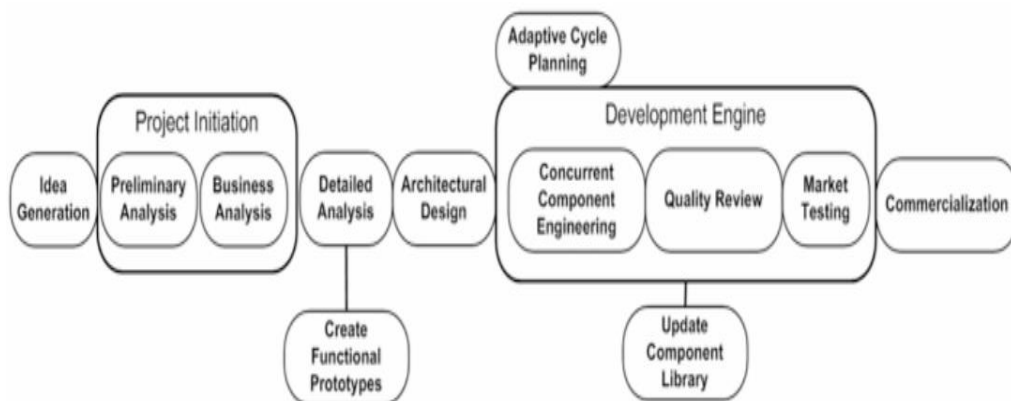
Como se observa en la figura de Android vs iPhone se observa que además de que Apple una gran compañía, ofrece productos Premium de la mejor calidad en software y hardware, por lo que es muy importante que el desarrollador tenga en cuenta este mercado a la hora de hacer la respectiva programación de la app, a continuación, se muestra una gráfica de ventas en las que se observa el incremento de venta en dispositivos móviles inteligentes en comparación con las computadoras portátiles.



**Figura 3.** Figura de venta dispositivos inteligentes. Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, p., (2009)

En esta figura de venta dispositivos inteligentes se puede observar el gran incremento de venta de dispositivos móviles, en donde se hace alta referencia en que hoy en día los dispositivos móviles inteligentes son de gran utilidad, prevalecen en prioridad a la hora de comprar tecnología, ya que un computador portátil puede brindar el mismo Word, correo electrónico etc., siempre y cuando tenga internet, por el contrario, en un dispositivo móvil inteligente, se toman en cuenta otras características como lo es la cámara, velocidad del dispositivo (RAM), capacidad de almacenamiento etc.

Teniendo en cuenta lo anterior mencionado, se procede a utilizar la metodología Mobile -D, esta se basa en el desarrollo de aplicaciones de una forma rápida, en donde se tienen cuenta factores de negocio, requerimientos y detalles de análisis, para este caso como se ha hecho una aplicación híbrida, en donde a partir de una base web para el front-End se ha construido un ciclo para cada apartado del desarrollo, ya que solo se cuenta con un desarrollador Full-Stack en desarrollo móvil, este se ha encargado de dividir el proceso de desarrollo en tres etapas, la primera es requerimientos y necesidades, la segunda es diseño y consolidación del prototipo, la tercera es la programación de la app, el tercer apartado se dividió en dos secciones, siendo la primera Front-End y la segunda Back-End, se requirió hacerlo de esta forma ya que el desarrollador no contaba con la experiencia necesaria para el entrenamiento de algoritmo e inteligencia artificial, por lo que este curso una materia de Inteligencia Artificial en la universidad autónoma del estado de prueba (UPAEP) en donde fue entrenado específicamente para cumplir este requisito y que este proyecto se pudiera dar a cabo, mientras este tomaba habilidad en el desarrollo Back-End , fue adelantando el diseño de la aplicación y la funcionalidad en el modelo (MVC), para esto se tuvo en cuenta también la gráfica proporcionada por la metodología MOBILE-D, donde explica los procesos que se deben seguir a la hora de realizar una aplicación móvil.



**Figura 4.** Programación híbrida. Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, p., (2009)

En la figura de programación híbrida, se observa que hay que realizar prototipos funcionales, en este caso gracias al framework que se utilizó (Angular 9), este nos permite crear modelos reales de la aplicación y que por medio de la aplicación Ionic DevApp, podamos visualizar los modelos y estos sean mostrados al público, este modelo fue presentado en la feria de internacionalización de la UPAEP, donde fue reconocida como una gran innovación potencialmente escalable, por lo

cual daba solución a un problema y era bien recibida por los usuarios, aun sin estar el algoritmo de reconocimiento de voz al 50% de escalabilidad.

Se realizaron algunos test, de los cuales el 30% soluciono la necesidad de cada usuario, por lo cual se realizó un entrenamiento con palabras (ordenes) más específicas para el funcionamiento de la app, lo cual llevo a un retraso en la versión final, ya que se estaba limitando el software, por lo cual el usuario que utilizara la aplicación debía leer un manual de instrucciones, en donde este aprendía a manejar óptimamente la aplicación y a conocer las palabras claves, que fue la versión (1.0.2). Esta versión recibió muchos impactos negativos ya que hoy en día el mercado es tan avanzado que se tiene que ofrecer una ventaja en la aplicación para que esta sea usada, ya que nativamente existen aplicaciones según el dispositivo móvil que el usuario tenga , como lo es Apple con Siri y Android con Google Voice, estos algoritmos tienen más experticia ya que gozan de una comunidad de desarrolladores que trabajan diariamente en facilitar la vida al usuario, por el contrario el desarrollador a cargo de este proyecto solo tenía una base para el correcto entrenamiento del algoritmo el reconocimiento de voz que este utiliza es un algoritmo de código libre realizado en Arizona, Estados unidos, el cual es utilizado para el framework de ionic, las ventajas que este ofrecía al desarrollador, era la fácil calibración de las redes neuronales, en donde estos por peso podían determinar un árbol de decisiones, pero genero una limitación a la hora de la realización del software y fue el reconocimiento de diferentes lenguajes, ya que en el caso del inglés este no detectaba acrónimos y en el español no detectaba las entonaciones , por lo cual afectaba todo el entrenamiento del algoritmo, para esto se utilizó un recurso que ofrece el algoritmo y era crear un árbol de decisiones relacional, en el cual el desarrollador podía omitir después de pasado el nivel en el árbol de decisión.



**Figura 5.** Modelo de planeación MOBILE-D. Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, p., (2009)

En la figura de modelo de planeación MOBILE-D se observa que el apartado de exploración fue cubierto por el stakeholder que fue la universidad autónoma del estado de Puebla, ya que cubrieron la etapa de exploración y la planeación del proyecto, como parte evaluativa de los requerimientos, estos también han sido evaluados previamente en la UPAEP por la profesora Patricia Muratalla en la asignatura de ingeniería de requerimientos, para el apartado de producto, en esta metodología se han realizado apk con los ejecutables para el caso de la plataforma Android, para el caso de la plataforma IOS no se contaba con la licencia de desarrollador de Apple, por lo cual esta plataforma solo fue testada en simuladores de alto flujo (XCODE) con las versiones actuales 14.0.2, cuando se distribuyeron los respectivos apk para la distribución y comercialización, se observó que como sugerencia, pedían cambiar el logo de Partner Car por algo más llamativo, así como los colores de la respectiva aplicación por algo que fuera más futurista, por cual la aplicación quedo con una interfaz morada, otra sugerencia que se recibió, fue el tiempo de respuesta del reconocimiento de voz, como este algoritmo era genérico, su tiempo de respuesta es alto y si no se cuenta con una buena conexión a internet, este demora en dar una respuesta, por esto se decide dejar el código de esta aplicación abierto, para que cualquier persona pueda editarlo y seguirlo mejorando.

Para la arquitectura de este software se utilizó el modelo MVC (modelo, vista, controlador) ya que por defecto el framework maneja esta arquitectura, se hace alta referencia en el controlador, ya que se manejó todo con Javascript.

### 1.5 Definiciones, acrónimos y abreviaturas

Partner Car: Herramienta de asistencia virtual mientras se ejerce la tarea de conducir.

WhatsApp: Aplicación móvil para la comunicación de usuarios a nivel global.

Asistente Virtual: Herramienta que utiliza reconocimiento de voz para hacer tareas del usuario.

Web: Deberá entenderse, cuando se emplee este término, la utilización conjunta del protocolo HTTP y del lenguaje de diseño de páginas HTML para la difusión de contenidos mediante la red Internet.

Aplicación Móvil: Programa de ejecución dirigido a un usuario final para utilizarse en un dispositivo móvil.

Base de datos: Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Lenguaje híbrido: Lenguaje de desarrollo de software para aplicaciones móviles en varios sistemas operativos.

(IEEE Recommended Practice for Software Requirements Specification., 2000)

IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830-1998.

