

UNIVERSIDAD PRIVADA ANTENOR ORREGO
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y
SISTEMAS



APLICACIÓN MÓVIL PARA DIAGNOSTICAR POSIBLES FALLAS
AUTOMOTRICES UTILIZANDO LA HERRAMIENTA IBM WATSON
PARA LA EMPRESA VECARS & TRUCKS S.A.C.

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERO DE COMPUTACIÓN Y SISTEMAS

LÍNEA DE INVESTIGACIÓN: Sistemas Inteligentes

AUTOR: Br. Aliaga Gálvez, Gustavo Helí
Br. Aznaran Abarca, Jorge Arturo

ASESOR: Ing. Heber Gerson Abanto Cabrera

TRUJILLO - PERÚ

2019

ACREDITACIONES

TÍTULO:

**“APLICACIÓN MÓVIL PARA DIAGNOSTICAR POSIBLES FALLAS
AUTOMOTRICES UTILIZANDO LA HERRAMIENTA IBM WATSON PARA LA
EMPRESA VECARS & TRUCKS S.A.C”**

AUTORES:

**Br. Aliaga Gálvez, Gustavo Helí
Br. Aznaran Abarca, Jorge Arturo**

APROBADO POR:

**Ing. Heber Gerson Abanto Cabrera
ASESOR
N° CIP 106421**

**Ing. Carlos Alberto Gaytan Toledo
PRESIDENTE
N° CIP 84519**

**Ing. Agustin Eduardo Ullón Ramirez
SECRETARIO
N° CIP 137602**

**Ing. Edward Fernando Castillo Robles
VOCAL
N° CIP 192352**

PRESENTACIÓN

Señores Miembros del Jurado:

Dando cumplimiento y conforme a las normas establecidas en el Reglamento de Grados y Títulos y Reglamento de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego, para obtener el título profesional de Ingeniero de Computación y Sistemas, se pone a vuestra consideración el informe del Trabajo de Investigación Titulado “**APLICACIÓN MÓVIL PARA DIAGNOSTICAR POSIBLES FALLAS AUTOMOTRICES UTILIZANDO LA HERRAMIENTA IBM WATSON PARA LA EMPRESA VECARS & TRUCKS S.A.C**”, con la convicción de alcanzar una justa evaluación y dictamen.

Trujillo, 07 de Mayo de 2019

Aliaga Gálvez, Gustavo Helí
Aznaran Abarca, Jorge Arturo

DEDICATORIAS

Dedico esto a mi mamá Patty por apoyarme en cada paso que di, ser un ejemplo de superación y fuente de inspiración para buscar ser cada día mejor, y junto a mis abuelos ser la más grande motivación para seguir adelante, Gracias por cada lección.

A mi hermana Ángela que ha sido la razón de esforzarme aún más y demostrarle que con entrega y empeño se puede llegar a donde sea.

Gustavo.

Dedico esta tesis a Dios, sostén para sacar adelante este proyecto, a mi hijo Arthur por quien llegue hasta aquí y seguiré adelante. Mi familia Jorge, Patricia y Pedro por impulsarme y convencerme que puedo. Mi esposa Raizza por ser el apoyo y soporte de este largo camino. Gracias a todos.

Jorge.

AGRADECIMIENTOS

Agradecemos a cada persona que nos ha tendido la mano y fue un soporte para nosotros en esta travesía hasta la culminación de nuestra tesis. Nos sentimos realmente agradecidos con todos ustedes, a nuestro asesor Heber Abanto y a nuestros jurados Agustín Ullón, Fernando Castillo y Carlos Gaytán gracias a sus observaciones pudimos hacer un mejor trabajo, a nuestros seres queridos que nos motivaron aún más a lograr cada objetivo. A nuestros amigos de la Universidad Omar Becerra, Alexandra Castillo, con quienes siempre compartimos grandes momentos y “con fe” llegamos a esto muchachos, ahora les toca a ustedes, gracias a Karique un amigo que conocimos en el paso de esta aventura quien nos apoyó bastante a aterrizar en ciertas ideas y aconsejarnos.

Gracias a todos los que la idea que pensábamos realizar les pareció descabellada, pero dijeron “Yo estoy contigo”, y nunca perdieron la fe en nosotros que podríamos lograrlo fue largo el viaje, y en titánicos momentos llegaron siempre a tener las palabras de aliento para levantarnos y seguir. Esto es para ustedes.

RESUMEN

En la presente tesis se elaboró el desarrollo de la Aplicación Móvil para Diagnosticar Posibles Fallas Automotrices utilizando IBM Watson para la Empresa Vecars & Trucks SAC, que sirva como canal de atención a los usuarios de dicha empresa, con la finalidad de brindar un servicio de diagnóstico accesible para sus clientes y en el momento que ellos lo requieran.

El Proyecto se enmarco en la investigación aplicada en base a los sistemas inteligentes y como a través de ellos se pueden elaborar específicamente soluciones que ofrezcan servicios de ayuda e información por medio de bases de conocimiento, Para la elaboración y desarrollo del proyecto se recopiló información para tener una idea clara de lo que se buscaba hacer. De esta manera se pudo trazar un plan y dividirlo en dos partes: La construcción de la base conocimiento y flujos de comunicación utilizando la herramienta IBM Watson, y por otro lado el análisis, diseño e implementación de la aplicación móvil basándose la metodología ICONIX en la plataforma Android Studio 2.3.

Luego del desarrollo de la aplicación se estableció un entorno de prueba e interacción para evaluar la capacidad de la solución; posteriormente se utilizó un conjunto de técnicas e instrumentos para la obtención de información con el fin de conocer mejor el desempeño que se obtuvo a través de la propuesta empleada.

De esta manera se puedo estimar la mejora significativa del 10 % aproximadamente obtenida tras la evaluación realizada a la propuesta de solución.

Palabras Clave: Aplicación Móvil, fallas automotrices, diagnóstico, Aplicación móvil, Sistemas Inteligentes, IBM Watson.

ABSTRACT

In this thesis the development of the Mobile Application to Diagnose Possible Automotive Faults was developed using IBM Watson for the Vecars & Trucks SAC Company, that serves as a channel of attention to the users of said company, in order to provide a diagnostic service accessible to their clients and at the time they require it.

The Project is framed in applied research based on intelligent systems and through them can be specifically developed solutions that offer help and information services through knowledge bases. For the elaboration and development of the project information was collected for Have a clear idea of what you were looking for. In this way a plan could be drawn and divided into two parts: The construction of the knowledge base and communication flows using the IBM Watson tool, and on the other hand the analysis, design and implementation of the mobile application based on the ICONIX methodology in the Android Studio 2.3 platform.

After the development of the application, a test and interaction environment was established to evaluate the solution's capacity; Subsequently, a set of techniques and instruments was used to obtain information in order to better understand the performance obtained through the proposal used.

In this way we can estimate the significant improvement of approximately 10% obtained after the evaluation made to the proposed solution.

Key Words: Mobile Application, automotive failures, diagnostics, Mobile Application, Intelligent Systems, IBM Watson.

ÍNDICE

ACREDITACIONES	ii
PRESENTACIÓN	iii
DEDICATORIAS.....	iv
AGRADECIMIENTOS.....	v
RESUMEN.....	vi
ABSTRACT	vii
ÍNDICE.....	viii
ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS	xii
1. INTRODUCCIÓN	1
1.1. Planteamiento del Problema.....	1
1.2. Delimitación del Problema.....	2
1.3. Formulación del Problema	2
1.4. Formulación de la Hipótesis	2
1.5. Objetivos de Estudio	2
1.5.1. Objetivo general.....	2
1.5.2. Objetivos específicos	2
1.6. Justificación del estudio	3
1.7. Limitaciones del estudio	4
2. MARCO TEÓRICO	5
2.1. Antecedentes	5
2.1.1. Internacional	5
2.1.2. Nacional.....	6
2.1.3. Local	7
2.2. Definiciones	7
2.2.1. Sistemas	7
2.2.2. Cognición.....	8
2.2.3. Inteligencia.....	8
2.2.4. Artificial.....	9
2.2.5. Inteligencia Artificial:.....	9
2.2.6. Computación Cognitiva	10
2.2.7. Software (Castellano Casas, 2001)	10
2.2.8. Bot (Haj-Salej, 2017).....	10
2.2.9. Chat (Vanegas, Marin, & Vargas, 2007)	11
2.2.10. Chatbox (Anda_Lucía_es_Digital, 2017).....	12
2.2.11. Framework (COMPANYRED, 2016)	13

2.2.12.	Android (GITBOOKS_AJGALLEGO, 2010).....	15
2.2.13.	API (Merino, 2014).....	19
2.2.14.	Interface (Wikibooks_Anon, 2016).....	19
2.2.15.	Interface de Usuario (Digital_GUIde_1&1, 2017).....	20
2.2.16.	IBM Watson (Ramirez Meson, 2016).....	23
2.2.17.	Watson Assistant (IBM Cloud Documents, 2018).....	25
2.2.18.	Bluemix (Stupia, Pamela, 2015).....	25
2.2.19.	Metodología de desarrollo de Software (Ian Sommerville, 2005).....	26
2.2.20.	ICONIX (CASTELLANOS, 2012).....	28
2.2.21.	Base de Datos (Volaya, Hub, 2016).....	31
2.2.22.	SQL (Plasencia Prado, 2017).....	32
2.2.23.	SQLite (Fileim, 2017).....	32
3.	MATERIAL Y MÉTODOS.....	36
3.1.	Material.....	36
3.1.1.	Población.....	36
3.1.2.	Muestra.....	36
3.2.	Método.....	36
3.2.1.	Tipo de investigación.....	36
3.2.2.	Diseño de investigación.....	36
3.2.3.	Variables de estudio y operacionalización.....	36
3.2.4.	Instrumentos de recolección de datos.....	38
3.2.5.	Procedimientos y análisis de datos.....	38
4.	RESULTADOS.....	39
4.1.	Proyectos similares a nivel nacional.....	39
4.2.	Proyectos similares a nivel internacional.....	41
4.3.	Desarrollo del Proyecto.....	42
4.3.1.	Organigrama.....	42
4.3.2.	Análisis y Requerimientos.....	42
4.3.3.	Análisis y Requisitos.....	43
4.3.3.1.	Modelo de Dominio.....	43
4.3.4.	Análisis y Diseño Preliminar.....	44
4.3.5.	Diseño.....	48
4.3.6.	Implementación.....	52
4.4.	Testeo y Retroalimentación:.....	73
5.	DISCUSIÓN DE RESULTADOS.....	74
5.1.	Contrastación de Hipótesis.....	74
5.1.1.	Grado de Satisfacción en el Diagnóstico.....	74

5.1.2. Nivel de Facilidad de Uso de la Aplicación	74
5.1.3. Hipótesis Estadística	74
5.1.4. Nivel de Significancia.....	75
6. CONCLUSIONES	84
7. RECOMENDACIONES.....	85
8. REFERENCIA BIBLIOGRÁFICAS.....	86
ANEXOS	89

ÍNDICE DE TABLAS

Tabla 1. Operacionalización de Variable Dependiente	37
Tabla 2. Operacionalización de Variable Independiente.....	38
Tabla 3. Instrumentos de Recolección de Datos.....	38
Tabla 4. Ventajas y Desventajas - Proyecto Innova School.....	39
Tabla 5. Ventajas y Desventajas - Proyecto ARTURITO del BCP	40
Tabla 6. Ventajas y Desventajas - CENTRUM Católica.....	41
Tabla 7. Ventajas y Desventajas - Proyecto Barrow Neurological Institute e IBM.....	41
Tabla 8. Ventajas y Desventajas - Caso Ripley.....	42
Tabla 9. Ficha de Caso de Uso: Registrar Usuario.....	45
Tabla 10. Ficha de Caso de Uso: Obtener Nueva Contraseña.....	45
Tabla 11. Ficha de Caso de Uso: Realizar Consulta.....	46
Tabla 16. Pruebas de Registro de Usuario.....	73
Tabla 17. Pruebas de Login de Usuario.....	73
Tabla 18. Pruebas de Actualizar Contraseña de Usuario.....	73
Tabla 19. Distribución de las Muestras	75
Tabla 20. Formato de Valoración.....	76
Tabla 21. Tabla de Ponderación de Valores Obtenidos.....	81
Tabla 22. Resultados de la Variable Dependiente en el Ítem 1	82
Tabla 23. Resultados de la variable independiente en el ítem 1	83
Tabla 24. Resultados finales de la variable independiente.....	83

ÍNDICE DE FIGURAS

Figura 1. Estructura de un Sistema.....	7
Figura 2. Línea de Tiempo hasta la era de la Tecnología.....	10
Figura 3. Estructura del Framework (Acens_Sinfony 3 Framework, 2014)	13
Figura 4. Estructura Interna de Android (Android Developer, 2018)	18
Figura 5. Metodología Clásica (Ian Sommerville, 2005)	27
Figura 6. Metodología Evolutiva (Ian Sommerville, 2005)	27
Figura 7. Organigrama de la Empresa	42
Figura 8. Modelo de Dominio	44
Figura 9. Diagrama de Casos de Uso	44
Figura 10. Diagrama de Robustez: Registrar Usuario.....	46
Figura 11. Diagrama de Robustez: Actualizar Contraseña.....	47
Figura 12. Diagrama de Robustez: Realizar Consulta.....	47
Figura 13. Diagrama de Secuencia: Registrar Usuario.....	48
Figura 14. Diagrama de Secuencia: Realizar Consulta	48
Figura 15. Diagrama de Secuencia: Actualizar Contraseña	49
Figura 16. Prototipo: Interface Login	49
Figura 17. Prototipo: Interface Actualizar Contraseña.....	50
Figura 18. Prototipo: Interface Registrar Usuario	50
Figura 19. Prototipo: Interface ChatBot	51
Figura 20. Diagrama de Clases.....	51
Figura 21. Resultado de Pregunta N° 1, Indicador de Satisfacción del Cliente	77
Figura 22. Resultado de Pregunta N° 2, Indicador de Satisfacción del Cliente	77
Figura 23. Resultado de Pregunta N° 3, Indicador de Satisfacción del Cliente	78
Figura 24. Resultado de Pregunta N° 4, Indicador de Satisfacción del Cliente	78
Figura 25. Resultado de Pregunta N° 5, Indicador de Satisfacción del Cliente	79
Figura 26. Resultado de Pregunta N° 1, Indicador de Nivel de Facilidad de Uso.	79
Figura 27. Resultado de Pregunta N° 2, Indicador de Nivel de Facilidad de Uso	80
Figura 28. Resultado de Pregunta N° 3, Indicador de Nivel de Facilidad de Uso	80
Figura 29. Resultado de Pregunta N° 4, Indicador de Nivel de Facilidad de Uso	80

1. INTRODUCCIÓN

1.1. Planteamiento del Problema

En los últimos años las comunicaciones ha sido una necesidad indispensable de los seres humanos en la búsqueda de erradicar estas limitaciones, aparecieron soluciones como Internet, y de este, un abanico de posibilidades que abre nuevas puertas a formas de comunicación antes no pensadas. Además de esto hoy en día Internet, y los dispositivos que permiten el acceso a la red, nos han facilitado la vida en distintos campos como en Educación, Política, Comercial, Social, etc.

Por otro lado, las empresas han tenido que asimilar estos cambios para mantenerse en el mercado competitivo y optar por el uso de estos medios para presentar su marca, dando a conocer con ellos sus productos y servicios, todo esto con la finalidad de captar nuevos clientes, identificar a potenciales usuarios y fortalecerla.

En el Perú. las aplicaciones móviles han incrementado la productividad de las empresas en un 63%, un 50% en la satisfacción del cliente y 13% en ventas, según estadísticas manejadas por Telefónica del Perú (GESTION, 2017) . algunos sectores como el rubro Bancario o Telefónico ya están innovando a través del uso de tecnologías como los sistemas cognitivos basado en inteligencia artificial (IA) para dar soluciones y brindar soporte de manera rápida a través de servicios de atención al cliente; el Banco de Crédito del Perú (BCP) por ejemplo, como parte de su estrategia de transformación digital, ha desarrollado una táctica de Inteligencia Artificial y Tecnología Cognitiva a fin de ofrecer a sus clientes mejores y más eficientes experiencias, según sostuvo Arturo Johnson, gerente de Área de Canales Alternativos del BCP, presentó Arturito BCP, el primer chatbot del sistema financiero peruano que atiende consultas financieras a través de Facebook Messenger (LA REPUBLICA, 2017).

Lo cierto es que, en el Perú, según informó IBM, ya hay gestándose alrededor de 20 proyectos que utilizan la inteligencia artificial como base para obtener mejoras en el manejo de los negocios, aunque el avance de estas soluciones en el país está dándose rápidamente. En la actualidad, se ha identificado un problema que se presenta y se desea combatir por medio de soluciones como las mencionadas: El tiempo que los usuario se toman para acudir a un centro de apoyo o soporte (de cualquier rubro) que diagnostique y/o de solución a los problemas con los que

puedan encontrarse; en el sector automotriz, rubro donde se ubica la empresa de enfoque: VECARS & TRUCKS S.A.C, además del problema mencionado se sabe que no se han desarrollado aplicaciones de ayuda o canales de atención online que brinden soporte en base a soluciones a las que el cliente pueda acceder en cualquier momento que lo requiera o sin tener que acudir directamente a un establecimiento.

1.2. Delimitación del Problema

- Delimitación Espacial: El proyecto se desarrollará e implantará en la ciudad de Trujillo, específicamente en la empresa Vecars & Trucks S.A.C.
- Delimitación Temporal: La duración del proyecto es de 4 a 6 meses.
- Delimitación del Universo: Clientes de VECARS & TRUCKS S.A.C.

1.3. Formulación del Problema

¿Cómo VECARS & TRUCKS S.A.C. podría brindar un servicio de diagnóstico más accesible y en el momento que el cliente lo necesite?

1.4. Formulación de la Hipótesis

El desarrollo una aplicación móvil que utiliza IBM Watson permite mejorar el diagnóstico de problemas automotrices en la empresa VECARS & TRUCKS S.A.C.

1.5. Objetivos de Estudio

1.5.1. Objetivo general

Desarrollar una aplicación móvil para el diagnóstico de problemas automotrices utilizando la herramienta IBM Watson para la empresa VECARS & TRUCKS S.A.C.

1.5.2. Objetivos específicos

- Recolectar y analizar información de proyectos similares en otros ámbitos para determinar las fortalezas y los puntos débiles del proyecto.
- Definir el alcance del sistema de acuerdo a la realidad problemática del entorno.
- Desarrollar un prototipo con el fin de evaluar la funcionalidad y viabilidad de la aplicación.
- Probar el prototipo de la aplicación para contrastar la hipótesis y determinar el éxito del proyecto.

1.6. Justificación del estudio

La importancia del presente proyecto se enfoca en el desarrollo de nuevos conocimientos y aplicaciones que sean soporte de las futuras investigaciones, además de fomentar la innovación y búsqueda de soluciones tecnológicas que afronten la problemática del momento.

A través de la solución inteligente el proyecto tiene como finalidad el desarrollo de un sistema cognitivo en base a herramientas de IBM que brinden soporte en la creación de un sistema que pueda interactuar con seres humanos, para poder diagnosticar y tratar problemáticas a través de bases de conocimiento y mantenerse aprendiendo ante cada interacción, además llegar a recopilar datos frente a cada iteración para su explotación y análisis.

Uno de los puntos sobresalientes es el desarrollo de la aplicación que se divide en las siguientes partes: la estructura, el flujo de comunicación y la construcción de la base de conocimiento utilizando la plataforma de tecnología cognitiva IBM Watson específicamente la herramienta que provee llamada Watson Assistant, por otro lado, el análisis, modelamiento, diseño e implementación de la aplicación móvil se hizo basándose la metodología ICONIX en la plataforma Android Studio versión 3.2.1, las partes mencionadas se traducen en conjunto como el proyecto de investigación planteado.

La conclusión de lo que se busca con este proyecto es lograr responder a la hipótesis planteada, además de cubrir las necesidades y brindar el soporte que los usuarios, clientes de la empresa de enfoque requieren. De esta manera el beneficio será un canal de ayuda que la empresa brinde a sus clientes y nuevos potenciales, abriéndole la puerta a un mercado más amplio y generando a su vez una ventaja competitiva que llegue a superar a la de otras entidades a través de una solución innovadora.

Como se mencionó al principio parte del trasfondo de la investigación es ser un soporte y fomentar la innovación e investigación a las futuras generaciones de la Universidad Privada Antenor Orrego.

1.7. Limitaciones del estudio

El presente proyecto de investigación se delimita al estudio del desarrollo de una solución para diagnóstico y tratamiento de posibles fallas automotrices, por ende, se realizará un prototipo de aplicación utilizando la herramienta Watson Assistant que provee la plataforma de IBM Watson y el marco de desarrollo ICONIX para el diseño, e implementación de la aplicación móvil cabe resaltar que esta contiene una base de datos para almacenar información de sus usuarios basada en SQLite; la aplicación estará disponible para plataformas Android, a lo mencionado se liga el tiempo de 4 meses aproximadamente para su desarrollo, el manejo de la infraestructura necesaria o en su defecto que permita su desarrollo, así como el acceso y disponibilidad a fuentes de datos como base del proyecto.

2. MARCO TEÓRICO

2.1. Antecedentes

2.1.1. Internacional

"Agentes virtuales con capacidades cognitivas utilizando IBM Watson" (Carrillo Calderón, 2017).

En la actualidad, los avances en la tecnología informática y la creciente globalización por medio de Internet y las redes sociales han obligado a que los comercios tradicionales luchan por digitalizarse, a la par que los comercios online traten de ser cada vez más personales y cercanos a los clientes. En esto consiste el comercio electrónico conversacional, una evolución del ecosistema del comercio electrónico.

Este Trabajo tiene como objetivo principal el diseño y desarrollo de un sistema conversacional, Watson AV, capaz de analizar e interpretar diversos tipos de datos, incluyendo texto no estructurado y audio para interactuar con una persona y proporcionar recomendaciones individualizadas mediante la comprensión de su personalidad, su tono y emoción.

"Propuesta de un Modelo de Proceso para Resolver

Vulnerabilidades de Seguridad en Infraestructura Utilizando Herramientas de Computación Cognitiva" (Darío Propato & Marisa Panizzi, 2017).

El objetivo de este trabajo de investigación consiste en el desarrollo de un modelo de proceso que permita resolver vulnerabilidades de Seguridad Informática utilizando herramientas de la computación cognitiva. Este servirá como guía a los profesionales de informática que se desempeñen en áreas de seguridad de la información. Los sistemas cognitivos pueden aprender de sus experiencias, encontrar correlaciones, crear hipótesis y recordar los resultados y aprender de ellos. Esta capacidad de los sistemas cognitivos tendrá una influencia crucial en la toma de decisiones por parte de múltiples interesados y la solución a construir tomará como base las ventajas de este tipo de sistemas. Se realizó una revisión sistemática de un par de modelos de procesos de software como por ejemplo MoProSoft y Métrica versión 3, para considerar de ellos los elementos propuestos y ser contemplados en la propuesta de solución

de esta investigación. Para llevar a cabo la validación del modelo propuesto se utilizará un caso real de una organización.

2.1.2. Nacional

"Modelo de implementación de tecnología cognitiva sobre intereses de consumo para el sector bancario en Cloud Computing" (Eyzaguirre Alberca & Placencia Roman, 2018).

El proyecto se desarrolla en base a la investigación sobre cloud computing; donde se incluye definiciones, servicios, despliegue, seguridad y metodologías utilizadas en modelos de implementaciones. Así mismo, se expone el benchmarking realizado para la elección de la plataforma PaaS, utilizado en la realización de la prueba de concepto del proyecto. Se evalúa puntos claves para la realización del cuadro comparativo, sobre la información obtenida de Gartner e IDC, con ello se tiene como objetivo justificar el uso de la plataforma seleccionada. Por otro lado, se utiliza metodologías tales como Project Management Body of Knowledge (PMBOK) para su gestión y la Metodología IBM CCRA 4.0 para el desarrollo de la aplicación sobre Cloud Computing, la cual consta de 5 fases: Understand Client, Define Client Requirements, Design Solution, Detail Design to define BOM, Define Roadmap y First Project, esta metodología nos brinda una mejor definición del alcance, requisitos y arquitectura. Con este modelo de implementación contamos con el análisis adecuado para obtener el perfil psicológico de los clientes a partir del análisis de los textos que publican en sus redes sociales, blog, correo electrónico, mensajes de texto, mensajes de twitter y mensajes en foros.

“Aplicaciones actuales de la inteligencia artificial y su uso con la tecnología IBM Watson” (Lepage Chumpitaz, 2016).

Una de las aplicaciones emergentes en lo ahora se denomina computación cognitiva, es la tecnología IBM Watson, que presentada en el 2011 en la televisión estadounidense en un concurso de televisión; ha continuado su desarrollo por parte de la Cía. Americana y ha formado un ecosistema de asociados, clientes y proyectos, poniendo ya a disposición de cualquier interesado, un conjunto de APIs (Interfases de programación de aplicaciones) para su uso en cualquier industria que lo desee. Se hará una descripción de

dicha tecnología IBM Watson. Presenta los estudios y tendencias de los expertos, sobre los efectos futuros de las aplicaciones de estas tecnologías, su impacto en la sociedad y; en la formación de los nuevos profesionales de tecnología.

2.1.3. Local

“Implementación de un Sistema de Información de análisis predictivo para la toma de decisiones en el proceso de Atención médica del Hospital Víctor Lazarte usando la Herramienta de Cognos BI de IBM” (Horna Gutiérrez & Rodríguez Marquina, 2016).

La tesis resalta la problemática encontrada en la Toma de Decisiones en el Proceso de Atención Medica en el hospital Víctor Lazarte Echegaray además explica los procedimientos para la implementación de un sistema de información de análisis predictivo para la toma de decisiones en donde nos basaremos en informaciones obtenidas directamente de la realidad.

2.2. Definiciones

2.2.1. Sistemas

Se denomina (Gastelum Lopez, 2007) como un conjunto de elementos dinámicamente relacionados entre sí que realizan una actividad para alcanzar un objetivo, operando sobre entradas (datos, energía o materia) y proveyendo salidas (información, energía o materia) procesadas y también interactúa con el medio entorno que lo rodea el cual influye considerable y significativamente en el comportamiento de este.

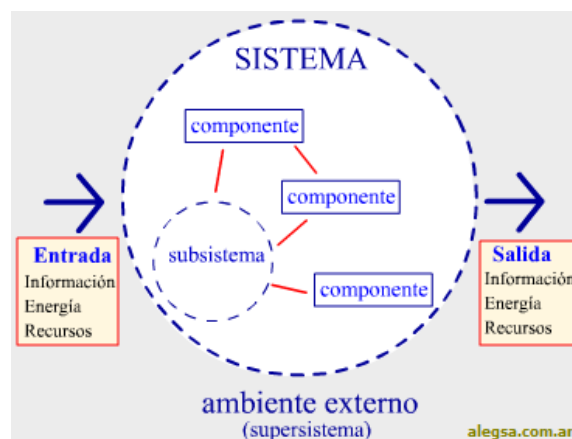


Figura 1. Estructura de un Sistema

Los sistemas pueden ser:

– **Sistema abierto:**

Relación permanente con su medio ambiente. Intercambia energía, materia, información. Interacción constante entre el sistema y el medio ambiente.

– **Sistema cerrado:**

Hay muy poco intercambio de energía, de materia, de información, etc., con el medio ambiente. Utiliza su reserva de energía potencial interna. Si no ocurre importación o exportación en ninguna de sus formas, como información, calor, materia física, etc. y por consiguiente sus componentes no se modifican. Insecto, el hombre, un grupo social. La familia, por tanto, la consideraremos un Sistema Abierto.

2.2.2. Cognición

La cognición se refiere al acto, acción o efecto de pensar, es el proceso mental de adquirir comprensión a través del pensamiento y las experiencias personales o compartidas.

Se define como "La adquisición sucesiva de estructuras lógicas cada vez más complejas, que subyacen a las distintas tareas y situaciones que el sujeto es capaz de ir resolviendo a medida que avanza en su desarrollo" (Fritz Piaget, 1868-1980).

Y para (Semionóvich Vygotski, 1989) "...el rasgo esencial de nuestra hipótesis es la noción de que los procesos evolutivos no coinciden con los procesos de aprendizaje. Por el contrario, el proceso evolutivo va a remolque del proceso de aprendizaje, es decir, el aprendizaje despierta una serie de desarrollos evolutivos internos capaces de operar solo cuando el niño está en interacción con las personas de su entorno y en cooperación con algún semejante"

2.2.3. Inteligencia

La palabra proviene del latín "intelligentia", que a su vez proviene de "intellegere", término compuesto de "inter", entre y "legere" leer, escoger, por lo que etimológicamente, inteligente es quien sabe escoger.

"Inteligencia, es la potencia intelectual, la facultad de conocer o de entender. El grado en que un individuo puede resolver satisfactoriamente una nueva

situación o un problema. La inteligencia está basada en el nivel de conocimientos individuales y en la habilidad de manipular y reformular apropiadamente los conocimientos en base a los datos que se proporcionan como requerimientos para resolver algún problema o situación" (Arauz, 1998). Según (Goleman, 1996), "La inteligencia es la capacidad para reconocer sentimientos propios y ajenos y la habilidad de manejarlos".

En otro punto de vista se dijo que "la inteligencia es un potencial biosociológico de procesamiento de información que se puede activar en uno o más marcos culturales para resolver problemas o crear productos que tienen valor para dichos marcos" (Howard, 2000)

2.2.4. Artificial

Se refiere a un adjetivo que se da a algo hecho por mano o arte del hombre y su ingenio, que no es natural falso (Real Academia de la Lengua Española, 2007)

Por otro lado, según (Oxford_Dictionary, 2016) se define como "Algo que ha sido hecho por el ser humano y no por la naturaleza "

2.2.5. Inteligencia Artificial:

El nombre fue surgió en 1956 por primera vez en una conferencia en el Dartmouth College, New Hampshire. Años más tarde fue que se tuvo una definición más concisa.

Según (Daniel Bourcier, 2003) dice que "la inteligencia artificial es una rama de la informática que intenta reproducir las funciones cognitivas humanas como el razonamiento, la memoria, el juicio o la decisión y, después, confiar una parte de esas facultades, que se consideramos signos de inteligencia, a los ordenadores".

Se puede decir que en la actualidad la inteligencia artificial es una disciplina fundamental en la ciencia y la tecnología, misma que tras el paso del tiempo ha creado una serie de conocimientos básicos que le permiten emular diversas capacidades del ser humano para exhibir comportamientos inteligentes. Debido a lo anterior se han desarrollado diversos sistemas que tienen por objetivo

perfeccionar las distintas capacidades del ser humano con el fin de la reproducción de las mismas.

2.2.6. Computación Cognitiva

Es la simulación del proceso de pensamiento humano mediante un modelo computarizado las características clave de los sistemas cognitivos son:

- a) Interactuar a través de lenguaje natural.
- b) Aprender digiriendo documentos.
- c) Hacer recomendaciones con niveles de confianza.

La computación cognitiva busca producir máquinas que aumenten las capacidades del ser humano en vez de máquinas que piensen por los humanos (Ramos, Herrera, & Ramirez, 2009)

El avance del tiempo ha dado como respuesta la llegada de los nuevos sistemas y tecnologías (véase **Era de la Tecnología, IBM**).

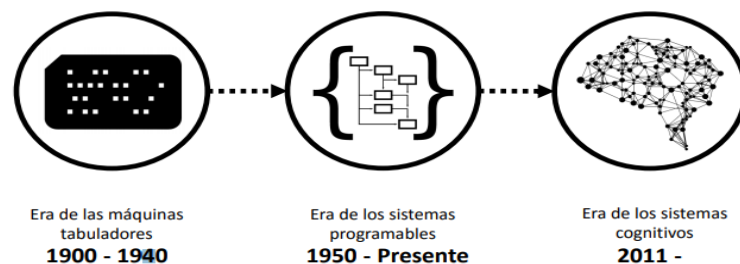


Figura 2. Línea de Tiempo hasta la era de la Tecnología

2.2.7. Software (Castellano Casas, 2001)

Es el conjunto de instrucciones y datos en formato binario almacenados en la memoria principal, que le indica a una computadora que debe hacer y cómo. es decir, el software dirige al hardware el software es la parte lógica del sistema informático.

2.2.8. Bot (Haj-Salej, 2017)

Es un software de inteligencia artificial desarrollado con la finalidad de resolver un conjunto de procesos y/o tareas por cuenta propia si apoyo de ningún ser humano.

2.2.8.1. Usos

Los principales usos de los bots son:

Rastrear información en la web (web crawler): con un script automático se capta, analiza y almacena información de un servidor web de forma automatizada y mucho más rápido de lo que lo haría una persona. Cada servidor puede tener un archivo llamado robots.txt, que contiene reglas para su rastreo.

Dar respuestas rápidas: Los bots de juegos o los de webs de subastas en línea ofrecen respuestas de manera mucho más rápida y efectiva que un humano.

Mantener conversaciones: Los bots conversacionales simulan una conversación humana al proveer respuestas automáticas a entradas hechas por el usuario.

Editar de manera automática: En sitios wiki, los bots realizan funciones rutinarias de edición, como corregir faltas de ortografía, mantener la coherencia de los enlaces, detectar y revertir las contribuciones vandálicas, etc.

Simular tráfico en Internet y las redes sociales: Los bots se utilizan para aumentar el número de seguidores de manera artificial, para generar artificialmente visitas a sitios web, para posicionar hashtags y temas para convertirlos en trending topics, o para influir positiva o negativamente en una conversación.

2.2.9. Chat (Vanegas, Marin, & Vargas, 2007)

Un chat es una conversación realizada por medios informáticos. La palabra chat es un anglicismo, usado para describir este tipo de conversación. Chatear es entonces el hecho de participar en este tipo de conversación.

Un chat tiene las siguientes características:

- Es sincrónico: esto quiere decir que la información es recibida por el destinatario inmediatamente ha sido enviada. Esto implica que los participantes deben estar conectados a la red en el mismo momento.
- Es basado en texto: los participantes escriben mensajes con sus teclados y estos son enviados a través de la red para luego ser

visualizados por los demás participantes en sus pantallas. La palabra chat es un término anglosajón que se deriva de chatter, término en inglés que significa "conversación o charla".

Los chats son conversaciones que se llevan a cabo al instante gracias al uso de un software conectado a una red de Internet y se diferencian por ser chats públicos, grupos de conversación en los cuales puede participar cualquier persona o, chats privados, que solo pueden participar usuarios autorizados.

El objetivo primordial de los chats es lograr que las personas se comuniquen e intercambien información al instante sin importar donde se localicen, acortando de esta manera las barreras de la distancia y el tiempo de manera mucho más económica.

2.2.10. Chatbox (Anda_Lucía_es_Digital, 2017)

Los chatbots o bots conversacionales son programas que simulan mantener una conversación con el usuario y que, en realidad, están ofreciendo una serie de respuestas automatizadas, previamente definidas, por parte de sus desarrolladores.

2.2.10.1. API.ai (MEDIUM_ADM, 2018)

Esta plataforma de inteligencia artificial está especializada en la creación de chatbots para plataformas como Slack, Facebook Messenger o canales de mensajería instantánea como Telegram, Whatsapp o Snapchat. API.ai fue recientemente adquirida por Google lo que la convierte en una de las aplicaciones de referencia para el desarrollo de chatbots que tienen como punto fuerte la interpretación del lenguaje natural. Dispone de una versión de prueba gratuita, antes de contratar sus servicios de pago.

2.2.10.2. Botsify (Suja, 2018)

Botsify es otra de las plataformas para crear chatbots para Facebook Messenger o servicios de mensajería sin tener conocimientos avanzados de programación. Solo hace falta que te registres como usuario y accedas a la versión gratuita para comenzar a diseñar tus chatbots que puedes integrar en la página de tu empresa en Facebook para automatizar las respuestas a tus clientes.

2.2.10.3. Chatfuel (Developers_Society_Chatfuel, 2015)

Es otro servicio para crear chatbots de forma intuitiva y sin apenas conocimientos de programación. Está muy centrada en el diseño y la creación de bots conversacionales que puedes integrar en tu página de fans de Facebook y que responderá las dudas de los usuarios a través de Facebook Messenger. Dispone de una versión gratuita que, como en los casos anteriores, te permite disfrutar hasta un tope de mensajes.

2.2.10.4. Motion.AI (Hidalgo, Francisco José, 2016)

Es otra plataforma para la creación de chatbots sin tener conocimientos de programación. Al igual que las anteriores está centrada en dos plataformas sociales para la creación de bot conversacionales para empresas: Facebook Messenger y Slack. Te ofrece un servicio gratuito, limitado a un número máximo de mensajes, a partir del que se activaría la versión de pago.