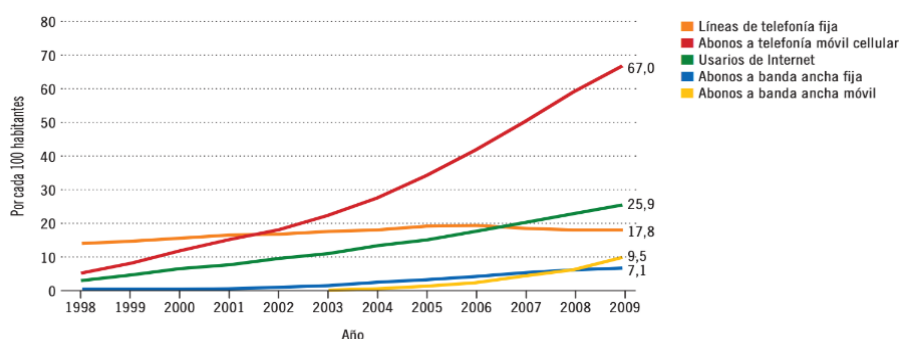


## 2. Marco teórico y estado del arte

Análisis de uso:

Para el análisis de este proyecto se ha tomado en cuenta una publicación de la OMS(Organización mundial de la salud) en donde se analizan los casos en donde los conductores cometen diferentes tipos de faltas , una de ellas y la más común es el uso del celular mientras se ejerce la labor de conducir , un problema muy potencial que cada día crece más, pero esto tiene un origen y es el acceso a la comunicación en todo momento ya que hoy en día los seres humanos sufren de una adicción denominada nomofobia.

Pero este problema conlleva a que hoy en día la tecnología ha avanzado exponencialmente, por lo cual los seres humanos quieren estar conectados y enterados de todo lo que pasa a su alrededor, ya que existen las redes sociales, en la figura 2 se muestra una gráfica realizada por la OMS mostrando como ha crecido el internet por cada 100 habitantes.



**Figura 6.** Uso del celular al volante: Un problema creciente de distracción del conductor. (OMS, 2010)

Como podemos observar los abonos a telefonía móvil celular, estos datos han sido tomados en países de bajo y mediano ingreso, donde se duplica la cantidad de usuarios en donde se ve reflejado el tamaño relativo de esos mercados, ya que el bajo costo de las líneas telefónicas cada día es menor, lo que permite acceder a una gran cantidad de población.

Dado que hoy en día la telefonía móvil es más usada para el uso de las aplicaciones que para las llamadas, debido a que el coste de un mensaje es mucho menor al de una llamada, incluso se conocen tarifas mensuales donde la conexión a redes sociales es la más baja y con mayor impacto en la sociedad con la relación coste y beneficio.

También se puede observar en esta publicación que la mayoría de usuarios de redes sociales son los jóvenes que comprenden la edad desde los 16 años – 24 años y es que en algunos países los

jóvenes pueden obtener su licencia desde los 16 años, por lo que hoy en día es muy común encontrar jóvenes testean con su entorno social mientras conducen.

Reconocimiento de voz:

El reconocimiento de voz hoy en día es una gran herramienta, la cual podemos utilizar en diferentes proyectos que ayuden a la realización de tareas, sin el uso de las manos, para el caso de este proyecto se requiere implementar reconocimiento de voz ya que el usuario necesita sus manos para la conducción del automóvil, pero durante esta labor surgen otra tarea como el cambio de estación de radio o la utilización del móvil para enviar un texto.

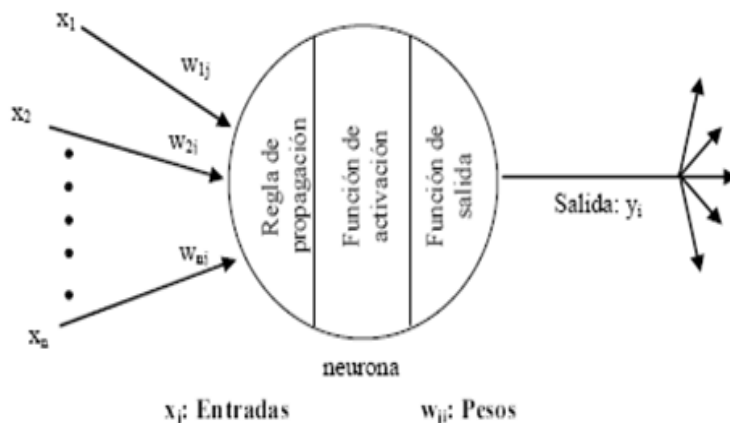
El reconocimiento de voz es una disciplina de la inteligencia artificial que tiene como propósito comunicar a los seres humanos con una computadora , en este caso será la conexión con el teléfono móvil ,el problema existente con este sistema es la fuente del sonido , debido a que hay muchos limitantes y ruidos que perjudican a los sistemas entender las palabras que el usuario dice e incluso algunas palabras del lenguaje nativo que el sistema no puede comprender , por lo que los sistemas terminan haciendo una tarea totalmente diferente a la que el usuario requiere .

Para el aprendizaje del sistema se tienen dos tipos como lo son:

Aprendizaje deductivo: Es donde el desarrollador del sistema le transfiere el conocimiento al sistema, por lo cual es sistema estará sujeto a lo que ya está definido en su código.

Aprendizaje Inductivo:

Es donde el sistema a partir de la información y de su funcionamiento, genera conocimiento automáticamente, estos sistemas, se derivan de modelos ocultos de Márkov o las redes neuronales artificiales que son configuradas automáticamente, a partir del aprendizaje humano.



**Figura 7.** Estructura de una red neuronal. (Advanced tech computing group UTPL,2007)

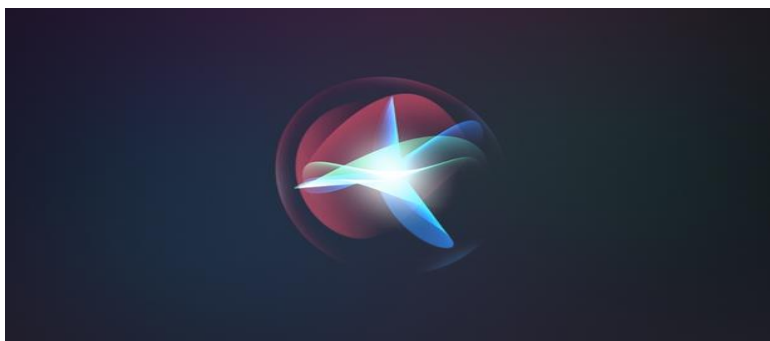
En la figura 3 podemos observar cómo es la estructura de una red neuronal.

Para el desarrollo de este proyecto se utilizará aprendizaje deductivo, ya que el sistema tendrá funciones específicas que cumplir y las cuales el usuario tiene que seguir para el funcionamiento del sistema.

Estado del arte:

Hoy en día observamos que existen muchas herramientas para la optimización de tareas en los seres humanos, en el caso de este proyecto existen algunas herramientas que coinciden en su funcionamiento, una de estas es Siri herramienta desarrollada para dispositivos con sistema operativo Ios y Mac Os. Siri es una inteligencia artificial con funciones de un asistente personal con su propia personalidad para IOS, MacOS, TvOS y WatchOS. Esta aplicación utiliza procesamiento del lenguaje natural para responder preguntas, hacer recomendaciones y realizar acciones mediante delegación de solicitudes hacia un conjunto de servicios web que ha ido aumentando con el tiempo. Esta aplicación para IOS fue el primer producto que se lanzó al público de SRI venture group, el cual es un grupo que se enfoca en el desarrollo de aplicaciones basadas en inteligencia virtual.

Siri tiene el método de aprendizaje inductivo, ya que afirma la empresa desarrolladora que Siri es capaz de adaptarse a las preferencias individuales de cada usuario.



**Figura 8.** Aplicación Siri. (Apple, 2011)

Como siguiente aplicación para el reconocimiento de voz, encontramos a Alexa que es un asistente virtual desarrollado por Amazon, utilizado por primera vez en los altavoces inteligentes de Amazon Echo.

Esta herramienta se compone tanto de un hardware definido por Amazon como por la inteligencia artificial que comprende, esta puede realizar actividades específicas, como lo es:

Reproducción de música

Crear listas de comprobación

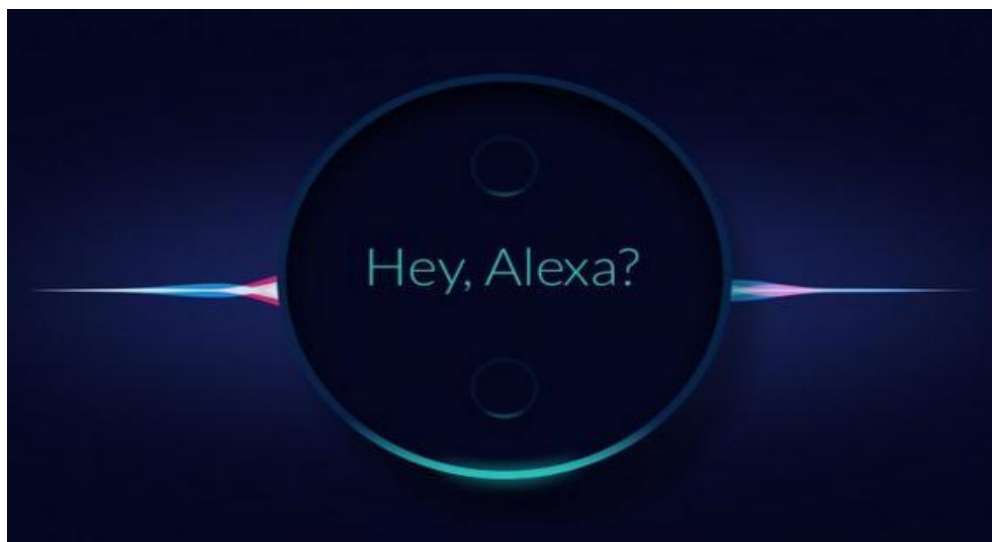
Establecer alarmas o temporizadores

Reproducir Podcasts y audiolibros  
Controlar dispositivos inteligentes  
Proveer información en tiempo real con respecto al clima  
Brindar resúmenes de noticias

Esta herramienta actualmente, cuenta con reconocimiento de voz para los idiomas Inglés, Alemán, Japonés, Frances, Italiano y Español.

Esta herramienta puede controlar dispositivos inteligentes, siempre y cuando tengan la compatibilidad, ya sea dispositivos de una casa, empresa e inclusive auto.

Esta herramienta cuenta con aprendizaje deductivo, ya que cumple funciones específicas que han sido designadas, sin embargo, cuenta con la opción de añadir habilidades (Skills) que los usuarios podrán designarle echas por desarrolladores terceros y que se pueden catalogarse como aplicaciones extra.



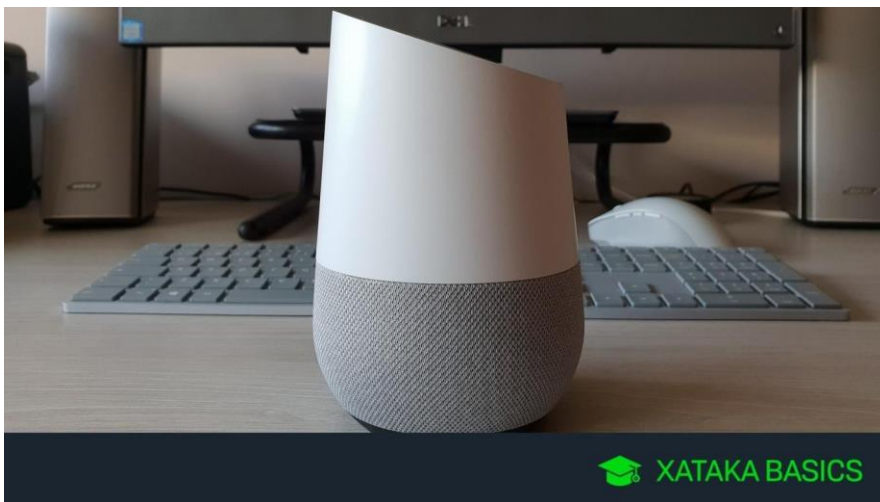
**Figura 9.** “Amazon Alexa”. (Payo, A., 2020)

Estas son las dos herramientas más desarrolladas en la actualidad, pero en los índices de uso, se muestra que estas no son empleadas habitualmente por los usuarios, debido a su complejidad de uso o la rapidez con la que procesan la información, por lo cual para el desarrollo de este proyecto se requiere que el usuario vea un beneficio al usar la aplicación y que en realidad la emplee en su rutina diaria mientras ejerce la tarea de conducir.

Es una gama de altavoces Google en el cual se domótiza el hogar, se toma como referencia para el proyecto partner-car debido a que este utiliza un algoritmo de reconocimiento de voz, lo cual se

asocia un poco al algoritmo que el desarrollador tomo en cuenta, para la realización del proyecto, se toma en cuenta también los patrones que Google Home utiliza.

Google home tienen también la aplicación de Google assistant, donde se pueden pedir tareas básicas como decir la hora, hasta tareas complejas como apagar las luces o cargar el teléfono del usuario, se dice que esta tecnología va en evolución gracias a los avances en domótica, permitiendo así que con solo una actualización del software se puedan manejar muchas más tareas en el futuro.



**Figura 10.** Aplicación Google home. (Fernández, Y., 20202).

En nuestra actualidad hacemos uso de herramientas que hagan más fácil el diario vivir como es el caso de Cortana, según Microsoft (2014):

Cortana es un asistente virtual que funciona con inteligencia artificial específicamente en la rama de reconocimiento de voz, la cual es propiedad de Microsoft por ende esta solo sirve en sistemas operativos Windows, esta herramienta fue innovativa para el proyecto partner- car, ya que utiliza una interfaz muy llamativa, la cual le llamo la atención al desarrollador y tomo como base y paleta de colores esta aplicación.

Como se ha observado esta herramienta es muy potente y llamativa, su funcionamiento también ha sido criticado ya que algunos usuarios experimentan problemas a la hora del reconocimiento de voz.



**Figura 11.** “Aplicación Cortana”. (Sabán, A., 2019).

Dragon Naturally Speaking es una aplicación que se basa en la lectura de correos electrónicos, la cual brinda a las compañías una gran ventaja a la hora de recibir correos e incluso a las personas con dificultad para moverse.

Esta aplicación es sin dudas uno de las mejores aplicaciones de reconocimiento de voz junto a Cortana. Desarrollado por Nuance, esta aplicación de reconocimiento de voz reconoce dictado en varios idiomas con una correcta interpretación de las palabras y comandos de voz para publicar en redes sociales, enviar mensajes de correo, buscar en el navegador de Internet, etc., es realmente muy completo.

Claro que para poder utilizar esta aplicación debemos hacernos con una licencia, ya que no es gratuito. La versión básica de Dragon Naturally Speaking cuesta unos 99 euros. Como se puede observar esta herramienta, es de gran utilidad, pero una de sus desventajas es el precio.



**Figura 12.** Aplicación Dragon Naturally Speaking. (Amazon, 2016)

### 3. Especificación de requisitos de software (ieee 830)

Este documento es una Especificación de Requisitos Software (ERS) del Desarrollo de un aplicativo Móvil Partner-Car que sirva como herramienta de comunicación para los conductores mediante el uso de reconocimiento de voz. La presente ERS ha sido validada por un equipo de pruebas independientes, por lo que se considera libre de errores con las naturales reservas de esta afirmación en el contexto del desarrollo de software.

La presente ERS cumple las directrices establecidas por el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830- 1998”.

Propósito: La presente ERS tiene como audiencia: el equipo de desarrollo de software, la dirección de evaluación de proyectos de la Universitaria Agustiniiana y los usuarios finales del sistema. En especial, y dado que la presente ERS ha sido previamente validada, los principales interesados serán los desarrolladores del sistema para poder iniciar inmediatamente con las actividades de diseño.

Ámbito del Sistema: Actualmente, existen muchos accidentes de tránsito debido al uso del teléfono móvil, ya que los usuarios quieren estar conectados permanentemente a sus redes sociales, por lo que hoy en día los teléfonos móviles no cuentan con una aplicación para solucionar este problema. Ello dificulta enormemente el reducir los accidentes de tránsito causados por el teléfono móvil, ya que el creciente número de conductores adolescentes hace como requisito hoy en día tener un sistema de este tipo.

Es por esto que el sistema de Partner-Car, tiene como objetivo principal automatizar la realización de ciertas tareas que el conductor hoy en día hace manuales, en concreto son las siguientes:

Envío y lectura de mensajes de WhatsApp del dispositivo móvil

Usabilidad del sistema, solo hasta los 80km/h

Confirmar el estado del conductor mientras este va conduciendo.

#### 3.1 Perspectiva del producto

Se proyecta implementar un software de tipo aplicativo móvil que permita enviar y recibir mensajes de WhatsApp mientras el usuario está conduciendo a partir de comandos de voz , además de que permita tener a cada usuario un perfil con su información personal , para en caso de accidente , la aplicación se pueda comunicar con la policía. El software a implementar es de



tipo independiente, ya que la aplicación móvil es la única plataforma en donde se brindará el servicio.

### **3.2 Funcionalidad del producto**

El software está diseñado 100 % funcional donde todos sus módulos funcionan, se hicieron las respectivas pruebas con usuarios con conocimientos en el área de software, quienes brindaron unas críticas constructivas para futuras mejoras y/o escalabilidad del mismo, el producto actualmente tiene una licencia gratis de base de datos con firebase, en donde esta tiene un límite de usuarios (1000) al superarse esta cifra se empieza a regir una cuota mensual por los servicios de Google y el alojamiento de la información de los usuarios, por lo que se recibe como sugerencia hacer una versión Premium, donde se ofrezcan más herramientas en la app y así poder cubrir con los gastos de base de datos, si bien se sabe que podría utilizarse una base de datos de tipo gratuito y desarrollarla desde cero, por cuestiones de tiempo, se realiza la integración con firebase, sin embargo haciendo un balance de costo /beneficio , se observa que la herramienta de firebase es altamente potente y sin ningún problema obvia al desarrollador tiempo en la programación.