2.2.11. Framework (COMPANYRED, 2016)

Un framework no es ningún software ni herramienta que se ejecuta y que nos ofrece una interfaz gráfica desde la que trabajar, sino que es un conjunto de archivos y directorios que facilitan la creación de aplicaciones, ya que incorporan funcionalidades ya desarrolladas y probadas, implementadas en un determinado lenguaje de programación.

En la siguiente imagen se puede apreciar la estructura de directorios correspondiente a un proyecto desarrollado con Symfony 2.

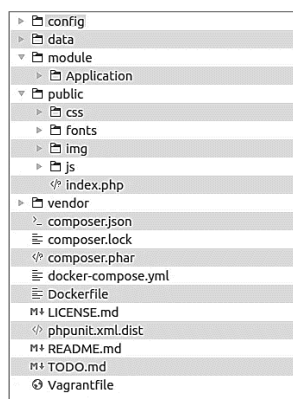


Figura 3. Estructura del Framework (Acens_Sinfony 3 Framework, 2014)

El objetivo principal de todo framework es facilitar las cosas a la hora de desarrollar una aplicación, haciendo que nos centremos en el verdadero problema y nos olvidemos de implementar funcionalidades que son de uso común como puede ser el registro de un usuario, establecer conexión con la base de datos, manejo de sesiones de usuario o el almacenamiento en base de datos de contenido cacheado.

2.2.11.1. Características

A continuación, os dejamos algunas características que suelen incluir todos los frameworks existentes.

- Abstracción de URLs y sesiones. No es necesario manejar directamente las URLs ni las sesiones, ya que el framework se encarga de hacerlo.
- Acceso a datos. Incluyen herramientas e interfaces necesarias para comunicarse con bases de datos, independientemente del tipo que estemos utilizando.
- Uso de controladores. Suelen implementar una serie de controladores para la gestión de los eventos y peticiones realizadas a la aplicación.
- Autenticación y control de acceso. Incluyen mecanismos para la identificación de usuarios mediante el uso de login y password.
- Internalización. Son mecanismos para poder mostrar la aplicación en todos aquellos idiomas que consideremos oportunos.

2.2.11.2. Ejemplos de Frameworks

A continuación, os dejamos un listado de frameworks que nos podemos encontrar.

- **Ruby on Rails.** Framework MVC basado en Ruby orientado al desarrollo de aplicaciones web
- **CodeIgniter.** Framework basado en PHP liviano y rápido
- **Django.** Un framework para Python capaz de crear diseños muy limpios.
- **Zend Framework.** Es un framework de código abierto en PHP para desarrollar aplicaciones web y servicios web con PHP 5.

- **Symfony.** Completo framework en PHP diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador .
- **Yii.** Framework en PHP basado en componentes
- **Struts.** Herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition) .
- **ASP .NET.** Es un framework para aplicaciones web desarrollado y comercializado por Microsoft.

2.2.12. Android (GITBOOKS_AJGALLEGO, 2010)

Android es un sistema operativo de código abierto para dispositivos móviles, se programa principalmente en Java y su núcleo está basado en Linux.

2.2.12.1. Estructura

Android está basado en Linux. Para ser más específicos, hablamos del kernel. Android utiliza como base el kernel de Linux. Esto no significa que por estar basado en el algo que se desarrolló en Linux funcione para Android, por ejemplo, Android no tiene soporte glibc.

2.2.12.2. Composición

- Núcleo basado en el de Linux para el manejo de memoria, procesos y hardware. (Se trata de un branch, de manera que las mejoras introducidas no se incorporan en el desarrollo del núcleo de GNU/Linux).
- Bibliotecas open source para el desarrollo de aplicaciones, incluyendo SQLite, WebKit, OpenGL y manejador de medios.
- Entorno de ejecución para las aplicaciones Android. La máquina virtual Dalvik y las bibliotecas específicas dan a las aplicaciones funcionalidades específicas de Android.
- Un framework de desarrollo que pone a disposición de las aplicaciones los servicios del sistema como el manejador de ventanas, de localización, proveedores de contenidos, sensores y telefonía.
 - SDK (kit de desarrollo de software) que incluye herramientas, plug-in para Eclipse, emulador, ejemplos y documentación.

- Interfaz de usuario útil para pantallas táctiles y otros tipos de dispositivos de entrada, como por ejemplo, teclado y trackball.
- Aplicaciones preinstaladas que hacen que el sistema operativo sea útil para el usuario desde el primer momento. Cabe destacar que cuenta con las últimas versiones de Flash Player.
- Muy importante es la existencia del Android Market, y más todavía la presencia de una comunidad de desarrolladores que suben ahí aplicaciones, tanto de pago como gratuitas. De cara al usuario, el verdadero valor del sistema operativo está en las aplicaciones que se puede instalar.

2.2.12.3. Android SDK (GITBOOKS_AJGALLEGO, 2010)

El SDK de android incluye numerosas y completas API's para facilitar el desarrollo. Algunas de las características más relevantes son:

- Licencias, distribución y desarrollo gratuitos, tampoco hay procesos de aprobación del software.
- Acceso al hardware de WiFi, GPS, Bluetooth y telefonía, permitiendo realizar y recibir llamadas y SMS.
- Control completo de multimedia, incluyendo la cámara y el micrófono.
- APIs para los sensores: acelerómetros y brújula.
- Mensajes entre procesos (IPC).
- Almacenes de datos compartidos, SQLite, acceso a SD Card.
- Aplicaciones y procesos en segundo plano.
- Widgets para la pantalla de inicio (escritorio).
- Integración de los resultados de búsqueda de la aplicación con los del sistema.
- Uso de mapas y sus controles desde las aplicaciones.
- Aceleración gráfica por hardware, incluyendo OpenGL ES 2.0 para los 3D.

Muchas de estas características ya están, de una manera o de otra, para los SDK de otras plataformas de desarrollo móvil. Las que diferencian a Android del resto son:

- Controles de Google Maps en nuestras aplicaciones
- Procesos y servicios en segundo plano
- Proveedores de contenidos compartidos y comunicación entre procesos
- No diferencia entre aplicaciones nativas y de terceros, todas se crean igual, con el mismo aspecto, y con las mismas posibilidades de usar el hardware y las APIs.
- Widgets de escritorio.

2.2.12.4. Capas

El núcleo de Linux es la capa encargada de los controladores (drivers) del hardware, los procesos, la memoria, seguridad, red, y gestión de energía. Es la capa que abstrae el resto de las capas del hardware.

El Android run time es lo que hace a Android diferente de una distribución de Linux embebido. Está compuesto por las librerías "core" (núcleo) y por Dalvik, la máquina virtual de Java, basada en registros que cuenta con el núcleo de Linux para la gestión de hilos y para el manejo de memoria a bajo nivel.

El framework de aplicaciones está compuesto por las clases que se utilizan para crear aplicaciones Android: actividades, servicios, views, proveedores de contenidos, etc.

Finalmente, la capa de aplicaciones está compuesta por las aplicaciones nativas y por las de terceros, así como las de desarrollo.

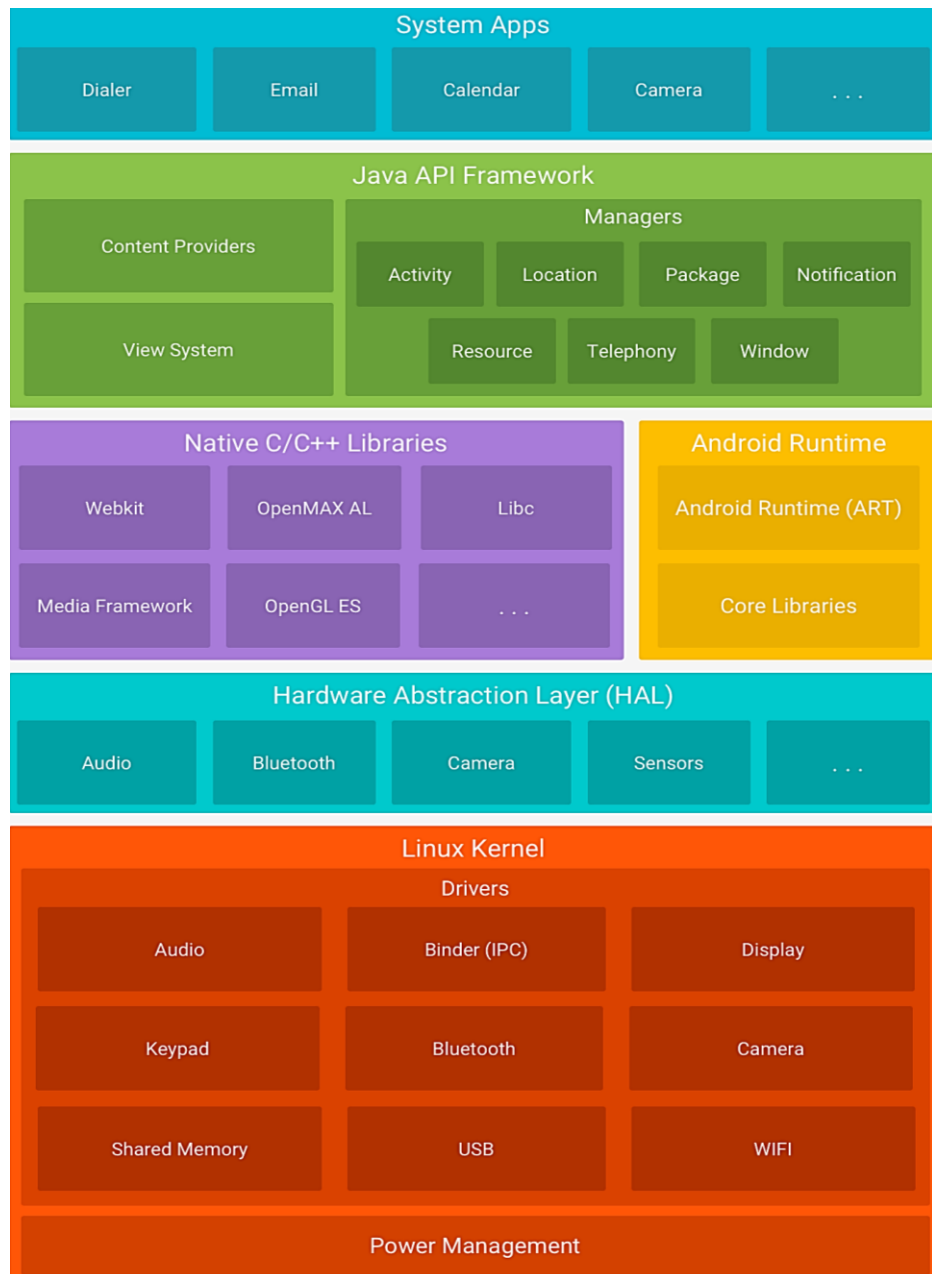


Figura 4. Estructura Interna de Android (*Android Developer, 2018*)

Las clases más importantes para el desarrollo de aplicaciones en Android son las siguientes:

- **ActivityManager:** Controla el ciclo de vida de las actividades.
- **View:** Se usan para construir interfaces en las actividades.
- **NotificationManager:** Mecanismo no intrusivo para mostrar avisos al usuario.

- **ContentProvider:** Permiten intercambiar datos de una manera estandarizada.
- **Resource Manager:** permite usar en la aplicación recursos que no forman parte del código, como XML, strings, recursos gráficos, audio, vídeo, etc.

2.2.13. API (Merino, 2014)

Una API (siglas de ‘Application Programming Interface’) es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

Las API pueden servir para comunicarse con el sistema operativo (WinAPI), con bases de datos (DBMS) o con protocolos de comunicaciones (Jabber/XMPP). En los últimos años, por supuesto, se han sumado múltiples redes sociales (Twitter, Facebook, Youtube, Flickr, LinkedIn, etc) y otras plataformas online (Google Maps, WordPress...), lo que ha convertido el social media marketing es algo más sencillo, más rastreable y, por tanto, más rentable.

2.2.14. Interface (Wikibooks_Anon, 2016).

Una interfaz de usuario es todo cuadro físico o representado en una pantalla, que presenta controles para accionar un dispositivo y pilotos que representan el estado de funcionamiento del sistema.

En caso de sistemas de gestión, habrá numerosos campos de introducción de datos alfanuméricos y, en caso de sistemas físicos controlables "en tiempo real", se dispondrá de controles rotativos o deslizantes autoindicativos de su posición, así como de indicadores gráficos de tipo "barras" o "de aguja" además de simples pilotos "encendido/apagado".

Las interfaces físicas de los equipos electrónicos (tanto domésticos como profesionales) suelen ser electromecánicas (botones que se manipulan mediante palanquitas o botones giratorios o deslizantes) aunque la progresiva "digitalización" está sustituyendolos cada vez más por pulsadores "arriba/abajo" que controlan un regulador interno electrónico.

2.2.15. Interface de Usuario (Digital_GUIde_1&1, 2017)

Una interfaz de usuario, en inglés user interface, a menudo conocida por la abreviatura UI, hace referencia a la interfaz con la que las personas interactúan con las máquinas. Se trata de la interfaz que permite usar un ordenador para realizar un pedido en una tienda online o para acceder a una app a través de un smartphone. Para ello, esta cuenta con los elementos de control de una interfaz, que son visibles para el usuario o a través de los que puede efectuar alguna acción. En ello se incluyen desde simples líneas de comandos basadas en texto hasta interfaces gráficas de usuario con un diseño más complejo. Al mismo tiempo, una UI también permite que el ordenador envíe un feedback al usuario, de modo que este pueda dar cuenta de si su acción se ha llevado a cabo con éxito.

La interfaz de usuario está íntimamente relacionada con la facilidad de uso de un software o de una página web. Hace tiempo que el objetivo ya no solo es conseguir una interfaz de usuario útil, sino que el aspecto estético también juega un papel importante. Por consiguiente, la interfaz de usuario es importante para una buena experiencia de usuario, es decir, para la experiencia que un usuario tiene grosso modo con una página web o un software. A este respecto, el objetivo de los diseñadores web es sentar las bases de una buena experiencia de usuario con una user interface intuitiva, lo que funciona especialmente bien a través de una interfaz gráfica de usuario. Sin embargo, también existen otros tipos de UI innovadoras que también facilitan la interacción con los ordenadores y con otros dispositivos.

2.2.15.1. Tipos de Interface (Digital_GUIde_1&1, 2017)

Desde que se inventaron los ordenadores, cada vez se han desarrollado métodos más avanzados de interacción persona-ordenador (IPO), por lo que hoy en día existen numerosas interfaces de usuario diferentes. Al comienzo de este desarrollo ya se podía hablar de las sencillas Command Line Interfaces (CLI), aunque las numerosas innovaciones que se realizaron dieron lugar a una manejabilidad más sencilla o directa, de modo que desde hace tiempo también se aplican las Natural User

Interfaces (NUI). Así, mediante la medición y traducción de ondas cerebrales es posible —y se sigue investigando— incluso llevar a cabo el control de los dispositivos. A continuación te presentamos las diferentes UI y sus características.

- **Command Line Interfaces (CLI)** (Digital_GUIde_1&1, 2017)

En el principio no solo era la Tierra, sino también la UI vacía y desolada. Tan solo una línea de comandos y un prompt para mostrar la posición actual o un mensaje cubrieron la oscura pantalla, por ejemplo en el caso de los primeros ordenadores MS-DOS. Con una interfaz de líneas de comandos o Command Line Interface, en la cual se tecleaban unos comandos específicos, era posible establecer la comunicación con el ordenador. Tras el procesamiento de la petición, el ordenador mostraba el resultado en formato de texto. Esta modalidad de interfaz de usuario ya se ha quedado obsoleta, pero se utiliza para cmd.exe en los sistemas operativos Windows, en los que la sintaxis está ampliamente orientada a los antepasados de DOS. El control mediante una CLI se lleva a cabo a través del teclado y sin necesidad de usar el ratón.

- **Text User Interface (TUI)** (Digital_GUIde_1&1, 2017)

Las Text User Interfaces son algo más cómodas y, en ellas, la interacción con el ordenador también tiene lugar por medio del teclado. Las TUI marcan el paso de las meras interfaces de líneas de comando a interfaces gráficas de usuario. El término se estableció con posterioridad al desarrollo de las interfaces gráficas de usuario, con el objetivo de poder diferenciar de un modo abstracto a las Text User Interfaces de las CLI y de las interfaces gráficas de usuario. Así, el término se convierte en un retrónimo.

La diferencia con las CLI consiste en que la interfaz utiliza la pantalla como una superficie plana y no orientada a las líneas. Sin embargo, la interfaz sigue ejecutándose, tal y como revela su nombre, en modo texto. Los programadores pueden recurrir normalmente a 256

caracteres y hoy en día, las TUI se utilizan, por ejemplo, en gestores de arranque o en programas BIOS Setup.

- **Graphical User Interfaces (GUI)** (Digital_GUIde_1&1, 2017)

Las interfaces gráficas de usuario son, en la actualidad, el estándar más establecido. Este software se aplica con elementos gráficos de control e imágenes simbólicas que toman como referencia de forma creativa a objetos del “mundo real”. En este sentido, los usuarios suelen recurrir al ratón y al teclado como dispositivos de control, pero cada vez es más común que entren en juego pantallas táctiles que funcionan mediante toques. A través de lo que se conoce como interfaz gráfica de usuario, los iconos también se introducen en el mundo digital, al igual que el escritorio, las ventanas o la papelera. Con el ratón, que sirve para desplazarse por la pantalla, se pueden seleccionar los elementos deseados y abrir con un ligero clic.

El diseño gráfico se basa, en la mayoría de los casos, en el diseño de oficina tradicional. Como consecuencia, todos los elementos pueden identificarse de forma sencilla y su manejo resulta más intuitivo a como lo es a través de la lista de comandos de una Command Line Interface. Así, los usuarios con menos experiencia pueden entender las funciones de los iconos con rapidez (ni la papelera ni el símbolo de la carta son símbolos del programa de correo que necesiten explicaciones adicionales para su comprensión). Este tipo de simbología se ha convertido en un factor imprescindible tanto para las user interfaces como para las GUI. De este modo, no hay apenas ningún programa cuyos iconos no remitan a objetos del mundo real tales como, por ejemplo, el pincel, el bolígrafo o la goma de borrar.