### **3.3 Características de los usuarios**

Existen dos tipos de usuarios: Conductores y Supervisores del sistema. Cada uno de estos usuarios, deberá tener mínimos conocimientos en uso de dispositivos móviles Por ello, Partner-Car deberá ofrecer una interfaz de usuario intuitivo, fácil de aprender y sencillo de manejar. Se deberá proporcionar ayuda en línea para todas las funciones del sistema.

### **3.4 Restricciones**

Partner-Car proporcionará dos interfaces para el sistema, la primera será de tipo Web, donde se podrán registrar los usuarios, tener información acerca del sistema y finalmente tener un chat en línea con el chat box de la aplicación para resolver cualquier duda con el sistema.

La segunda interfaz será la de aplicación móvil, donde el sistema se ejecutará con las funciones que posteriormente serán descritas en el apartado de requisitos.

El sistema en la primera fase de desarrollo se verá afectado por un tipo de restricciones, como lo es en la velocidad de respuesta y se verá afectado por la velocidad de red que contenga el usuario final, ya que hay zonas que por su localización geográfica afectan al funcionamiento de la conexión a internet.

Es por esto que la disponibilidad de Partner-Car deberá ser absoluta, excepto en el caso de cortes en la red de comunicaciones.

La restricción natural que contendrá el sistema, sin ninguna opción a cambiar es el limitante de velocidad para el uso de la aplicación, ya que Partner-Car pretende garantizar la seguridad del conductor en el uso de la misma, por ello el límite será amplio acatando a las normas de tránsito que es de 80km/h.

### **3.5 Suposiciones y dependencias**

Los datos de los usuarios serán válidos de por vida, con opción a actualizarse por parte del usuario. La carga de los datos deberá ser realizada por el usuario. Los datos mínimos que el usuario deberá suministrar a Partner-Car son los siguientes:

- Nombre del usuario
- Apellidos del usuario
- Correo electrónico del usuario
- Edad del usuario
- Dirección postal del usuario
- Usuario -email
- Contraseña
- Foto del usuario

### **3.6 Requisitos específicos**

#### **3.6.1 Actores y roles.**

Usuario: Tener un perfil, en el cual suministra información según el uso que va pidiendo la aplicación móvil.

Partner Car: Desarrollo de funcionalidad de la aplicación, manejando la información del usuario y proporcionando la información que este necesite.

Base de datos (Google): Guardar la información de inicio de sesión de cada usuario, y encriptar su información personal.

#### **3.6.2 Requisitos funcionales.**

Envío de mensajes:

Req (01) El usuario debe tener un login que se conecte a la aplicación donde sus datos deben estar registrados.

Req (02) Los datos que se debe registrar el sistema son:

Usuario y mensaje que envió

Velocidad del vehículo

Hora de envió del mensaje

Estado del conductor

Recepción de mensajes:

Req (03) El sistema emitirá una alerta, donde el usuario responderá si desea que el mensaje sea reproducido

Req (04) No se podrán leer dos mensajes al mismo tiempo.

Req (05) El sistema reproducirá el mensaje y mandará una alerta, donde el usuario decidirá si desea responder dicho mensaje o hacer una llamada a ese usuario.

Req (06) El sistema comprobará la velocidad del vehículo para habilitar la opción de hacer las peticiones del usuario.

Req (07) El sistema estará escuchando permanentemente al usuario por si este desea responder un mensaje.

Comprobación del estado del usuario:

Req (08) El sistema emitirá una alerta 10 minutos después de la no actividad del conductor para comprobar el estado.

Req (09) En caso de que el usuario tenga un accidente, este podrá decir la palabra clave “Emergencia” donde el sistema procederá a llamar al servicio de emergencias, para que se informe acerca del accidente.

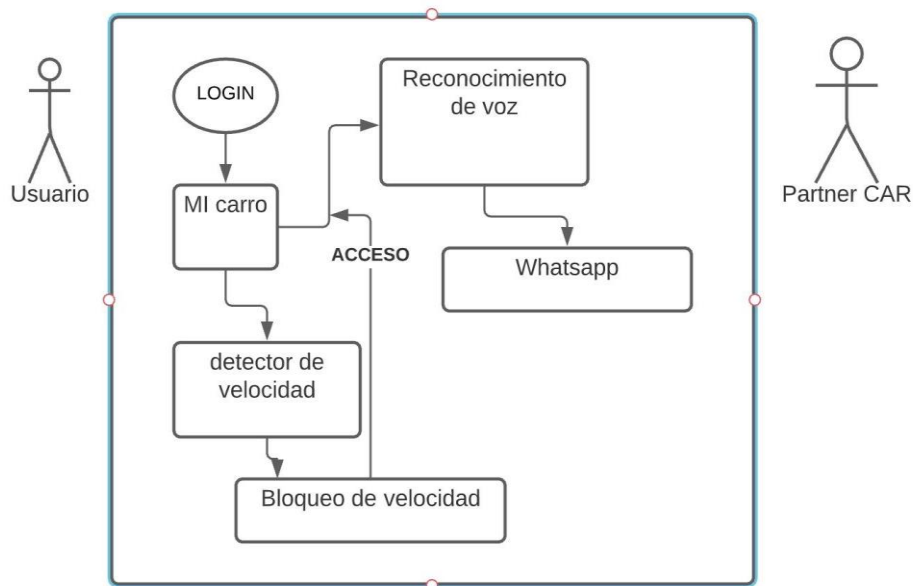
Límite de velocidad

Req (10) Si la aplicación se encuentra desactivada por límite de velocidad, almacenará un estado de “Inactiva”.

Req (11) El sistema deberá emitir una alerta en caso de que la aplicación se desactive por límite de velocidad, para que el conductor quede enterado.

Req (12) En caso de accidente por límite de velocidad, el sistema procederá a emitir todas las alertas para emitir una ficha técnica del usuario, emitiendo sus datos personales y números de contacto.

### 3.6.3 Diagrama de casos de uso.



**Figura 13.** Diseño diagrama de casos de uso. Autoría propia (2020).

#### ACTOR:

Usuario: Este actor es el encargado de brindar toda la información que se le solicite en la aplicación, comenzado por el Login ya que, para hacer este procedimiento, primero debió haberse registrado digitando los siguientes datos:

Nombre

Apellido

Ciudad

Tipo de documento

Número del documento

Foto anexa

Placas de automóvil

Email

Contraseña

Teniendo este registro previo, ya puede hacer uso de la aplicación, como fuente principal se asociará la cuenta de WhatsApp con la aplicación para que esta pueda enviar y reproducir mensajes, que es la función principal de la misma.

Partner Car: Partner-Car como aplicación, solo se encarga de procesar la información que el usuario suministra, no recolecta datos, para uso propio de la aplicación, ya que estos datos van a una base de datos directamente de Google (Firebase), en donde esta cumple con todas las políticas de privacidad de datos.

#### **3.6.4 Especificaciones de los casos de uso.**

Tiempo:

Al ser un proyecto tan ambicioso, se requiere de cierto tiempo de la investigación en el cual el desarrollador entienda correctamente el funcionamiento de la inteligencia artificial y como esta se implementa en el reconocimiento de voz.

Por lo cual el desarrollador ha implementado una técnica de programación llamada recursividad de Datos en la cual no se envían todos los datos para entrenar el algoritmo de inteligencia artificial, si no, se hace una consulta a la base de datos y según esta se hace la respuesta, reduciendo así los tiempos de programación.

Disponibilidad de los expertos en la materia:

El desarrollador ha tenido la experiencia de viajar a Puebla, México a la universidad de UPAEP , donde ha conocido profesores expertos en inteligencia artificial como, donde el desarrollador curso una materia de inteligencia artificial y realizó diferente proyecto, teniendo así un tipo de estudio en el tema de entrenamiento de algoritmos, lastimosamente no ha sido suficiente el tiempo para que el desarrollador conozca todo el mundo de la inteligencia artificial y se recurren a técnicas menos extensas.

### **3.7 Requisitos de rendimiento**

Partner-Car debe dar servicio simultáneamente a 100 usuarios, con un tiempo de respuesta inferior a 10 segundos. Si el número de usuarios que utiliza Partner-Car es mayor a 100, es necesario que el tiempo de servicio sufra una degradación progresiva, hasta un máximo de 200 usuarios. A partir de 200 usuarios se producirá una denegación de servicio.

El tiempo de servicio exigido debe cumplirse tanto en la aplicación como en la página web, salvo en el caso de congestión de la red.

#### **3.7.1 Requisitos tecnológicos.**

La plataforma de implementación de Partner-Car se definirá durante la actividad de diseño detallado, en consonancia con lo indicado en la sección 2.1.

### **3.8 Restricciones de diseño**

Se ha observado que no hay muchos proyectos sobre el tema de manejo de aplicaciones por medio del reconocimiento de voz, las que se han tomado como referencia en la vista son Siri (Apple), Alexa (Amazon), Google Home (Google), pero en ella se observa que respecto a la funcionalidad, no son libres en código, por lo que tiene cierto nivel de dificultad basarse en la tecnología que estas usan.

### **3.9 Atributos del Software del Sistema**

#### Seguridad

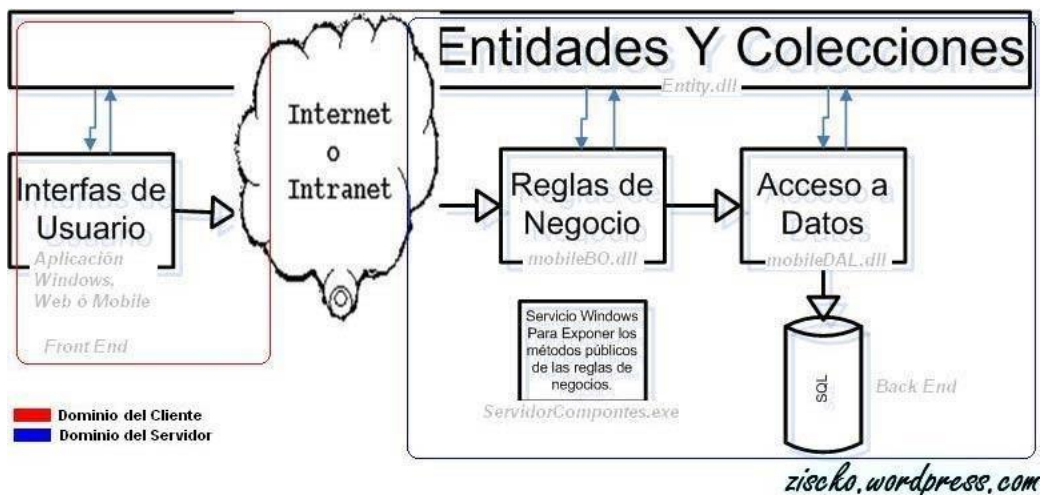
Las operaciones de registro de usuarios, deberán ser almacenadas en la base de datos con su respectivo cifrado, garantizando así la seguridad de los mismos.

#### Auditoría

Todos los estados, registros y usabilidad de la aplicación, quedarán registrados para su posible explotación estadística.

## 4. Diseño del software (iso-12207-1)

### 4.1 Diseño de la arquitectura del Software



**Figura 14.** Arquitectura de Software. (Anónimo, 2010).

Para la realización de la arquitectura, lo primero que se tuvo en cuenta fue una interfaz de usuario amigable, en donde en esta se hace un patrón muy común en el desarrollo de aplicativos móviles, como lo es LCP (less click possibles) en donde efectivamente se logró cumplir con el objetivo.

Después de analizar la arquitectura y patrones de diseño se procedió a hacer un correcto patrón de escalabilidad en donde se observaron algunas falencias, ya que el framework Ionic a la hora de hacer http calls, permite mostrar el resultado de la información siempre y cuando se haga cambio entre módulos(paginas), por lo cual se creó un módulo extra en el desarrollo en donde este se encarga de procesar toda la información de las http calls.

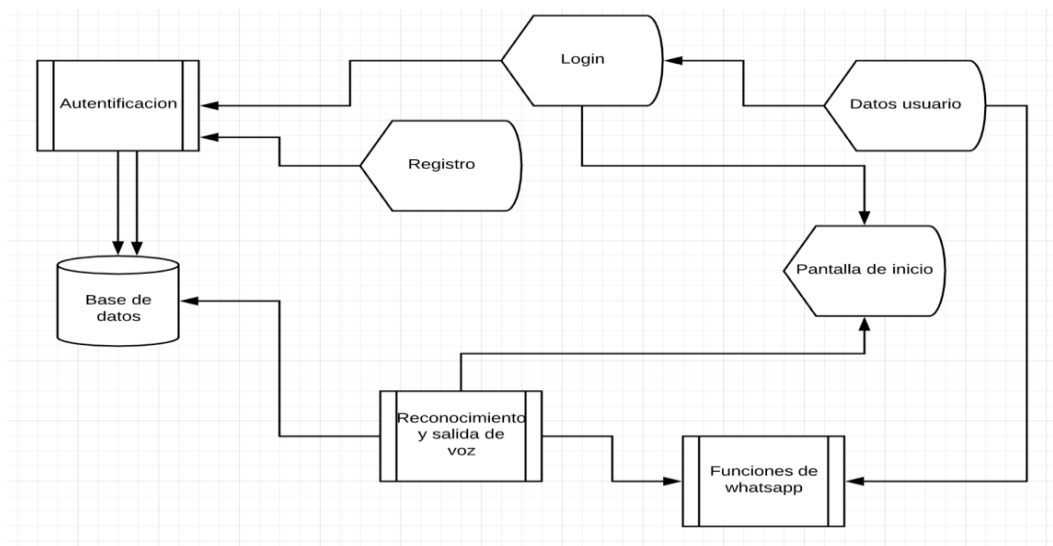
Para las imágenes de la aplicación , se hicieron unos diseños abstractos y futuristas con una herramienta gratuita , llamada ZEDGE móvil, en donde se descargaron y quedaron fijas las imágenes , para futuras escalabilidades se recomienda hacer cambio de estas por cuestiones de políticas de privacidad , como es el caso de Partner-car no se contaba con un diseñador se procedió a tomar este recurso y como esta no iba a ser distribuida comercialmente , fue avalada por la UPAEP para su uso de pruebas.

La arquitectura de la recopilación de datos fue avalada previamente, ya que por políticas de privacidad Google (firebase) no deja ver el diagrama de tablas de la base de datos, por lo cual no

es posible saber cómo o donde a nivel servidor se almacena la información, lo único que se pudo estudiar, fue que la base de datos es no relacional, por lo cual no existen relaciones directamente.

#### 4.2 Diseño detallado del software

Para esta sección, se ha hecho un bosquejo de la funcionalidad del software y la relación con los sistemas asociados.



**Figura 15.** Diseño detallado de software. (Autoría propia, 2020).

Para este sistema se toma en cuenta los siguientes subsistemas del proyecto.

**Autenticación:**

Proceso en el cual se comprobará que el usuario que se está ingresando en la aplicación exista.

**Login:**

Interfaz donde el usuario digitara el usuario y contraseña.

**Registro:**

Interfaz donde el usuario digitara la información personal, para la creación de una nueva cuenta.

**Base de datos:**

Subsistema donde se almacenarán los datos de los usuarios registrados.

**Reconocimiento y salida de voz:**

Subsistema que reconocerá los patrones de voz que el usuario mencione y procesará las peticiones del usuario.

**Funciones de WhatsApp:**



Funciones principales, como lo son el envío y recepción de mensajes.

#### 4.2.1 Diagrama de clases.

En el diagrama de clases se toma como punto de inicio el módulo de inicio de sesión, donde el usuario se loguea con usuario y contraseña, después de que el usuario se loguea, se procede a una pantalla de inicio que es el siguiente módulo, en este se hace la conexión a la base de datos, donde se recopilan los datos que se almacenan y los mensajes que se van a enviar por WhatsApp.

En entrada de datos quedan mensajes para WhatsApp, para el apartado de salida de datos el sistema retornada un mensaje de voz en donde se le informa al usuario el estado del mensaje.

Para el caso de recibir mensajes, la aplicación se activará brindando un mensaje de voz de confirmación y si el usuario está de acuerdo, este mensaje se reproducirá.