- **Voice User Interface (VUI)** (Digital_GUIde_1&1, 2017)

Aun cuando las interfaces gráficas de usuario están presentes en cualquier ámbito, el desarrollo de interfaces de usuario con GUI todavía no ha llegado a su fin: la Voice User Interface también permite la interacción con los ordenadores por medio del control por voz. Cada sistema operativo actual ofrece este tipo de UI de una forma o de otra:

Apple ha creado Siri, Amazon introdujo a Alexa, Microsoft incluye a Cortana en las versiones actuales de Windows y la búsqueda de Google también puede llevarse a cabo mediante voz siempre y cuando se cuente con el hardware correspondiente. De esta manera, puesto que se puede acceder a las aplicaciones por medio de llamadas o lo que se transmite a viva voz también se escribe de manera automática, los usuarios pueden trabajar de un modo más efectivo. Otra de las ventajas es que el control por voz garantiza una mayor accesibilidad.

- **Natural User Interface (NUI)** (Digital_GUIde_1&1, 2017)

La comunicación con el ordenador se establece de un modo especialmente intuitivo a través de una Natural User Interface, que puede ser contemplada como una evolución de la interfaz gráfica de usuario y de la Voice User Interface. En este sentido, la interfaz de usuario interactiva reacciona tanto a los gestos como a los movimientos y al lenguaje. Asimismo, también es capaz de reconocer rostros y objetos. Los diferentes sensores, cámaras y micrófonos permiten una comunicación versátil con el ordenador. Muchos de los smartphones y tablets actuales apuestan por tecnologías NUI, al igual que ocurre con las consolas Wii de Nintendo, que utilizan el reconocimiento de gestos, rostros e idiomas de un modo lúdico.

Microsoft causó furor en el año 2011 con la presentación de un nuevo proyecto: bajo el nombre de OmniTouch se desarrolló una tecnología que permite proyectar pantallas táctiles en cualquier tipo de interfaz. Omnitouch se basa en Kinect, el control de movimiento de Microsoft, que fue instaurado en el mercado para Xbox 360 como competencia de la Wii de Nintendo. Asimismo, también entra en juego un proyector láser y una cámara especial. Esta forma especial de Natural User Interfaces también se ha desarrollado para los dispositivos móviles, en cuyo caso se debe colocar el hardware sobre el hombro.

2.2.16. IBM Watson (Ramirez Meson, 2016)

IBM Watson es una plataforma de inteligencia artificial para los negocios.

Cuando hablamos de ‘cognitivo’, nos referimos a la capacidad de procesar información, aprender, razonar, memorizar, resolver problemas y tomar decisiones. Y ‘computación cognitiva’ cuando las máquinas imitan esas funciones cognitivas para hacerlas inteligentes.

En publicaciones como Qué es la industria 4.0 y qué beneficios tiene e IoT aplicado a la industria hemos visto cómo los avances tecnológicos de los últimos años hacen posible entornos de fabricación inteligentes, una importante mejora en la productividad, procesos productivos más óptimos y comunicaciones más robustas y ágiles para una toma de decisiones más eficiente.

En el caso de IBM Watson, esta tecnología pionera puede entender todas las formas de datos, interactuar naturalmente con las personas, y aprender y razonar. Y toda la información está siempre disponible en la nube. La ‘inteligencia’ de Watson se basa en 4 pilares:

- **Entender**

Esta herramienta permite analizar e interpretar todo tipo de datos, tanto estructurados (bases de datos) como no estructurados (textos, e-mails, imágenes, audio, vídeo, tweets, mensajes) ya que es capaz de leer y entender el lenguaje natural.

- **Razonar**

La tecnología de IBM Watson es capaz de comprender rasgos de personalidad, el tono o las emociones para proporcionar recomendaciones personalizadas a usuarios. Y también es capaz de establecer la secuencia lógica de por qué ha alcanzado un determinado razonamiento.

- **Aprender**

El software es capaz de utilizar su propia experiencia (machine learning) para crear aprendizaje y aplicarlo en las aplicaciones y sistemas de la organización.

- **Interactuar**

También permite participar y crear conversaciones y diálogos con usuarios, por ejemplo. Una de las principales innovaciones que introduce esta tecnología es que no solo es capaz de pensar, sino que es capaz de hacerlo como las personas e interactuar como una de ellas.

2.2.17. Watson Assistant (IBM Cloud Documents, 2018)

Es una herramienta de la plataforma IBM Watson, Watson Assistant, es un servicio que combina la inteligencia artificial, cloud y el Internet de las cosas para ayudar a las empresas a mejorar la experiencia que ofrecen a sus clientes. A diferencia de Alexa, Siri o Google Assistant, Watson Assistant no operará con el mismo nombre. Por el contrario, el plan de IBM es convertirlo en una solución personalizada para diferentes empresas, que podrán etiquetar en blanco esta tecnología.

Asimismo, como ya han comprobado algunas marcas que se han unido al proyecto, Watson Assistant está pensado para integrarse en distintas ubicaciones, no solo en el hogar: un automóvil, una habitación de hotel, una tienda minorista o una sala de conferencias. De esta manera, ofrece una mayor conveniencia para el usuario según el contexto.

Así, el conocimiento del usuario se combina con los factores contextuales (como la localización o el momento del día) para anticipar las necesidades del cliente y hacer recomendaciones que mejoren, consecuentemente y progresivamente, su experiencia.

2.2.18. Bluemix (Stupia, Pamela, 2015)

IBM presentó recientemente su nueva plataforma Cloud, basada en open standards para construir, gestionar y ejecutar aplicaciones de todo tipo, desde web, pasando por mobile y big data, hasta dispositivos inteligentes. En la nota le contamos, a través de un video de menos de 2 minutos, todo lo que necesita saber sobre Bluemix de IBM.

BlueMix es un entorno PaaS (Platform-as-a-Service) donde se pueden desplegar aplicaciones, realizadas en diferentes lenguajes de programación: Java, node.js, ruby, y que pueden consumir servicios tanto tecnológicos –

bases de datos relacionales, bases de datos noSQL, analítica de negocio, servicios para aplicaciones de móviles, SSO – como no tecnológicos – un localizador de código postal basado en una dirección.

2.2.19. Metodología de desarrollo de Software (Ian Sommerville, 2005)

En el desarrollo de software, una metodología hace cierto énfasis al entorno en el cual se plantea y estructura el desarrollo de un sistema. Como lo mencioné al principio, existen una gran cantidad de metodologías de la programación que se han utilizado desde los tiempos atrás y que con el paso del tiempo han ido evolucionando. Esto se debe principalmente a que no todos los sistemas de la información, son compatibles con todas las metodologías, pues el ciclo de vida del software puede ser variable. Por esta razón, es importante que dependiendo del tipo de software que se vaya a desarrollar, se identifique la metodología para el diseño de software idónea.

Una Metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Regularmente este tipo de metodología, tienen la necesidad de venir documentadas, para que los programadores que estarán dentro de la planeación del proyecto, comprendan perfectamente la metodología y en algunos casos el ciclo de vida del software que se pretende seguir.

2.2.19.1. Objetivos

Los Objetivos de las Metodologías de Desarrollo de Sistemas de Información son:

- Definir actividades a llevarse a cabo en un Proyecto de S.I.
- Unificar criterios en la organización para el desarrollo de S.I.
- Proporcionar puntos de control y revisión
- Asegurar la uniformidad y calidad tanto del desarrollo como del sistema en sí
- Satisfacer las necesidades de los usuarios del sistema
- Conseguir un mayor nivel de rendimiento y eficiencia del personal asignado al desarrollo
- Ajustarse a los plazos y costos previstos en la planificación
- Generar de forma adecuada la documentación asociada a los sistemas

- Facilitar el mantenimiento posterior de los sistemas.

2.2.19.2. Tipos de Metodologías y enfoques de Desarrollo de Software (Ian Sommerville, 2005)

2.2.19.2.1. Estructurada

- Se maneja como proyecto
- Gran volumen de datos y transacciones
- Abarca varias áreas organizativas de la empresa
- Tiempo de desarrollo largo
- Requiere que se cumplan todas las etapas, para poder cumplir las siguientes (progresión lineal y secuencial de una fase a la otra)

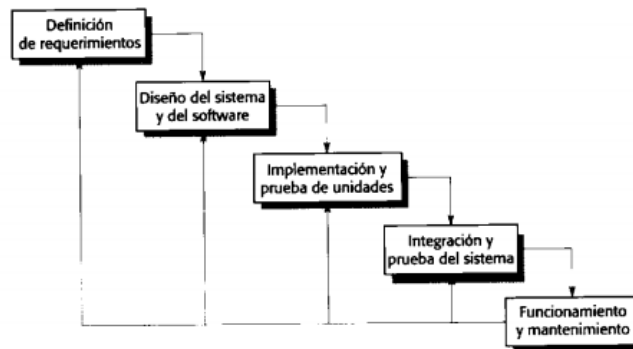


Figura 5. Metodología Clásica (Ian Sommerville, 2005)

2.2.19.2.2. Evolutiva-Incremental

- Se deriva de la estructurada
- Permite seguir secuencias ascendentes o descendentes en las etapas del desarrollo
- Permite cumplir etapas o fases en paralelo, por lo que es más flexible que la estructurada

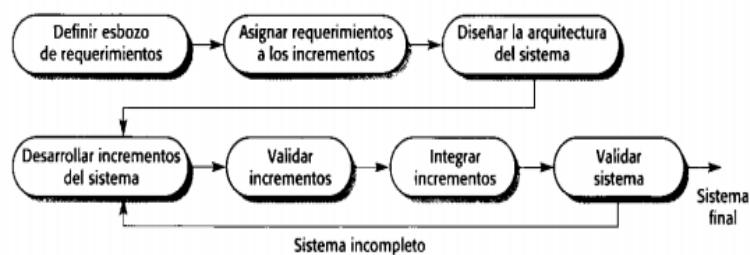


Figura 6. Metodología Evolutiva (Ian Sommerville, 2005)

2.2.19.2.3. Prototipos (CASTELLANOS, 2012)

- Desarrolla modelo en escala del sistema propuesto (sistema objetivo)
- En otras palabras, desarrolla un “demo”, que, de ser aprobado por el usuario, se culmina con todas las funcionalidades.
- Se prueba y refina hasta que usuarios conformes
- Identifica necesidades de información
- Maneja principales procedimientos orientados a transacciones
- Produce informes críticos
- Permite consultas rudimentarias
- Complementa especificaciones imprecisas y/o incompletas de los usuarios.

2.2.19.2.4. Orientada a Objetos (CASTELLANOS, 2012)

- No modela la realidad, sino la forma en que las personas comprenden y procesan la realidad
- Es un proceso ascendente basado en una abstracción de clases en aumento
- Se basa en identificación de objetos, definición y organización de librerías de clases, y creación de macros para aplicaciones específicas
- Utiliza menor cantidad de código
- Es más reutilizable

2.2.20. ICONIX (CASTELLANOS, 2012)

Es una metodología de desarrollo de software, basada en la complejidad de análisis de la metodología RUP (Rational Unified Processes) y la practicidad para desarrollar de la metodología XP (Extreme Programming).

Unifica un conjunto de métodos de orientación a objetos, con el objetivo de abarcar todo el ciclo de vida de un proyecto. Se considera un enfoque “minimalista”, ya que comprende el conjunto mínimo de medidas que son necesarias para el proyecto de desarrollo.

Fue elaborado por Doug Rosenberg, y Kendall Scott a partir de una síntesis del proceso unificado de los “3 amigos” Booch, Rumbaugh y Jacobson. Es una metodología de desarrollo de software basada en UML.

2.2.20.1. Características

- **Iterativo e Incremental:** Ocurren varias iteraciones entre el desarrollo del modelo del dominio y los casos de uso. El modelo estático es incremental
- **Trazabilidad:** es la capacidad de seguir una relación entre los diferentes artefactos producidos, por lo que cada paso está referenciado por algún requisito.
- **Dinámica del UML:** ofrece un uso dinámico del UML, como los diagramas de caso de uso, diagramas de secuencia y de colaboración.
- El ciclo de vida incremental consiste en desarrollar por partes el producto de manera que puedas integrarlas funcionalmente.
- **Ciclo de vida Iterativo:** En cada ciclo de iteración se revisa y mejora el producto.

2.2.20.2. Fundamentos (CASTELLANOS, 2012)

- Tiene que ser lo suficientemente flexible como para adaptarse a diferentes estilos y tipos de problemas.
- Hay que apoyar la forma de trabajo del personal (incluidos los prototipos y desarrollo iterativo / incremental).
- Sirve como una guía para los menos experimentados
- Expone los productos anteriores al código de manera estándar y comprensible.

2.2.20.3. Fases (CASTELLANOS, 2012)

- a) **Revisión de los requisitos/ Análisis de Requisitos:** Identificar en el mundo real, los objetos y todas las relaciones de agregación y generalización entre ellos. Se deben analizar todos los requisitos formaran parte del sistema y con estos construir el diagrama de clases,

que representa las agrupaciones funcionales que estructuran el sistema en desarrollo.

Para esta fase se utilizan 3 herramientas

- **Modelo de Dominio:** esto se refiere a identificar objetos y cosas del mundo real que intervienen con nuestro sistema. (Estático)
- **Modelo de Casos de Uso:** describe las acciones o el comportamiento que un usuario realiza dentro del sistema. Comprende de actores, casos de uso y el sistema.
- **Prototipo de Interfaz de Usuario:** implica la creación de un modelo o modelos operativos del trabajo de un sistema, en el que analistas y clientes deben estar de acuerdo. (Dinámico/ los usuarios se hacen participantes activos en el desarrollo)

b) **Revisión del diseño preliminar /Análisis y Diseño Preliminar:** En esta fase a partir de cada caso de uso se obtendrán una ficha de caso de uso, (la cual no pertenece a UML) , está formada por un nombre, una descripción, una precondición que debe cumplirse antes de iniciarse, una postcondición que debe cumplirse al terminar si termina correctamente.

- **Diagrama de Robustez:** Un diagrama de robustez es un híbrido entre un Diagrama de Clases y un Diagrama de Actividades. Es una herramienta que nos permite capturar el Qué hacer y a partir de eso él Como hacerlo.

Facilita el reconocimiento de objetos y hace más sencilla la lectura del sistema. Ayuda a identificar los objetos que participan en cada caso de uso.

El diagrama de Robustez se divide en:

- **Objetos fronterizos:** usado por los actores para comunicarse con el sistema.
 - **Objetos entidad:** son objetos del modelo del dominio.
 - **Objetos de Control:** es la unión entre la interfaz y los objetos de entidad.
- **Diagrama de Clases:** describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos

c) **Revisión crítica del diseño/Diseño:** En esta fase se reconocen todos los elementos que forman parte de nuestro sistema.

- **Diagramas de Secuencia:** muestra los métodos que llevarán las clases de nuestro sistema. Muestra todos los cursos alternos que pueden tomar todos nuestros casos de uso.

Se debe terminar el modelo estático, añadiendo los detalles del diseño en el diagrama de clases.

d) **Implementación:** En esta fase a partir del buen diseño logrado se creará el software; que posteriormente se entregará.

La metodología ICONIX, es una combinación entre la RUP y XP; está basada en el desarrollo de sistemas a partir del análisis y la documentación.

Esta metodología se busca tener una retroactividad con el cliente, en la mitad de los procedimientos, comenzando con un prototipo en donde el analista y el cliente definirán pantallas, funcionalidades, en si lo que se espera obtener del programa.

Se definirán los modelos de casos de uso, de secuencia y de robustez, con la finalidad de conseguir un buen sistema.

2.2.21. Base de Datos (Volaya, Hub, 2016)

Una base de datos es una aplicación independiente que almacena una colección de datos. Así que podemos decir que se trata de una colección de información organizada por campos, registros y archivos, de manera que se pueda seleccionar rápidamente los fragmentos de datos que se necesiten.

2.2.21.1. Base de Datos Relacional (Volaya, Hub, 2016)

Una base de datos es relacional cuando esta cumple con el modelo relacional, que se refiere a la relación que existe entre las distintas entidades o tablas de la base. También conocidas como sistemas de gestión de bases de datos relacionales (RDBMS), las cuales nos permiten almacenar y gestionar gran cantidad de datos. Los datos se almacenan en

diferentes tablas y las relaciones se establecen usando claves primarias u otras llaves conocidas como claves externas o foráneas.

Existen un sinnúmero de sistemas de gestión de bases de datos relacionales y cada una de ellas posee una forma diferente de manejar su data, algunos ejemplos de RDBMS son: Oracle, MySQL, SQL Server, entre otras.

2.2.22. SQL (Plasencia Prado, 2017)

SQL (Structured Query Language) es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas, gracias a la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla. Las consultas toman la forma de un lenguaje de comandos que permite seleccionar, insertar, actualizar, averiguar la ubicación de los datos, y más.

SQL es un lenguaje declarativo estándar internacional de comunicación dentro de las bases de datos que nos permite a todos el acceso y manipulación de datos en una base de datos, y además se puede integrar a lenguajes de programación, por ejemplo ASP o PHP, y en combinación con cualquier base de datos específica, por ejemplo MySQL, SQL Server, MS Access, entre otras.

2.2.23. SQLite (Fileim, 2017)

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente.

2.2.23.1. Historia

SQLite apareció en mayo del año 2000 de la mano de su creador D. Richard Hip, quién ha liberado las diferentes versiones de SQLite en base a la licencia GPL por lo que su código es de dominio público y puede ser modificado por cualquier persona. Gracias a esto, SQLite ha sido mejorada a lo largo de 7 años por un gran número de colaboradores y también ha sido migrada a diversas plataformas.

2.2.23.2. Características

Estas son algunas de las características principales de SQLite:

- La base de datos completa se encuentra en un solo archivo.
- Puede funcionar enteramente en memoria, lo que la hace muy rápida.
- Tiene un footprint menor a 230KB.
- Es totalmente autocontenida (sin dependencias externas).
- Cuenta con librerías de acceso para muchos lenguajes de programación.
- Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits.
- Soporta funciones SQL definidas por el usuario (UDF).
- El código fuente es de dominio público y se encuentra muy bien documentado.

2.2.23.3. Plataformas de SQLite (Fileim, 2017)

SQLite está construida en C, lo cual facilita la migración a diversas plataformas de sistemas operativos y de dispositivos. Dado que una base de datos de SQLite se almacena por completo en un solo archivo, está puede ser exportada a cualquier otra plataforma y tener interoperabilidad al 100% sin ningún requerimiento de programación adicional o cambios de configuración.