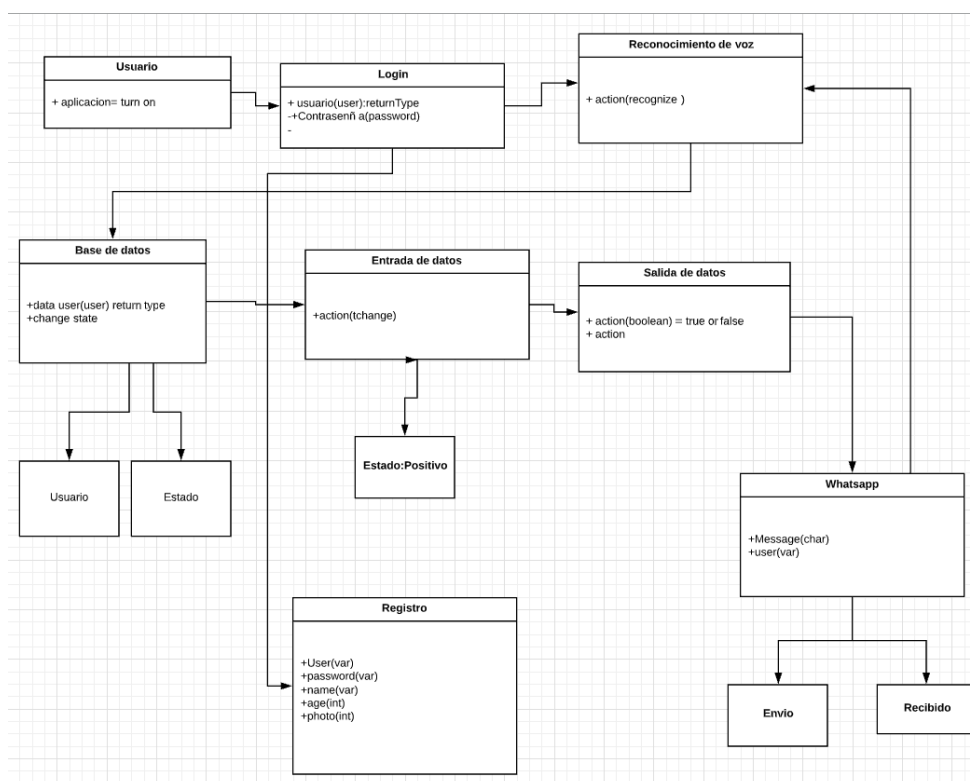


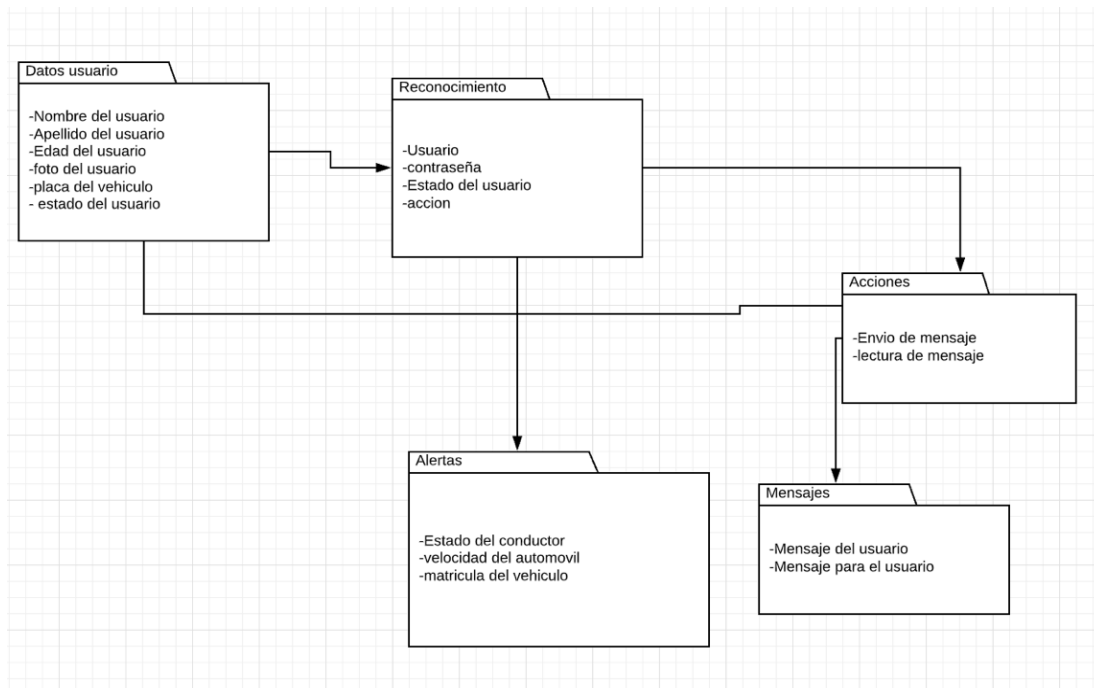
Figura 16. Diseño diagrama de clases”. Autoría propia (2020)

#### 4.2.2 Diagrama de paquetes.

Datos de usuario , en este paquete se guardar los datos de inicio de sesión , como tal utiliza dos módulos el paquete en donde se procede a guardar imágenes que la interfaz contiene , como lo es la foto de perfil del usuario que viene desde la base de datos encriptada en base 64 , se utiliza la directiva del framework Angularar fire , para futuras actualizaciones se recomienda hacer

actualización de las Apis y librerías del framework , ya que este framework es actualizado frecuentemente y los módulos de interfaz tienden a cambiar la estructuras.

Reconocimiento, para este paquete se utilizaron dos módulos internos para el procesamiento de la información y la muestra de la misma, como fue mencionado anteriormente, el framework tiene como regla que en la HTTP Calls se tiene que cambiar de modulo constantemente, por ello el uso de dos módulos.

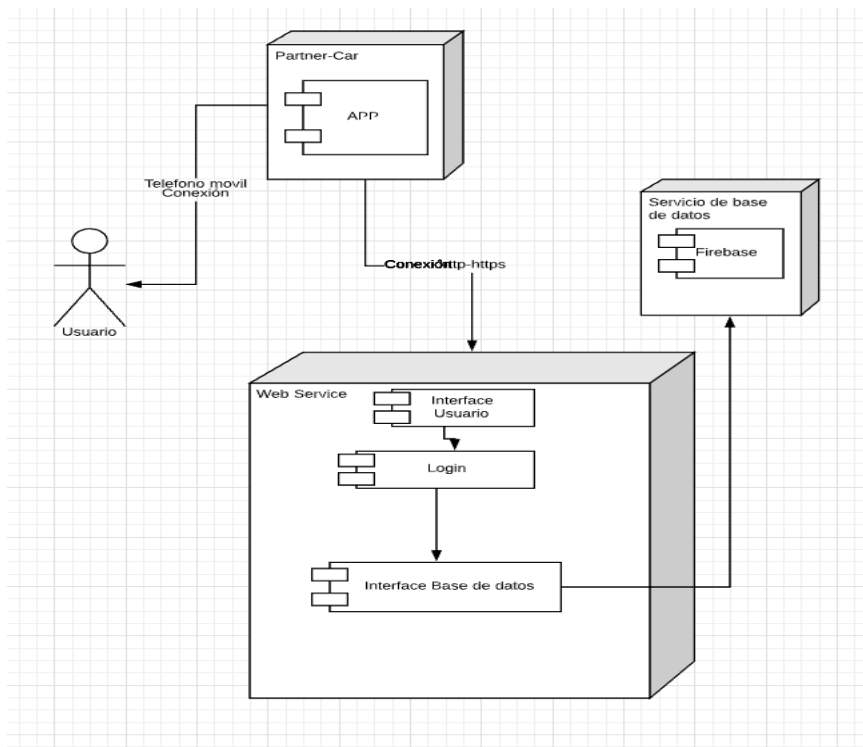


**Figura 17.** “Diseño diagrama de paquetes”. Autoría propia (2020)

#### 4.2.3 Diagrama de despliegue.

En el diagrama de despliegue, se pretende mostrar el recorrido o mapeo que la aplicación dispone. El orden de esta es:

1. Inicio de sesión
2. Pantalla inicio
3. Términos y condiciones
4. Bloqueo del teléfono



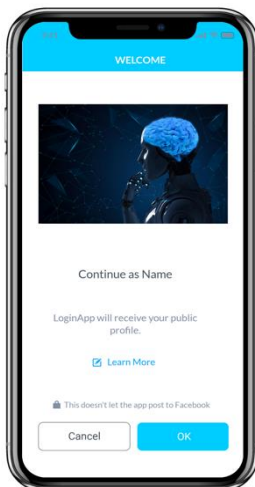
**Figura 18.** Diagrama de despliegue. Autoría propia (2020)

### 4.3 Diseño de interfaz

#### 4.3.1. Interfaz gráfica del usuario.

Pantalla de Inicio:

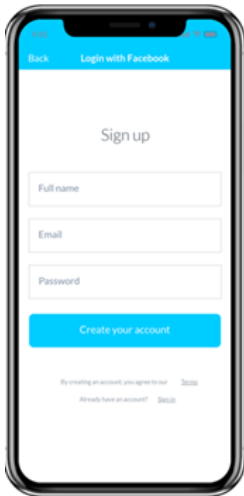
En esta interfaz lo que se pretende, es que el usuario pueda autenticarse, para acceder a la funcionalidad del sistema.



**Figura 19.** Screenshot Marvel App: Pantalla de inicio. Autoría propia (2020)

## Registro de Usuarios

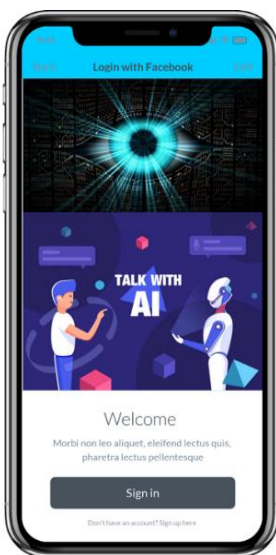
En esta interfaz lo que se pretende es registrar nuevos usuarios al sistema.



**Figura 20.** Screenshot Marvel App: Registro de usuario. Autoría propia (2020)

## Interfaz de inicio

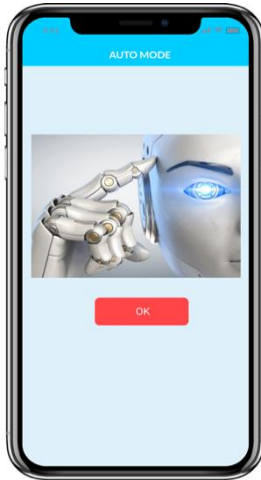
Esta interfaz esta designada para tener el mínimo contacto con el usuario, ya que la idea del sistema es que funcione con reconocimiento de voz, por lo cual solo contara con un botón de salida.



**Figura 21.** Screenshot Marvel App: Interfaz de inicio. Autoría propia (2020)

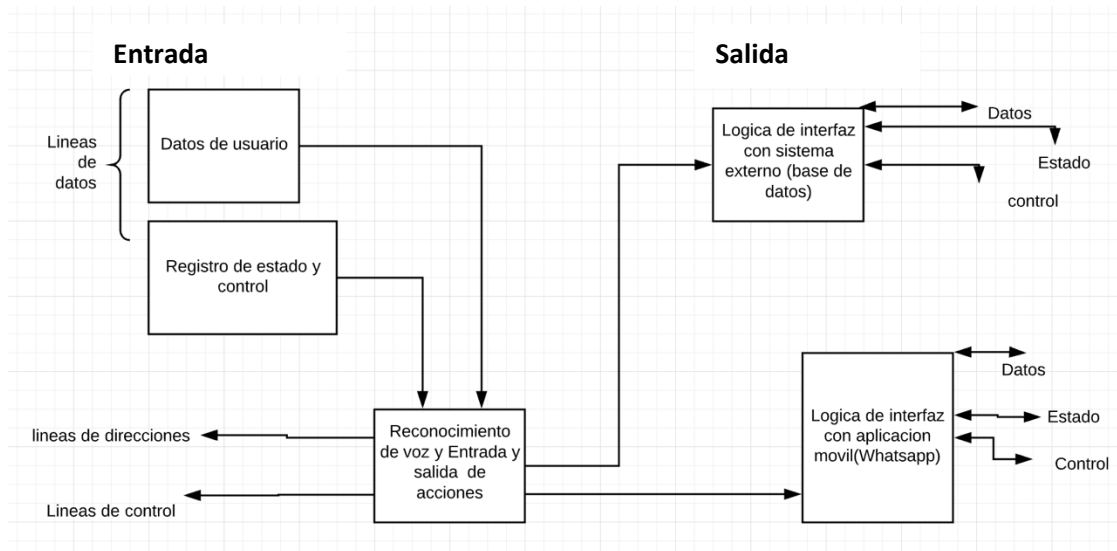
Interfaz de ayuda:

Esta interfaz lo que pretende, es prestar servicio al cliente, en caso de que tenga un inconveniente con la aplicación y este solo se activara cuando el sistema detecte una velocidad menor a 40km/h.



**Figura 22.** Screenshot Marvel App: Interfaz de ayuda. Autoría propia (2020)

#### 4.3.2 Interfaces de entrada y salida.



**Figura 23.** Interfaces de entrada y salida. Autoría propia (2020)

## 5. Implementación

### 5.1 Plataformas de desarrollo

Para el desarrollo de este proyecto se tienen en cuenta nuevas tecnologías que existen en el mercado y que ayudan al desarrollador a realizar la tarea de codificación más ágil, por lo cual obliga al desarrollador a capacitarse en estas tecnologías que posteriormente harán optimizar ciertas tareas que en el pasado eran más tediosas, como el desarrollo de Back-End con Android Studio y después organizar todo este desarrollo en una interfaz.

Se ha decidido utilizar el Framework de desarrollo híbrido de aplicaciones IONIC en su versión 5.4.6 que en la fecha de realización del proyecto es la versión actual, la cual cuenta con el lenguaje de programación de type script y Angular; para el desarrollo del diseño se utiliza el lenguaje de estilos Sass, el cual permite utilizar las funcionalidades del CSS tradicional y para el diseño de la estructura de interfaz, se utilizara HTML 5.

Lo que permite este framework a los desarrolladores, es poder realizar un solo desarrollo para dos plataformas móviles, como lo son Android y IOS, permitiendo optimizar el tiempo y que los desarrolladores eviten la tarea de capacitarse en cada herramienta para el desarrollo nativo, como lo puede ser Java (Android) o Swift 4(IOS).

Según la especificación del ERS, para este proyecto se utilizarán las siguientes plataformas:

Web (Pagina web, sección de ayuda y creación de usuarios)

Aplicación Móvil (sistemas operativos Android y IOS), donde se obtendrá una aplicación con completa funcionalidad designada para el usuario.

### 5.2 Base de datos

Para este proyecto se tiene designado utilizar el motor de base de datos de Firebase (Google), en cual hoy en día tiene una gran demanda en el mercado, permitiendo a los desarrolladores, tener una base de datos con almacenamiento determinado en Google, lo que permitirá en este proyecto almacenar toda la información en la nube y así en el desarrollo simplemente se pueda consumir un API, que en la creación de la base de datos Google inc, genera automáticamente.

Se ha decidido escoger esta base de datos , ya que con el motor MYSQL hoy en día se complica la tarea de realizar la conexión con aplicaciones móviles , por lo cual es importante en este desarrollo rápido de aplicaciones , aplicar herramientas que están al mercado y que un desarrollador con un perfil no tan alto pueda realizar este tipo de proyectos , teniendo también la ventaja de

almacenar audio y video en la base de datos , permitiendo que la aplicación en un futuro sea escalable , pudiendo adaptarle otro tipo de habilidades .

### **5.3. Infraestructura de Hardware y redes**

Para la implementación de este proyecto, es necesario que el usuario cuente con un hardware (Teléfono móvil), el cual tenga una versión de Android superior a la 6.0 que se le conoce como Android Marshmallow. En el caso de los dispositivos que cuenten con hardware iOS, tendrán que tener la versión superior a la 10.

## 6. Pruebas del Software

### 6.1 Pruebas del Software

Ciclo de Pruebas de Calidad de Software



Imagen elaborada por [www.pmoinformatica.com](http://www.pmoinformatica.com)

**Figura 24.** Ciclo de pruebas Software. (Oficina de Proyectos de Informática, 2018).

Para la realización de pruebas de software se tomó en cuenta el ciclo de vida del software, para esto fue necesario realizar pruebas unitarias de cada módulo y que estos hicieran su tarea independiente y colectiva.

Como prueba unitaria, se compiló la aplicación para Android y iOS, la cual salió 100% efectiva.

#### Prueba 1

```

? Starter template: sidemenu
✓ Preparing directory ./partnerCar - done!
✓ Downloading and extracting sidemenu starter - done!
? Integrate your new app with Capacitor to target native iOS and Android? (y/N) █
  
```

Para la integración del framework se requirió de capacitor, ya que cordova, utiliza librerías obsoletas que perjudican la escalabilidad del código y la aplicación.

#### Prueba 2



```

create mode 100644 src/index.html
create mode 100644 src/main.ts
create mode 100644 src/polyfills.ts
create mode 100644 src/test.ts
create mode 100644 src/theme/variables.scss
create mode 100644 src/zone-flags.ts
create mode 100644 tsconfig.app.json
create mode 100644 tsconfig.json
create mode 100644 tsconfig.spec.json
create mode 100644 tslint.json

```

**Your Ionic app is ready! Follow these next steps:**

Se hace la depuración de código, para detectar crashes que puedan afectar el rendimiento de la aplicación, se evitan estos por medio de feedback integrados en la aplicación con datos de contacto, como sugerencias se recibió que la aplicación necesitaba mejoras de diseño.

### Prueba 3

```

jorgemontoya@MacBook-Pro-de-Jorge partnerCar % ls
angular.json          e2e                  node_modules         src                  tsconfig.spec.json
browserslist         ionic.config.json   package-lock.json    tsconfig.app.json   tslint.json
capacitor.config.json karma.conf.js        package.json          tsconfig.json

```

Revisión de arquitectura de la aplicación, donde los json se complementen llamando a las librerías necesarias en el proyecto incluida la de la base de datos (FireAuth).

### Prueba 4

```

[INFO] Development server running!

Local: http://localhost:8100
External: http://192.168.0.7:8100

Use Ctrl+C to quit this process

[INFO] Browser window opened to http://localhost:8100!

[ng] Date: 2020-12-04T20:04:39.586Z - Hash: 93f801d7e3027726ba85
[ng] 61 unchanged chunks
[ng] Time: 996ms
[ng] : Compiled successfully.

```

Se activa un puerto en un servidor local, donde varios usuarios a través de la app de IONIC DEVAPP pueden acceder a la aplicación sin necesidad de un apk, por lo cual se realizan los primeros test de funcionalidad.