Las plataformas principales dónde SQLite se encuentra funcionando son:

- Windows 95, 98, ME, 2000, XP y Vista

- Windows CE & Pocket PC
- Mac OSX
- Linux
- OpenEmbedded
- PalmOS
- Symbian

2.2.23.4. Lenguajes de Programación de SQLite (Fileim, 2017)

Gracias a que SQLite es software libre, es posible encontrar una gran cantidad de componentes, librerías y drivers para interactuar con SQLite desde una gran diversidad de lenguajes y plataformas de programación. Ya sea que estemos utilizando lenguajes modernos como Java, Perl, Python, PHP, Ruby, C#, lenguajes más antiguos como Pascal, SmallTalk, Clipper, o lenguajes poco conocidos como Suneido, REXX, S-Lang, para todos podemos encontrar librerías y ejemplos de código para SQLite.

<http://www.sqlite.org/cvstrac/wiki?p=SqliteWrappers> ofrece más información sobre “wrappers” para SQLite sobre diferentes plataformas y lenguajes.

2.2.23.5. Aplicaciones de SQLite (Fileim, 2017)

Las características y plataformas previamente mencionadas hacen de SQLite una excelente opción en diversos casos tales como:

- Cuando se requiere una base de datos integrada dentro de una aplicación. SQLite es una excelente opción por su facilidad de configuración. El inconveniente es que no escala a bases de datos demasiado grandes (en el orden de los terabytes).
- Para realizar demostración de aplicaciones que utilizan un RDBMS (¿Para que utilizar un manejador de BD pesado que ocupa grandes recursos de sistema cuando solo se requiere hacer un demo de una aplicación?)

- Como cache local de un manejador de base de datos empresarial. Esto acelera el tiempo de respuesta y reduce la carga sobre la base de datos central.
- Para aplicaciones en dispositivos móviles que manejan una BD local que se sincroniza por batch con una base de datos remota.
- Almacenamiento persistente de objetos, configuraciones y preferencias de usuario. Permite fácilmente crear una estructura para almacenar configuraciones de la aplicación.

3. MATERIAL Y MÉTODOS

3.1. Material

3.1.1. Población

Para la investigación realizada la población está constituida por todos los clientes de la empresa Vecars & Trucks de la ciudad de Trujillo.

3.1.2. Muestra

Un conjunto de clientes de la empresa Vecars & Trucks, seleccionados aleatoriamente que cumplan las siguientes especificaciones:

- Tener entre 24 y 50 años.
- Poseer licencia de conducir.

3.2. Método

3.2.1. Tipo de investigación

Investigación Aplicada

3.2.2. Diseño de investigación

Se aplicó el diseño lineal con observación antes y después.

3.2.3. Variables de estudio y operacionalización

3.2.3.1. Variables

VI: Aplicación Móvil basado en IBM Watson

VD: Mejora en el diagnóstico de problemas automotrices.

3.2.3.2. Operacionalización

Variable Dependiente	Definición Conceptual	Definición Operacional	Indicadores	Instrumento	Unidad de Medida
Mejora en el diagnóstico de problemas automotrices	Capacidad de respuesta para determinar el problema y ofrecer soluciones utilizando conocimientos previamente adquiridos con el propósito de responder a las necesidades operativas de los usuarios.	Basándose en una evaluación previa se determinó los siguientes factores a tomar en cuenta en el diagnóstico. a) Facilidad de entendimiento b) Nivel de Coherencia al responder. c) Uso de palabras no técnicas para diagnósticos.	Grado de Satisfacción en el diagnóstico.	Encuesta	Porcentaje (%)

Tabla 1. Operacionalización de Variable Dependiente

Variable Independiente	Definición Conceptual	Definición Operacional	Indicadores	Instrumento	Unidad de Medida
Aplicación Móvil basado en IBM Watson	Es el conjunto de tecnologías cognitivas y herramientas de desarrollo móvil que forman parte de una aplicación que, a través de bases de conocimiento y aprendizaje cognitivo, brinde diagnósticos a problemas automotrices por medio del lenguaje natural.	Basándose en una evaluación previa se determinó los siguientes factores a tomar en cuenta en el diagnóstico. a) Capacidad de respuesta en cortos tiempos b) Fácil uso de la aplicación	Nivel de Facilidad de Uso.	Juicio de Expertos	Escala de Liker

Tabla 2. Operacionalización de Variable Independiente

3.2.4. Instrumentos de recolección de datos

Técnica	Instrumento
Encuesta	Cuestionario
Juicio de Expertos	Cuestionario

Tabla 3. Instrumentos de Recolección de Datos

3.2.5. Procedimientos y análisis de datos

a) Procesamiento de datos:

El procesamiento de los datos recolectados se realizará a través de tablas con resultados de las encuestas en Excel.

b) Análisis de datos:

c) El análisis de datos de la información contenida en las variables obtenidas de la muestra se basará en la estadística descriptiva.

4. RESULTADOS

4.1. Proyectos similares a nivel nacional

(Mendoza Riofrío, 2017) es un proyecto que Innova School.

Ha decidido utilizar a través de Watson para realizar el proceso de selección de profesores y a través de este proyecto dejan que Watson, vía una aplicación web, se encargue de filtrar entre los 10 mil candidatos a quienes cumplan con el perfil que buscan para cubrir las mil plazas que se necesitan para los nueve colegios que abrirán el próximo año, se concluyó que han logrado mejorar la calidad de este proceso mediante el uso de esta solución reduciendo el tiempo de selección de los maestros, además de permitirles dar un paso más en el análisis de sus personalidades y conocer más a fondo el perfil de sus trabajadores.

Ventajas	Desventajas
<ul style="list-style-type: none">• Mejora del Proceso de Selección de Personal.• Mejor Análisis del Personal.	<ul style="list-style-type: none">• Limitación a solo una parte del personal.• Limitación para generar posibles reportes por no tener esa posibilidad el servicio utilizado.

Tabla 4. Ventajas y Desventajas - Proyecto Innova School

(Moncada & Reyes, 2015), es el proyecto del BCP llamado “BCP ARTURITO”. Hace siete meses el BCP lanzó Arturito BCP en Facebook Messenger, como parte de los esfuerzos digitales con los que intenta sorprender al mercado. Arturito — llamado así por el droide R2-D2 de Star Wars— es un chat bot disponible 24/7 que contesta con un lenguaje natural preguntas como el saldo de una cuenta, el código interbancario o el estado de cuenta de un mes específico, luego de validar los datos del cliente con un código de verificación que envía por mensaje de texto o e-mail. Es simple: el cliente coloca su DNI, recibe el código y lo inserta en la conversación, y Arturito se encarga de cruzar los datos. El chatbot ya superó los 50,000 usuarios, de los cuales 12,000 son usuarios recurrentes. La cifra es pequeña frente a los 6,000,000 de clientes del banco, pero Arturito está en fase piloto y su principal beneficio no es el ahorro de costos, sino la fidelización que un servicio como éste

genera en los usuarios, destaca Arturo Johnson, gerente del área de canales alternativos del BCP.

La posibilidad de entrenar a Watson es lo que lo hace versátil. Recibe información, se le hace preguntas y corrige de forma sucesiva —con información adicional— hasta que adquiere su propia inteligencia.

Cada uso de Watson es un nuevo Watson, que el cliente customiza con los programadores de IBM o por sí mismo a través de un software.

Ventajas	Desventajas
<ul style="list-style-type: none">• Fácil acceso a través de un nuevo canal de atención virtual.• Fidelización de clientes.	<ul style="list-style-type: none">• Adaptación a preguntas nuevas.• No existe feedback de satisfacción del cliente.

Tabla 5. Ventajas y Desventajas - Proyecto ARTURITO del BCP

(Saya Comunicaciones S.A.C, 2017), es un proyecto de CENTRUM CATOLICA. Centrum Católica se convierte en la primera escuela de negocios del Perú en usar inteligencia artificial de IBM Watson en sus programas de MBA. En asociación con Cognitiva, socio estratégico de IBM Watson en Latinoamérica, el proyecto utiliza el API de Watson Personality Insights -un servicio basado en la plataforma en nube Bluemix, de IBM-, y permite, a través de lo que una persona ha escrito o comenta en redes sociales, analizar la personalidad de la gente. En base a esa descripción Centrum Católica asesora a su alumnos de maestría para su mejor desarrollo profesional.

Antes de la implementación de esta herramienta, el proceso se realizaba a través de la toma de cuestionarios que luego se enviaban a una consultora que devolvía los resultados después de varios días, sino meses. Con la nueva herramienta el proceso es prácticamente inmediato. "Lo importante de este modelo educativo es que, a diferencia de lo que se pudiera pensar, las empresas están demandando -además de los conocimientos- que los profesionales desarrollen capacidades que les permitan conseguir logros de negocio. Estas capacidades son desarrolladas a partir de lo que Personality Insights refiere como áreas a trabajar"

Ventaja	Desventaja
<ul style="list-style-type: none"> • Menor tiempo de análisis de perfiles de estudiantes. • Análisis superior e integral. 	<ul style="list-style-type: none"> • Costo variable. • Proceso puede llegar a dilatarse demasiado entre el envío de datos.

Tabla 6. Ventajas y Desventajas - CENTRUM Católica

4.2. Proyectos similares a nivel internacional

(Rodriguez, 2017) Es un proyecto en colaboración de Barrow Neurological Institute e IBM con la finalidad de poder dar un tratamiento a la esclerosis lateral amiotrófica mejor conocidas por sus siglas ELA, a través de Watson se pudo relacionar genes además de proteínas que guardan relación directa con dicha enfermedad para posteriormente por medio del aprendizajes poder clasificar los genes que podían estar asociados con esta enfermedad en conclusión a través de Watson analytics se pudo proporcionar información que puede utilizarse por los investigadores con el objetivo de poder desarrollar fármacos específicos.

Ventajas	Desventajas
<ul style="list-style-type: none"> • Encontrar Soluciones específicas a través de las bases de información que genera Watson. • Mayor espectro de detección de patrones relacionados con la enfermedad. 	<ul style="list-style-type: none"> • Información Limitada respecto a la enfermedad y sus síntomas.

Tabla 7. Ventajas y Desventajas - Proyecto Barrow Neurological Institute e IBM

(Diario TI, 2017) Es un caso lanzado por la empresa Ripley como parte de una estrategia de e-commerce, a través de la inteligencia artificial y el lenguaje natural de IBM, EVA asesora a sus clientes en la COMPRA online, generando recomendaciones o dando consejos sobre los productos que ofrece la empresa, enfocando esta solución a una estrategia de vanguardia y que sirva de soporte para la captación de nuevos clientes.

En conclusión, la solución además de ofrecer un canal de atención de asesoramiento ofrece un proceso ágil a sus clientes en una forma de atención personalizada. Además de llegar a captar nuevos clientes y fidelizando a los ya establecidos, y dejar una valla amplia para su competencia.

Ventajas	Desventajas
<ul style="list-style-type: none"> • Atención personalizada. • Captación de nuevos clientes. 	<ul style="list-style-type: none"> • Información incompleta debido a que las actualizaciones de los productos son frecuentes.

Tabla 8. Ventajas y Desventajas - Caso Ripley

4.3. Desarrollo del Proyecto

4.3.1. Organigrama

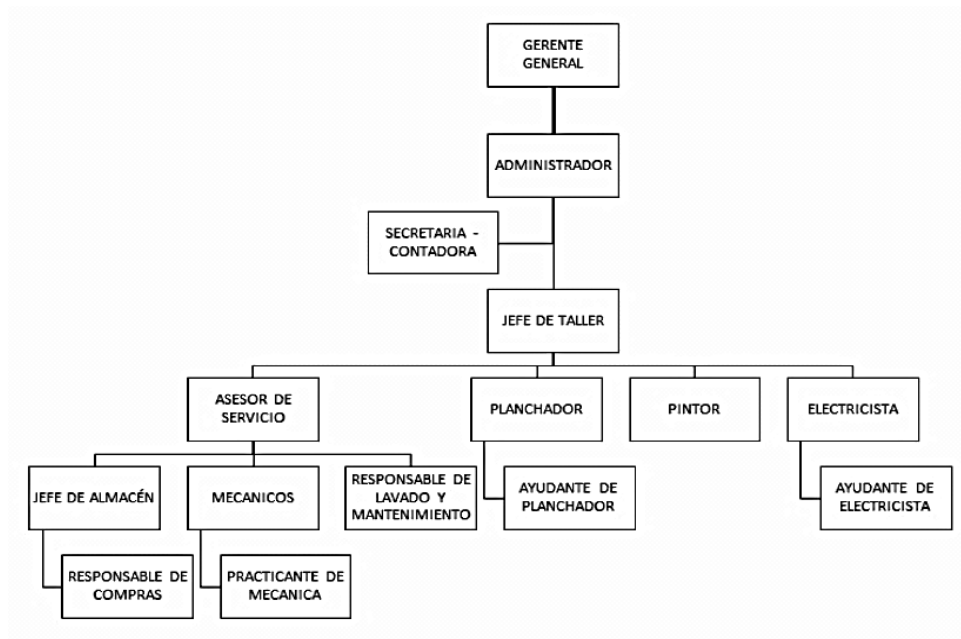


Figura 7. Organigrama de la Empresa

4.3.2. Análisis y Requerimientos

4.3.2.1. Requerimientos Funcionales

- La aplicación debe mostrar mensajes de autenticación fallida cada que las credenciales sean invalidas.

- La aplicación debe permitir registrar a nuevos usuarios para poder acceder a realizar consultas.
- La aplicación debe contar con la opción de modificar la contraseña de ingreso en caso no se recuerde la clave de acceso.
- La aplicación debe validar si el usuario se ha registrado con anterioridad o no al momento de registrarse y/o para realizar la actualización de la contraseña de ingreso.
- La aplicación móvil debe emitir un mensaje de invalidez cada vez que el usuario ingrese mal sus credenciales o sus datos.
- La aplicación debe responder a las consultas realizada por el usuario.
- La aplicación mostrará el diagnóstico emitido a los problemas que tengan los clientes.

4.3.2.2. Requerimientos No Funcionales

- La aplicación debe contar con una interfaz sencilla y fácil de entender.
- La aplicación debe ser diseñado según la arquitectura cliente-servidor.
- La aplicación debe tener una forma de respuestas sencillas y fácil de entender.
- La interface debe ofrecer mensajes e instrucciones que faciliten al usuario comprender con opciones puede realizar y que no.
- La interface de tener una estructura de navegación sencilla que facilite al usuario saber dónde se encuentra en cada momento.

4.3.3. Análisis y Requisitos

4.3.3.1. Modelo de Dominio

En esta la primera fase se construye el Modelo de Dominio, de un diagrama de clases simple que contiene las entidades/objeto que interactuarían dentro de basándose en el negocio se proyecta trabajar con este modelo a futuro.

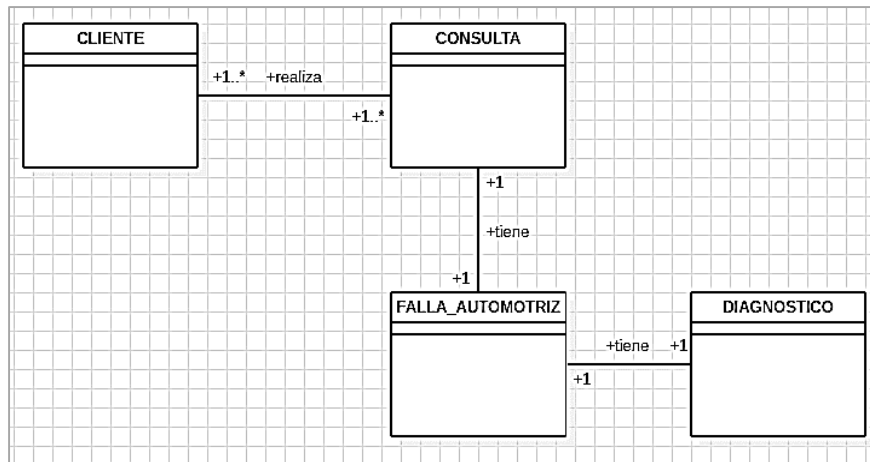


Figura 8. Modelo de Dominio

4.3.4. Análisis y Diseño Preliminar

En la fase de Análisis y diseño se construye la ficha de casos de uso, estás describe los flujos que puede tener un caso de uso.

Primero a través de un modelado se explica las acciones que podrán realizarse y luego una descripción más detallada que se ingresa en las fichas de flujo de casos de uso.

4.3.4.1. Modelado General de Casos de Uso

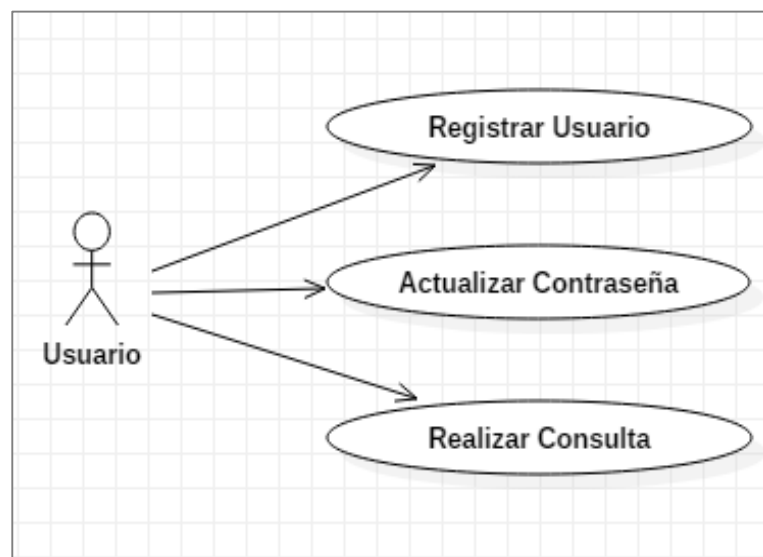


Figura 9. Diagrama de Casos de Uso

4.3.4.2. Elaboración de Fichas de Caso de Uso

Nombre:	<i>Registrar Usuario</i>
Autor:	<i>Jorge Aznarán Abarca</i>
Fecha:	<i>22/10/2018</i>
Actores:	<i>Usuarios, Clientes de la Empresa.</i>
Descripción:	<i>El usuario se registra en la aplicación para poder luego acceder y usar sus recursos.</i>
Precondiciones:	<i>No estar registrado.</i>
Flujo Normal:	<ol style="list-style-type: none"> <i>1. El Usuario llena los datos según solicite la aplicación.</i> <i>2. El Usuario Selecciona el botón registrar.</i> <i>3. La aplicación comprueba la validez de los datos y muestra un mensaje que el usuario fue registrado.</i>
Flujo Alternativo:	<ol style="list-style-type: none"> <i>1. La aplicación comprueba que los datos no son válidos y emite un mensaje de error para que pueda corregir la información.</i>
Postcondiciones:	<i>El usuario sea registrado</i>

Tabla 9. Ficha de Caso de Uso: Registrar Usuario

Nombre:	<i>Actualizar Contraseña</i>
Autor:	<i>Gustavo Aliaga Gálvez</i>
Fecha:	<i>22/10/2018</i>
Actores:	<i>Usuarios, Clientes de la Empresa.</i>
Descripción:	<i>El usuario realiza la actualización de sus credenciales obteniendo una nueva contraseña,</i>
Precondiciones:	<i>Proceso Registrar Usuario</i>
Flujo Normal:	<ol style="list-style-type: none"> <i>1. El Cliente ingresa su usuario e ingresa la nueva contraseña y el email.</i> <i>2. El Usuario ingresa nuevamente la contraseña como confirmación de la mencionada.</i> <i>3. El Usuario Selecciona el botón Actualizar.</i> <i>4. La aplicación valida que usuario se encuentre registrado y muestra un mensaje de actualización de contraseña.</i>
Flujo Alternativo:	<ol style="list-style-type: none"> <i>1. La aplicación comprueba que el usuario es incorrecto(a) y emite un mensaje de error especificando la inexistencia del usuario dentro de la base de datos.</i>

Tabla 10. Ficha de Caso de Uso: Obtener Nueva Contraseña

Nombre:	<i>Realiza Consulta</i>
Autor:	<i>Gustavo Aliaga Gálvez y Jorge Aznarán Abarca</i>
Fecha:	<i>24/10/2018</i>
Actores:	<i>Usuarios, Clientes de la Empresa.</i>
Descripción:	<i>El usuario realiza preguntas en la aplicación a través a través de un chat de consulta.</i>
Precondiciones:	<i>Procesos Registrar usuario y Loguear Usuario</i>
Flujo Normal:	<ol style="list-style-type: none"> 1. <i>El usuario ingresa por medio de una caja de texto su consulta.</i> 2. <i>La aplicación realiza la conexión con el servicio.</i> 3. <i>La aplicación responde con el diagnóstico del fallo al carro.</i> 4. <i>La aplicación pregunta si requiere algunas otro diagnostico</i> 5. <i>El usuario de no requerir algo más termina la consulta.</i>
Flujo Alternativo:	<ol style="list-style-type: none"> 1. <i>La aplicación no llega a comprender la consulta del cliente y le hace saber ello.</i> 2. <i>La aplicación le envía adjunto un link si en caso el cliente quiera registrar el problema sin solucionar.</i>

Tabla 11. Ficha de Caso de Uso: Realizar Consulta

4.3.4.3. Elaboración de Diagramas de Robustez

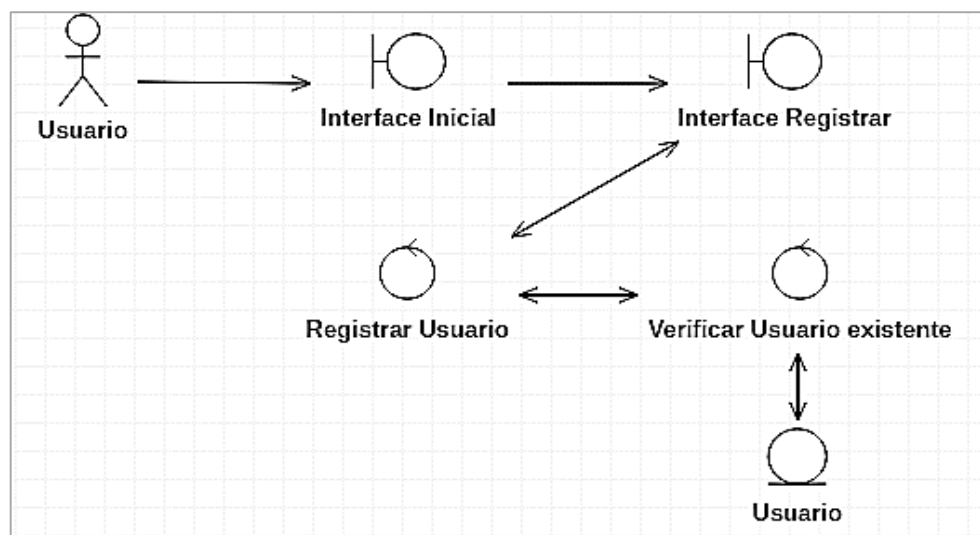


Figura 10. Diagrama de Robustez: Registrar Usuario

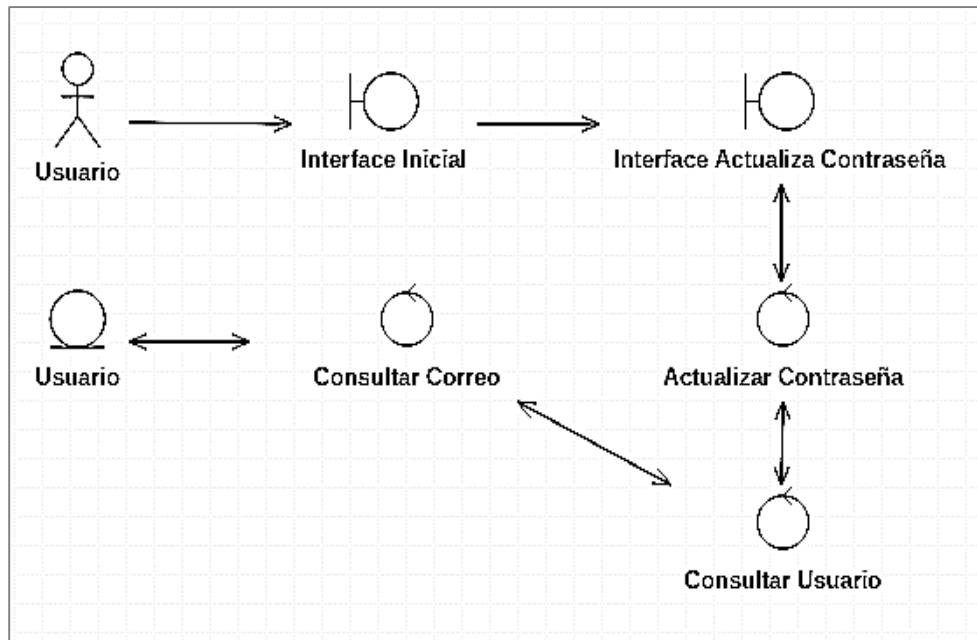


Figura 11. Diagrama de Robustez: Actualizar Contraseña

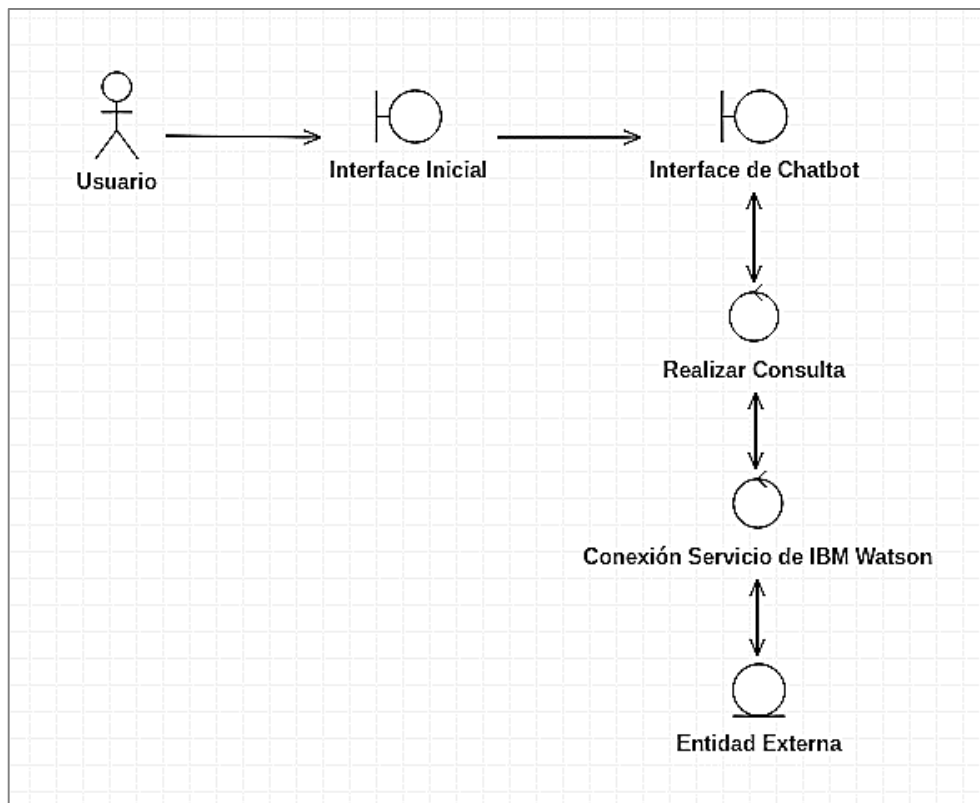


Figura 12. Diagrama de Robustez: Realizar Consulta

4.3.5. Diseño

4.3.5.1. Elaboración de Diagramas de Secuencia:

La elaboración de este elemento se hace directamente en base a las fichas de casos de uso, con la finalidad de evaluar el flujo interno que tienen cada requerimiento.

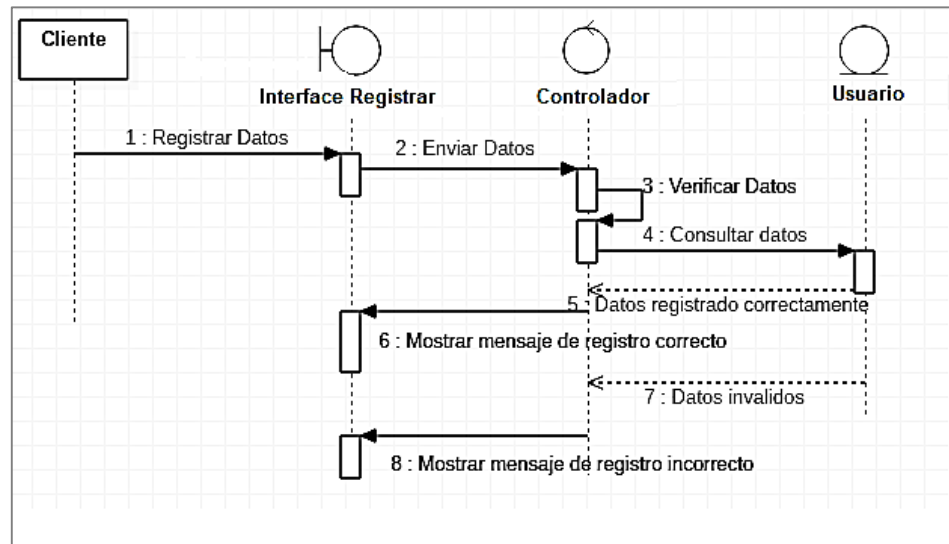


Figura 13. Diagrama de Secuencia: Registrar Usuario

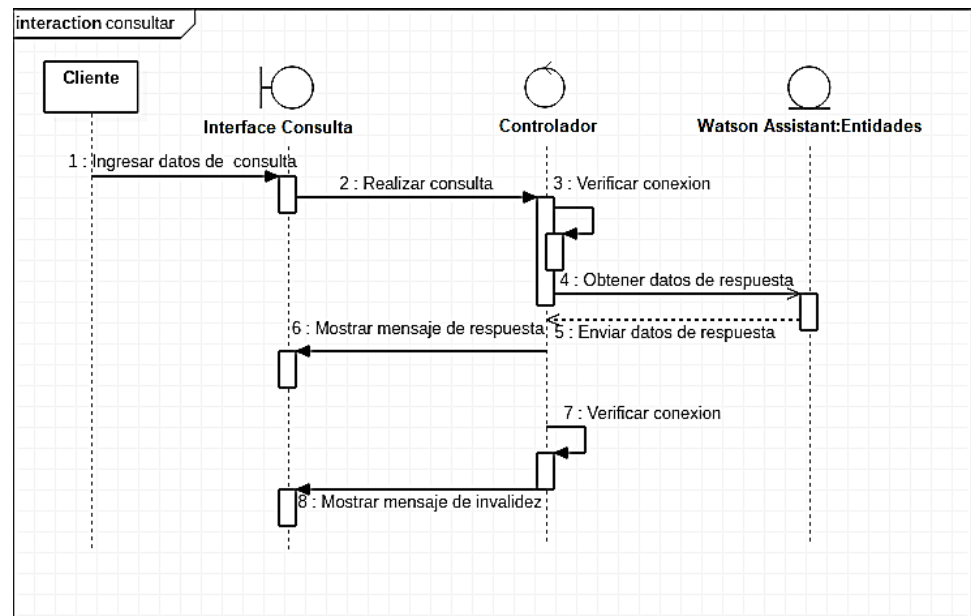


Figura 14. Diagrama de Secuencia: Realizar Consulta

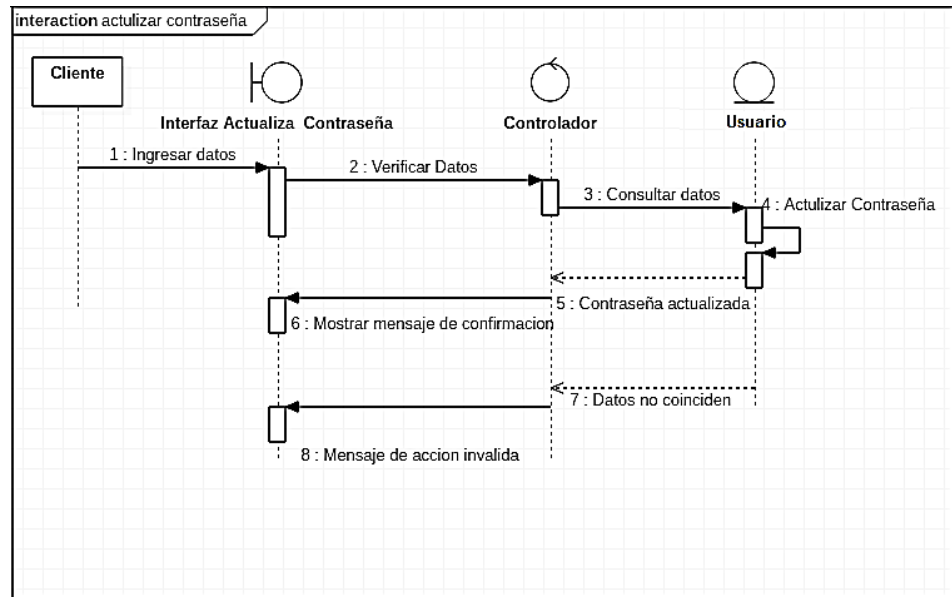


Figura 15. Diagrama de Secuencia: Actualizar Contraseña

4.3.5.2. Elaboración de Prototipos

4.3.5.2.1. Interface de Login:



Figura 16. Prototipo: Interface Login

4.3.5.2.2. Interface de Actualizar Contraseña:



Figura 17. Prototipo: Interface Actualizar Contraseña

4.3.5.2.3. Interface de Registro de Usuario:

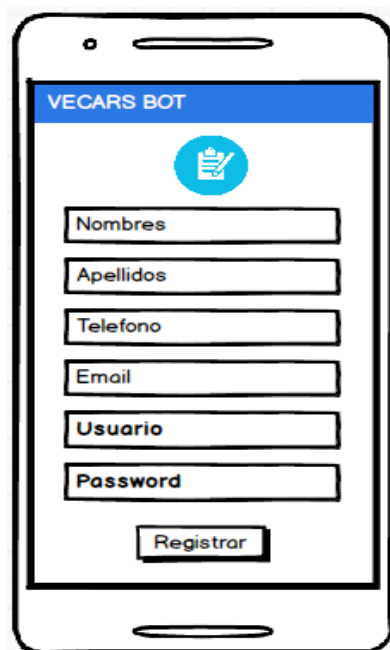


Figura 18. Prototipo: Interface Registrar Usuario

4.3.5.2.4. Interface de Chatbot:

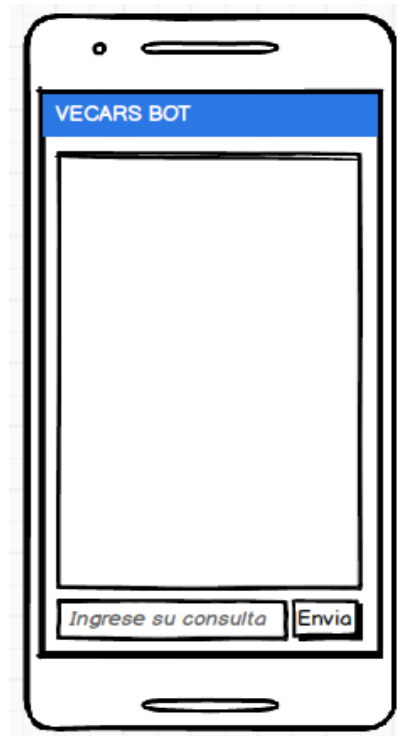


Figura 19. Prototipo: Interface ChatBot

4.3.5.3. Diagrama de Clases:

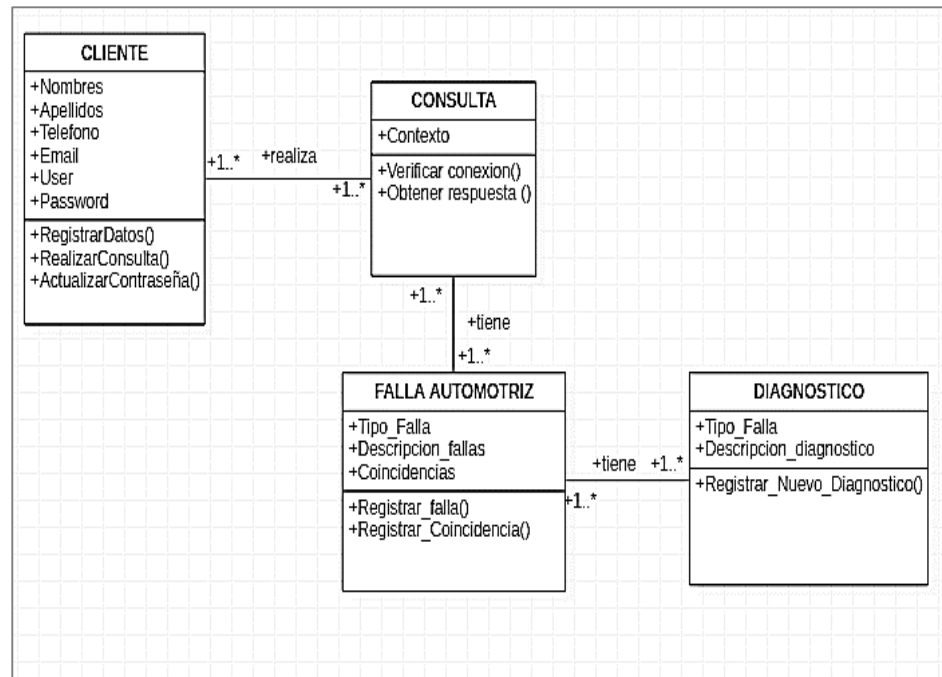
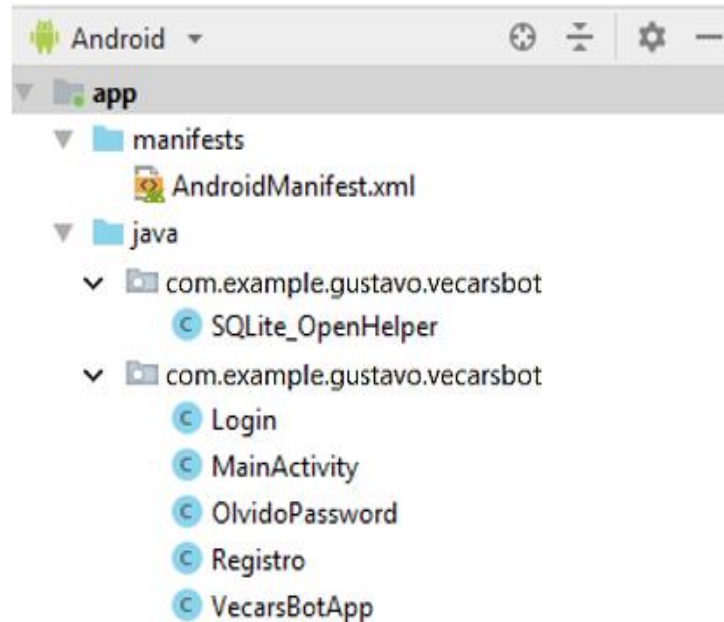


Figura 20. Diagrama de Clases

4.3.6. Implementación

4.3.6.1. Estructura y Composición del Proyecto en Android:

El proyecto creado se compone de la siguiente estructura.