## **6.2 Pruebas de usabilidad**

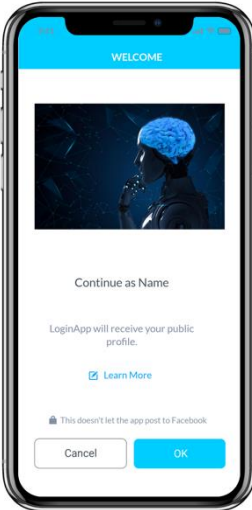
Para estas pruebas se realizó una distribución del software a 5 personas, estas personas fueron, Patricia Muratalla, Jorge Aguilar Cisneros, Paola Montoya, Alejandro Vela, Santiago Puentes. Estas personas tuvieron la versión beta de la aplicación en donde esta no tiene el mejor rendimiento ya que estas pruebas fueron en noviembre 16,17,18,19,20 del 2019, la prueba de usabilidad consistía en que ellos manejaban su vehículo por la ciudad de Puebla y solo podían utilizar su celular por medio de la aplicación, la primera prueba que se realizó el día 16 de noviembre de 2019 se tuvo con resultado erróneo ya que el reconocimiento de voz se encontraba fallando y solo reconocía patrones asociados a emergencias, por lo cual solo detectaba la velocidad del vehículo y mandaba los mensajes de alerta.

El día 17 de noviembre, se solucionó el problema de reconocimiento y se ajustó a un patrón más amplio, tuvo efectos positivos, pero se tuvo un error de detección de palabras (dichos) ya que el algoritmo solo detectaba palabras generales del idioma español, lo cual se complicó un poco, pero la experticia del desarrollador llevo a la integración de una librería externa que trabaja con el lenguaje de JavaScript en el cual se detecta un parafraseo y dichos del idioma español en diferentes países.

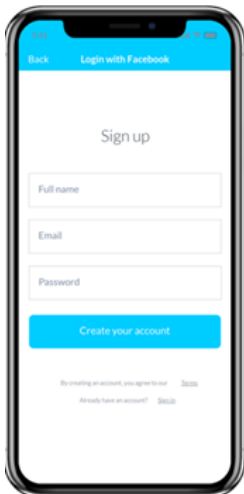
El día 18 de noviembre se observó que la aplicación funciona perfectamente, había sugerencias por parte de los docentes de la UPAEP (Jorge Cisneros, Patricia Muratalla) como lo fue el tiempo de respuesta que demoraba en cargar. a lo que se expresó que este tiempo tiene muchas variables de tiempo como lo son, la conexión estable de internet y la velocidad con que el árbol de decisiones del algoritmo actuaba, se recomienda que, si se desea escalar esta aplicación, se vuelva a analizar el árbol de decisiones y ver en el tiempo que se quiera realizar la escalabilidad alguna herramienta que clasifique las ramas de dicho árbol.

El día 19 de noviembre se dio visto bueno a la aplicación y se dio como una innovación en la UPAEP, en la cual se hace la invitación a seguirla mejorando, ya que tiene un gran potencial y puede solucionar otro tipo de problemas siguiendo el mismo patrón de desarrollo.

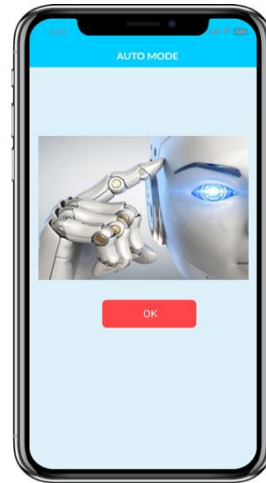
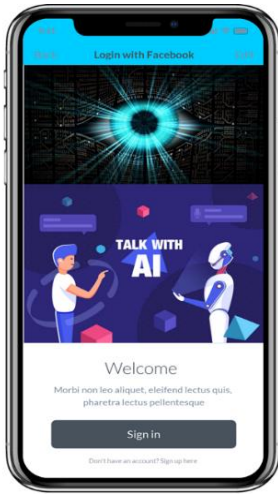
El día 20 de noviembre se hace un back up del código de la app con sus respectivas plataformas el cual se conserva en manos del desarrollador y se empieza a realizar la documentación de dicho software.



**Figura 25.** Screenshot Marvel App: Pantalla de inicio. Autoría propia (2020)



**Figura 26.** “Screenshot Marvel App: Registro de usuario”. Autoría propia (2020)



**Figura 27.** Screenshot Marvel App: Interfaz de inicio. Autoría propia (2020)

**Figura 28.** Screenshot Marvel App: Interfaz de ayuda. Autoría propia (2020)

### **Conclusiones y recomendaciones**

Para el Desarrollo de este proyecto, la universitaria Agustiniiana ha dado el beneficio de intercambio a Jorge Luis Montoya Parada para que se traslade a Puebla , México a la universidad autónoma del estado de puebla(UPAEP), en donde el estudiante curso la materia de inteligencia artificial , la cual sirvió como requerimiento para la realización de este proyecto y su vida profesional , el desarrollo de este proyecto estuvo dirigido por el director de proyectos de la Uniagustiniana Mauricio Alonso Villalba y asesorado por los docentes de la UPAEP Patricia Muratalla (Docente Ingeniería de requerimientos) , Jorge Cisneros (Director proyectos UPAEP en ingeniería de software), los cuales estuvieron pendientes en los procesos de realización del proyecto , como conclusiones se tiene que este software es potencialmente escalable y que puede brindar nuevas soluciones a diferentes modelos , como lo es la compra de alimentos a través de una aplicación web , que sirva a personas con discapacidad en su movilidad.

También es potencialmente escalable en la prevención de maltrato familiar, ya que a través de patrones en el árbol de decisiones ya creado, se puede alterar y entrenarlo para que detecte este tipo de problemas y se pueda comunicar con las autoridades pertinentes. Se hace una invitación a los jóvenes desarrolladores a que innoven con este tipo de tecnologías, ya que si se hace un estudio previo estas son altamente potentes y de fácil mantenimiento con respecto al código, el mercado también para esta tecnología es muy amplio ya que con herramientas así, se puede ayudar al usuario a que muchos de sus procesos sean automáticos y este solo con unos pequeños pasos, pueda hacer tareas alternas y estas no afecten su salud e incluso su vida.

Se provee una herramienta que puede combatir un problema que actualmente es muy común y ofrece un factor diferenciador de las demás aplicaciones el cual es, automatizar un proceso que hasta hoy en día es manual, como lo es el envío de mensajes a través de WhatsApp.

Queda también como experticia una gran experiencia profesional para el desarrollador, donde aumento su nivel de análisis y lógica en la programación y donde este puede aplicar sus conocimientos en empresas de alta demanda en desarrollo de software.

### Referencias

- Anónimo. (2012). Modelo de Desarrollo Rápido de Aplicaciones [Figura]. Recuperado de: <https://curiosisimos.wordpress.com/linux/modelo-de-desarrollo-rapido-de-aplicaciones/>
- Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, P., (2009). Android VS iPhone [Grafica]. Recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)
- Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, P., (2009). Figura de venta dispositivos inteligentes [Grafica]. Recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)
- Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, P., (2009). Programación híbrida [Figura]. Recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)
- Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, P., (2009). Modelo de planeación MOBILE-D [Esquema]. Recuperado de: [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf)
- Organización Mundial de la Salud (2019). Uso del celular al volante: Un problema creciente de distracción del conductor [Figura]. Recuperado de: <https://www.who.int/es>
- Palacios, F. (2003). La Neurona Artificial [Figura]. Recuperado de: <https://advancedtech.wordpress.com/2007/08/31/elementos-basicos-de-una-red-neuronal-artificial/>
- Apple. (2011). Aplicación Siri [Figura]. Recuperado de: <https://www.apple.com/mx/siri/>
- Payo, A. (2020). Amazon Alexa [Figura]. Recuperado de: <https://developer.amazon.com/docs/alexa-voice-service/logo-and-brand.html>
- Fernández, Y. (2020). Aplicación Google home [Figura]. Recuperado de: <https://www.xataka.com/basics/google-home-trucos-algun-extra-para-sacarle-todo-partido-a-tu-altavoz-inteligente#:~:text=Se%20trata%20de%20un%20gama,y%20conectado%20a%20tu%20Wi-Fi.>
- Sabán, A., (2019). Aplicación Cortana [Figura]. Recuperado de: <https://www.genbeta.com/windows/microsoft-silenciara-a-cortana-instalacion-windows-10-administradores-no-se-vuelvan-locos>

Amazon. (2016). Aplicación Dragon Naturally Speaking [Imagen]. Recuperado de:  
<https://www.amazon.com/-/es/K409A-G00-11-0/dp/B003VNCRNQ>

Anónimo. (2010). Arquitectura de Software [Figura]. Recuperado de:  
<https://zisko.wordpress.com/2010/09/04/arquitectura-de-software/>

Figura 24. Oficina de Proyectos de Informática. (2018). Ciclo de pruebas de calidad de Software [Figura]. Recuperado de: <http://www.pmoinformatica.com/p/pruebas-de-software.html>

¿Qué es Cortana? (2014). Recuperado de: <https://support.microsoft.com/es-es/topic/-qu%C3%A9-es-cortana-953e648d-5668-e017-1341-7f26f7d0f825>