Se crearon tres actividades, las cuales son las interfaces con las que el usuario tendría la posibilidad de interactuar:

- **Main.** Interface que muestra el chat donde podrá realizar las consultas.
- **Login.** Interface que muestra la pantalla de ingresos para ingresar con el usuario y contraseña.
- **Registro.** Interface que tiene como fin registrar un nuevo usuario para poder acceder a la aplicación.
- **Olvido Password.** Interface que tiene como fin actualizar contraseña en caso el usuario no pueda acceder por no recordar su contraseña de ingreso.
- **VecarsBotApp.** Es una clase creada solo con el fin de verificar la conexión de red.

A su vez, para cada una de las actividades se creó una clase, del mismo nombre, donde se realizó la elaboración del código para el

comportamiento de cada Actividad (Se hablará más a detalle de cada una en el siguiente punto).

Además de las clases mencionadas se elaboró una clase llamada SQLite_OpenHelper, donde se especifica la creación de una Base de Datos simple para almacenar a los usuarios que puedan acceder o registrarse en la aplicación.

4.3.6.2. Elaboración del Código

4.3.6.2.1. Actividades (Interfaces):

4.3.6.2.1.1. Login

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".Login">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="180dp"
        android:src="@drawable/login5"
        android:layout_margin="20dp"
        />

    <EditText
        android:id="@+id/txtUser"
        android:layout_width="match_parent"
        android:layout_height="62sp"
        android:layout_margin="2sp"
        android:drawableLeft="@drawable/ic_action_name"
        android:ems="10"
        android:hint="Usuario"
        android:drawablePadding="15dp"
        android:inputType="textPersonName"
        android:maxLength="15" />

    <android.support.design.widget.TextInputEditText
        android:id="@+id/txtPass"
        android:layout_width="match_parent"
        android:layout_height="62sp"
        android:layout_margin="2sp"
        android:drawableLeft="@drawable/ic_action_pass"
        android:drawablePadding="15dp"
        android:hint="Password"
        android:inputType="textPassword"
        android:maxLength="15" />
```

```
<Button
    android:id="@+id/btnLogin"
    android:layout_width="117dp"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:layout_margin="@dimen/default_margin"
    android:background="#FF2196F3"
    android:text="LOGIN"
    android:textColor="@android:color/background_light"
    android:textColorLink="@color/colorPrimaryDark"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="¿Olvidaste tu Contraseña?"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:textStyle="bold"
    android:text="Regístrate aquí" />

</LinearLayout>
```

4.3.6.2.1.2. Registrar

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".Registro">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="80sp"
        android:src="@drawable/login6"
        android:layout_marginBottom="10dp"
        android:layout_gravity="center" />

    <EditText
        android:id="@+id/txtNombres"
        android:layout_width="match_parent"
        android:layout_height="50sp"
        android:layout_margin="2sp"
        android:ems="10"
        android:digits="ABCDEFGHIJKLMNOPQRSTUVWXYZ "
        android:hint="Nombres"
        android:inputType="textCapCharacters"
        android:maxLength="20" />

    <EditText
        android:id="@+id/txtApellidos"
        android:layout_width="match_parent"
        android:layout_height="50sp"
        android:layout_margin="2sp"
        android:ems="10"
        android:digits="ABCDEFGHIJKLMNOPQRSTUVWXYZ "
        android:hint="Apellidos"
        android:inputType="textCapCharacters"
        android:maxLength="20" />
```

```
<EditText
    android:id="@+id/txtTelefono"
    android:layout_width="match_parent"
    android:layout_height="50sp"
    android:layout_margin="2sp"
    android:hint="Telefono"
    android:digits="0123456789- "
    android:inputType="textCapCharacters"
    android:maxLength="15" />

<EditText
    android:id="@+id/txtEmail"
    android:layout_width="match_parent"
    android:layout_height="50sp"
    android:layout_margin="2sp"
    android:hint="Email"
    android:inputType="textEmailAddress"
    android:maxLength="30" />

<EditText
    android:id="@+id/txtUsuario"
    android:layout_width="match_parent"
    android:layout_height="50sp"
    android:layout_margin="2sp"
    android:hint="Usuario"
    android:inputType="textPersonName"
    android:maxLength="15"
    android:textStyle="bold"/>

<EditText
    android:id="@+id/txtPassword"
    android:layout_width="match_parent"
    android:layout_height="50sp"
    android:layout_margin="2sp"
    android:hint="Password"
    android:inputType="textPassword"
    android:maxLength="15"
    android:textSize="18sp"
    android:textStyle="bold" />
```

```

<EditText
    android:id="@+id/txtPassword"
    android:layout_width="match_parent"
    android:layout_height="50sp"
    android:layout_margin="2sp"
    android:hint="Password"
    android:inputType="textPassword"
    android:maxLength="15"
    android:textSize="18sp"
    android:textStyle="bold" />

<Button
    android:id="@+id/btnRegistrar"
    android:layout_width="117dp"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:layout_margin="5dp"
    android:background="#FF2196F3"
    android:text="REGISTRAR"
    android:textColor="@android:color/background_light"
    android:textStyle="bold" />

</LinearLayout>

```

4.3.6.2.1.3. Main

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    tools:context=".MainActivity"
    android:orientation="vertical">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="636dp"
        app:layout_constraintBottom_toTopOf="@id/inputRelativeLayout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0">

        <TextView
            android:id="@+id/text_View"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:autoLink="web"
            android:textColorLink="@color/colorAccent"
            android:textSize="16sp" />

    </ScrollView>

```

```

<RelativeLayout
    android:id="@+id/inputRelativeLayout"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent">

    <EditText
        android:id="@+id/text_Edit"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"
        android:layout_gravity="start"
        android:layout_margin="5dp"
        android:padding="15dp"
        android:background="@drawable/caja_redonda"
        android:layout_toStartOf="@+id/btnEnviar"
        android:hint="Ingresa Consulta"
        android:imeOptions="actionDone"
        android:inputType="textShortMessage"
        android:textColorLink="@color/colorAccent" />

    <Button
        android:id="@+id/btnEnviar"
        android:layout_width="70dp"
        android:layout_height="match_parent"
        android:layout_alignParentEnd="true"
        android:hint="Enviar" />

</RelativeLayout>

```

4.3.6.2.1.4. Olvidar Contraseña

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16px"
    android:orientation="vertical"
    tools:context=".OlvidoPassword">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="180dp"
        android:src="@drawable/contrasena"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="8dp"
    />

    <EditText
        android:id="@+id/txtUser1"
        android:layout_width="match_parent"
        android:layout_height="50sp"
        android:layout_margin="2sp"
        android:drawableLeft="@drawable/ic_action_name"
        android:drawablePadding="15dp"
        android:ems="10"
        android:hint="Usuario"
        android:inputType="textPersonName"
        android:maxLength="15" />

    <EditText
        android:id="@+id/txtEmail1"
        android:layout_width="match_parent"
        android:layout_height="50sp"
        android:layout_margin="2sp"
        android:drawableLeft="@drawable/ic_action_name"
        android:drawablePadding="15dp"
        android:ems="10"
        android:hint="Email"
        android:inputType="textEmailAddress"
    />

```

```

        android:inputType="textEmailAddress"
        android:maxLength="30" />

<android.support.design.widget.TextInputEditText
    android:id="@+id/txtPass1"
    android:layout_width="match_parent"
    android:layout_height="50sp"
    android:layout_margin="2sp"
    android:drawableLeft="@drawable/ic_action_pass"
    android:drawablePadding="15dp"
    android:hint="Nueva Contraseña"
    android:inputType="textPassword"
    android:maxLength="15" />

<android.support.design.widget.TextInputEditText
    android:id="@+id/txtConfirmPass1"
    android:layout_width="match_parent"
    android:layout_height="50sp"
    android:layout_margin="2sp"
    android:drawableLeft="@drawable/ic_action_confirm"
    android:drawablePadding="15dp"
    android:hint="Confirmar Contraseña"
    android:inputType="textPassword"
    android:maxLength="15" />

<Button
    android:id="@+id/btnActualizar"
    android:layout_width="117dp"
    android:layout_height="50dp"
    android:layout_gravity="center"
    android:layout_margin="@dimen/default_margin"
    android:background="#FF2196F3"
    android:text="GUARDAR"
    android:textColor="@android:color/background_light"
    android:textColorLink="@color/colorPrimaryDark"
    android:textStyle="bold" />

```

4.3.6.2.2. Clases

4.3.6.2.2.1. Clase Login

La Clase se compone de los paquetes que se importan y que se utilizarán de apoyo en la construcción de la clase, a su vez esta se encuentra conectada a la Clase SQLite_OpenHelper, para realizar la validación del usuario al ingresarse el usuario.

Por otro lado, la clase tiene un vínculo con la Actividad de Registro, ya que en caso el usuario no se encuentre Registrado en la Base de Datos pueda ir a esta Interface y registrarse.

```

package pe.edu.upc.vecarsbot;

import android.content.Intent;
import android.database.Cursor;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import OpenHelper.SQLite_OpenHelper;

```

```

import OpenHelper.SQLite_OpenHelper;

public class Login extends AppCompatActivity {
    private EditText User;
    private EditText Password;
    private TextView Registrar, Recuperar;
    private Button Login;

    SQLite_OpenHelper helper = new SQLite_OpenHelper(this, "BD_USER", null,1);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        Login = findViewById(R.id.btnLogin);
        Login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent evento = new Intent(Login.this,MainActivity.class);
                startActivity(evento);
            }
        });

        Registrar = findViewById(R.id.textView);
        Registrar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent evento_1 = new Intent(Login.this,Registro.class);
                startActivity(evento_1);
            }
        });

        Recuperar = findViewById(R.id.textView1);
        Recuperar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent evento_2 = new Intent(Login.this,OlvidoPassword.class);
                startActivity(evento_2);
            }
        });
    }
}

```

```

Login=(Button)findViewById(R.id.btnLogin);
Login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //Enlazar Variables con Id los EditText
        EditText User = (EditText)findViewById(R.id.txtUser);
        EditText Password = (EditText)findViewById(R.id.txtPass);

        Cursor cursor = helper.ConsultarUP
            (User.getText().toString(), Password.getText().toString());

        if (cursor.getCount() > 0) {
            Toast.makeText(getApplicationContext(), ";BIENVENIDO!", Toast.LENGTH_LONG).
                show();
            Intent i = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(i);
            finish();
        } else {
            Toast.makeText(getApplicationContext(), "Usuario y/o Password son
                incorrectos", Toast.LENGTH_LONG).show();
        } //fin del else
        User.setText("");
        Password.setText("");
        User.requestFocus();
    }
});
}
}

```

4.3.6.2.2.2. Clase Main Activity

La Clase al igual que la anterior se compone de los paquetes que se importan para utilizar sus recursos.

```

package pe.edu.upc.vecarsbot;

import android.content.Context;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Html;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.androidnetworking.AndroidNetworking;
import com.androidnetworking.common.Priority;
import com.androidnetworking.error.ANError;
import com.androidnetworking.interfaces.JSONObjectRequestListener;

import org.json.JSONException;
import org.json.JSONObject;

```

En esta clase se elabora la construcción del Chat así como también la forma en que se activa con la acción del usuario el uso de los recursos con los Servicios de Watson Assistant a través de una la cadena de conexión indicando donde se encuentra dicho recurso.

```

public class MainActivity extends AppCompatActivity {
    private TextView text_View;
    private EditText text_Edit;
    private Button btnEnviar;
    private static Long contador;
    private Context context;
    private JSONObject contextJsonObject = null;

    @Override
    public void onBackPressed() {
        if(contador + 2000 > System.currentTimeMillis()) {
            super.onBackPressed();
            Intent miIntent = new Intent(MainActivity.this, Login.class);
            startActivity(miIntent);
        } else {
            Toast.makeText(getApplicationContext(),
                "Presione nuevamente para cerrar sesión",
                Toast.LENGTH_SHORT).show();
            contador = System.currentTimeMillis();
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        text_View = findViewById(R.id.text_View);
        text_Edit = findViewById(R.id.text_Edit);
        btnEnviar = findViewById(R.id.btnEnviar);
        context = this;
    }
}

```

Llamada al servicio de Watson Assistant:

```

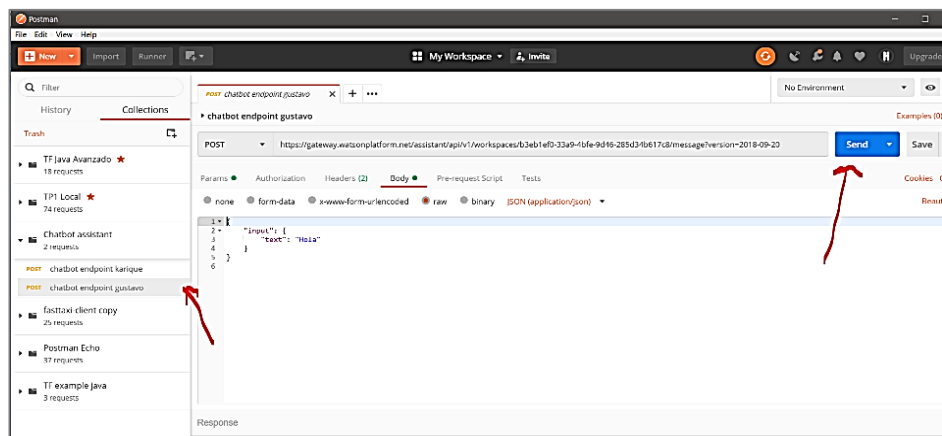
private void getResponseToUserInput(String userInputText){
    String urlAssistant =
        "https://gateway.watsonplatform.net/assistant/api/v1/workspaces/
        b3eb1ef0-33a9-4bfe-9d46-285d34b617c8/message?version=2018-09-20";
    String userPasswordBase64 = "
        NTBmY2M2ZWItY2E5M100MjktLWJjMmQyY2YyZTMzYzRjMzc40mpT1VzYVdIUxhIcg==";
}

```

Se realizó un proceso externo para tener el url que nos ayudaría al obtener respuestas del Assistant así como el usuario y la password para acceder a sus recursos:

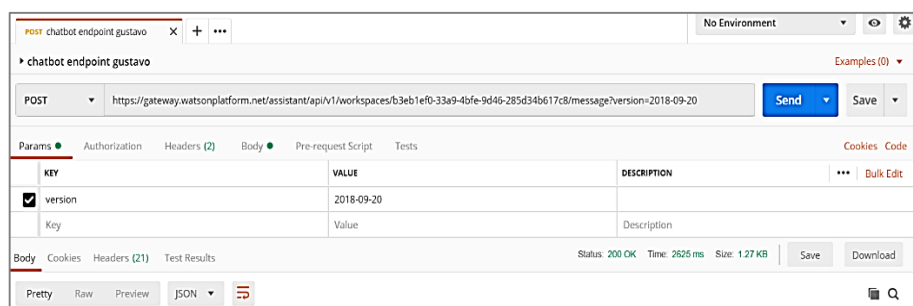
Se utilizó la herramienta **Postman** con dos fines: El primero era conocer y evaluar el proceso interno de la conversación y como es que se manejan los mensajes que recibe y envía Watson y es que este utiliza el lenguaje JSON como estructura para enviar los mensajes.

A continuación, se presenta el entorno básico de Postman, configurado para gestionar los mensajes internos.



El segundo fin es facilitarnos saber que ocurre tras realizar un llamado al proceso de Watson y tener un control de lo que queremos obtener.

La configuración antes mostrada requiere agregar una versión, así como también el ingreso del espacio de trabajo y las credenciales que el servicio de Watson provee:



Encode to Base64 format

Simply use the form below

username:password

i To encode binaries (like images, documents, etc.) upload your data via the [file encode form](#) below.

UTF-8 Destination charset.

LF (Unix) Newline separator.

Split lines into 76 character wide chunks (useful for MIME).

Live mode OFF Encodes in real-time when you type or paste (supports only unicode charsets).

> ENCODE < Encodes your data into the textarea below.

dXNlcm5hbWU6cGFzc3dvcmQ=

Luego de haber realizado ese proceso se inicia elaborando lo que sería la estructura de obtención del mensaje:

```

try {
    inputJSONObject.put("text", userInputText);
} catch (JSONException e) {
    e.printStackTrace();
}
JSONObject inputJSONObjectBody = new JSONObject();
// { "input": { "text": "hola" } }
try {
    inputJSONObjectBody.put("input", inputJSONObject);
    //adjunto el contexto para que el assistant entienda que sigo en la misma
    conversacion
    if (contextJsonObject != null)
        inputJSONObjectBody.put("context", contextJsonObject);
} catch (JSONException e) {
    e.printStackTrace();
}

```

El proceso se define como la consulta del cliente enviado a través de un objeto en JSON, este se envía y a su vez se le responde a través de esta estructura en dicho lenguaje, la respuesta es enviada con la misma estructura y la codificación se encarga de conectar la trama del mensaje para seguir el hilo

de la conversación además de mostrar los mensajes en el Visor de Texto.

```
AndroidNetworking.post(urlAlAssistant)
    .addHeaders("Content-Type", "application/json")
    .addHeaders("Authorization", "Basic " + userPasswordBase64)
    .addJSONObjectBody(inputJSONObjectBody)
    .setPriority(Priority.HIGH)
    .setTag(getString(R.string.app_name))
    .build()
    .getAsJSONObject(new JSONObjectRequestListener() {
        @Override
        public void onResponse(JSONObject response) {
            String outputText = "";

            try {
                outputText = response.getJSONObject("output").getJSONArray("text").
                    getString(0);
                if (contextJsonObject == null)
                    contextJsonObject = response.getJSONObject("context");
            } catch (JSONException e) {
                e.printStackTrace();
            }

            text_View.append(
                Html.fromHtml("<p><b>BOT:</b> " + outputText + "</p>")
            );
        }

        @Override
        public void onError(ANError anError) {
            Toast.makeText(context, "Hubo un error con la conexión a el assistant: "
                + anError.getErrorBody(), Toast.LENGTH_LONG).show();
        }
    });
```

4.3.6.2.2.3. Clase Registro Registro

La Clase Registro contiene librerías importadas que se utilizan como recurso también, para la elaboración del Código, a su vez esta clase extiende de la clase SQLiteOpenHelper ya que requiere utilizar la base de datos para comparar si el registro de usuario ya se encuentra en la base o puede realizar esta acción sin problema.

4.3.6.2.2.4. Clase Olvido_Contraseña

Esta Clase contiene la codificación realizada para la creación de una nueva contraseña pensando en referencia a que el usuario pierda u olvide su clave de acceso.

Se requiere que la aplicación reconozca un usuario antiguo y a su vez verifique la confirmación de la nueva contraseña.

```
public class OlvidoPassword extends AppCompatActivity {  
  
    private EditText User;  
    private EditText Email;  
    private EditText Password;  
    private EditText Confirma_Password;  
    private Button Recuperar;  
  
    SQLiteOpenHelper helper = new SQLiteOpenHelper(this, "BD_USER", null,1);  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_olvido_password);  
        Recuperar=(Button)findViewById(R.id.btnActualizar);  
  
        Recuperar.setOnClickListener(new View.OnClickListener() {  
@Override  
public void onClick(View v) {  
            User=(EditText)findViewById(R.id.txtUser1);  
            Email=(EditText)findViewById(R.id.txtEmail1);  
            Password=(EditText)findViewById(R.id.txtPass1);  
            Confirma_Password=(EditText)findViewById(R.id.txtConfirmPass1);  
            Cursor cursor = helper.ConsultarUserEmail(User.getText().toString(), Email.getText().toString());  
  
            if (User.getText().toString().isEmpty()){  
                Toast.makeText(getApplicationContext(),  
                    "El campo Nombre está vacío", Toast.LENGTH_LONG).show();  
            }  
            else {  
                if (Email.getText().toString().isEmpty()){  
                    Toast.makeText(getApplicationContext(),  
                        "El campo Email está vacío", Toast.LENGTH_LONG).show();  
                }  
                else {  
                    if (Email.length() < 4) {  
                        Toast.makeText(getApplicationContext(),  
                            "El campo debe contener un '@' y al menos 10 caracteres", Toast.LENGTH_LONG).show();  
                    }  
                }  
            }  
        }  
    }  
}
```

```
            else {  
                if (Password.getText().toString().isEmpty()){  
                    Toast.makeText(getApplicationContext(),  
                        "El campo Nueva Contraseña está vacío", Toast.LENGTH_LONG).show();  
                }  
                else {  
                    if (Password.length() < 4) {  
                        Toast.makeText(getApplicationContext(),  
                            "El campo debe contener más 3 caracteres", Toast.LENGTH_LONG).show();  
                    }  
                    else {  
                        if (Confirma_Password.getText().toString().isEmpty()) {  
                            Toast.makeText(getApplicationContext(),  
                                "El campo Confirmar Contraseña está vacío", Toast.LENGTH_LONG).show();  
                        }  
                        else {  
                            if (Confirma_Password.length() < 4){  
                                Toast.makeText(getApplicationContext(),  
                                    "El campo debe contener 3 o más caracteres", Toast.LENGTH_LONG).show();  
                            }  
                            else {  
                                if (!Password.getText().toString().equals(Confirma_Password.getText().toString())){  
                                    Toast.makeText(getApplicationContext(), "Las contraseñas no coinciden", Toast.  
                                        LENGTH_LONG).show();  
                                }  
                                else {  
                                    if (cursor.getCount() > 0) {  
                                        helper.actualizarPassword(User.getText().toString(),Email.getText().toString(),  
                                            Password.getText().toString());  
                                        Toast.makeText(getApplicationContext(), "SE ACTUALIZO LA CONTRASEÑA, CON EXITO",  
                                            Toast.LENGTH_LONG).show();  
                                        Intent eventIntent = new Intent(OlvidoPassword.this,Login.class);  
                                        startActivity(eventIntent);  
                                        finish();  
                                    }  
                                    else {  
                                        Toast.makeText(getApplicationContext(), "El Usuario o Email Ingresado no  
                                            coincide con los Registrados", Toast.LENGTH_LONG).show();  
                                        //fin del else  
                                    }  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

4.3.6.2.2.5. Clase SQLite_OpenHelper

Esta Clase contiene la codificación realizada para la elaboración de una Base de Datos simple y liviana para almacenar a los usuarios Registrados, con el fin de detectar ello y dar acceso o restrinja a los nuevos ingresos.

Se espera que esta también sirva como registro de clientes que acceden a la aplicación y tener también datos básicos de estos, para un seguimiento posterior de la empresa.

```
package OpenHelper;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class SQLite_OpenHelper extends SQLiteOpenHelper {

    public SQLite_OpenHelper(Context context, String name,
        SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    //Creación de estructura de Tablas
    @Override
    public void onCreate(SQLiteDatabase db) {
        //nombre apellidos placa de auto modelo usuario password
        String query=
            "create table usuarios(_ID integer primary key autoincrement, " +
            "Nombres text, Apellidos text, Telefono text, Email text, Usuario text, Password text);";
        db.execSQL(query);
    }

    //Modificación de Estructura de Base de Datos
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }

    //Método para Abrir BD
    public void abrir(){
        this.getWritableDatabase();
    }

    //Método para Cerrar BD
    public void cerrar(){
        this.close();
    }
}
```

```
public void insertarReg(String nombres, String apellidos, String telefono,
    String email, String usuario, String password){
    ContentValues valores = new ContentValues();
    valores.put("Nombres",nombres);
    valores.put("Apellidos",apellidos);
    valores.put("Telefono",telefono);
    valores.put("Email",email);
    valores.put("Usuario",usuario);
    valores.put("Password",password);

    this.getWritableDatabase().insert("usuarios",null,valores);
}

//Método para Validar si el Usuario Existe
public Cursor ConsultarUP(String usuario, String password) throws SQLException {
    //throw = Control de errores
    Cursor mcursor = null;
    mcursor=this.getReadableDatabase().query("usuarios",new String[]
    {"_ID", "Nombres", "Apellidos", "Telefono", "Email", "Usuario", "Password"},
    "Usuario like '"+usuario+"' "+"and Password like '"+password+"'"+",null, null, null);
    return mcursor;
}

public Cursor ConsultarUser(String usuario) throws SQLException {
    //throw = Control de errores
    Cursor cursor1 = null;
    cursor1=this.getReadableDatabase().query("usuarios",new String[]{"_ID",
    "Nombres", "Apellidos", "Telefono", "Email", "Usuario", "Password"},
    "Usuario like '"+usuario+"'"+",
    null, null, null, null);
    return cursor1;
}
```

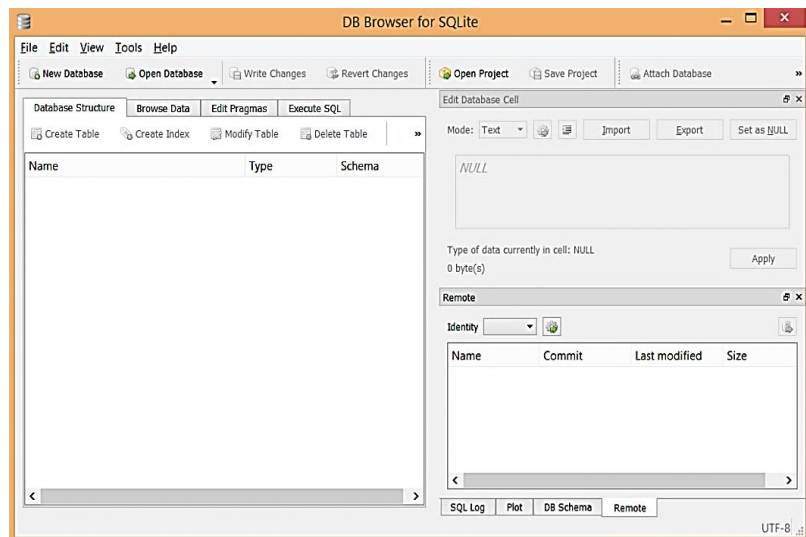
```

public Cursor ConsultarUserEmail(String usuario, String email) throws SQLException {
    //throw = Control de errores
    Cursor cursor2 = null;
    cursor2=this.getReadableDatabase().query("usuarios",new String[]{"_ID",
        "Nombres", "Apellidos", "Telefono", "Email", "Usuario", "Password"},
        "Usuario like '"+usuario+"' " +
        "and Email like '"+email+"'","", null, null, null, null);
    return cursor2;
}

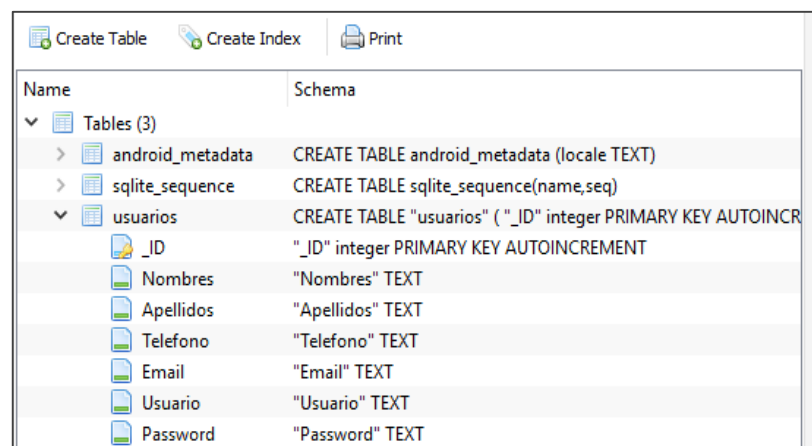
public void actualizarPassword(String usuario, String email, String password){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues Update_values = new ContentValues();
    Update_values.put("Password", password);
    db.update("usuarios", Update_values, "Usuario=? AND Email=?", new String[]{usuario,email});
    db.close();
}
}

```

Para la Visualización de la Base creada utilizamos la herramienta DB Browser for SQLite solo con el fin de visualizar la estructura de la Tabla, al generar por código la tabla y ejecutar el programa se crea la base de datos, en un tipo de archivo simple.



Se abre ese archivo generado solo con fines de revisar la estructura:



Se puede observar la Tabla creada “Usuarios”, donde además se registró para pruebas un primer Usuario que acceda para el Testeo.

Table: usuarios						
_ID	Nombres	Apellidos	Telefono	Email	Usuario	Password
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	GUSTAVO	ALIAGA GALVEZ	220036	guhek.tweet...	GHAG	GHAG

4.3.6.3. Construcción del Servicio de Watson Assistant

4.3.6.3.1. Creación de intenciones

En este bloque, básicamente tenemos la creación de la estructura.

Dialogo (los problemas, el saludo, la dirección del local, etc), dentro de cada intención se encuentra las diversas opciones por la cual el Bot puede responder.

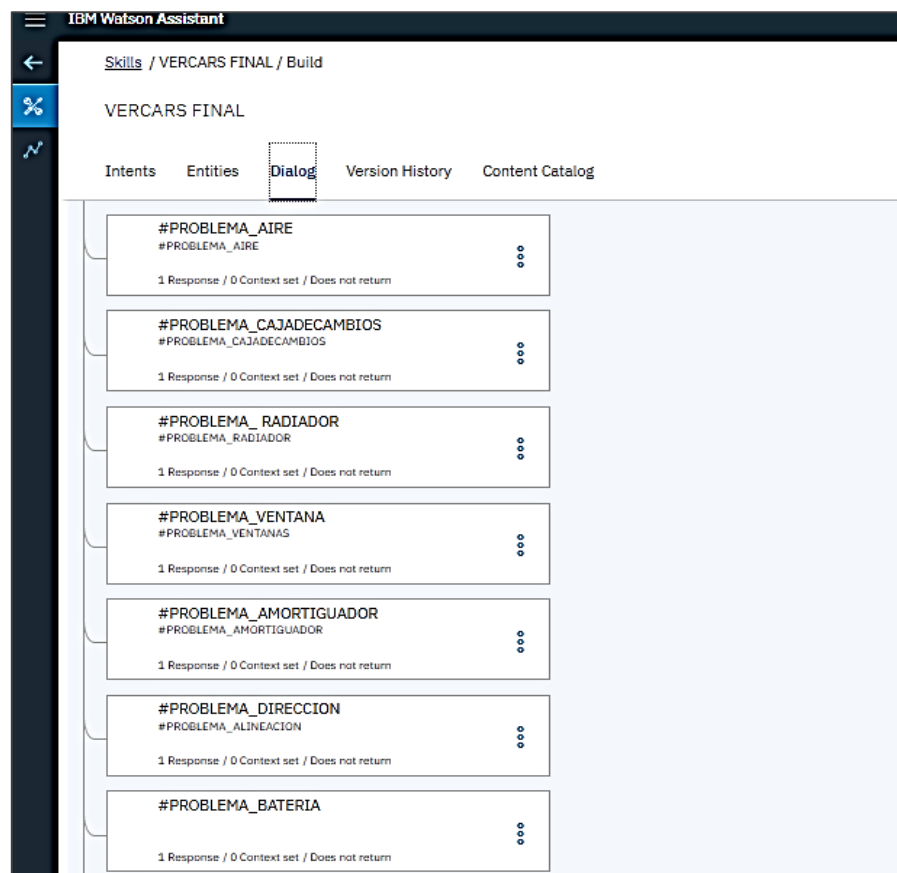
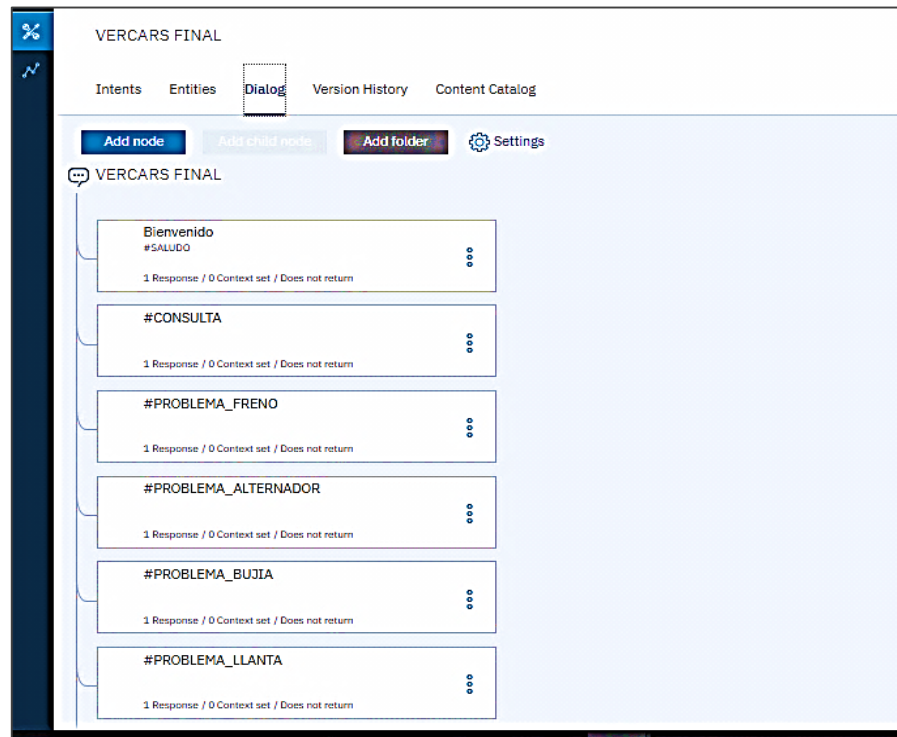
Intents	Entities	Dialog	Analytics	Versions	Content Catalog
Create intent					
<input type="checkbox"/>	Intent (17)		Description (optional)		Modified
<input type="checkbox"/>	#NEGACION		NO TIENE OTRA CONSULTA		a month ago
<input type="checkbox"/>	#PROBLEMA_ACEITE		cliente explica problema con el cambio de aceite		3 months ago
<input type="checkbox"/>	#PROBLEMA_AIRE		Problema con el aire acondicionado		2 months ago
<input type="checkbox"/>	#PROBLEMA_ALINEACION		cliente presenta problemas con la direccion del c...		2 months ago
<input type="checkbox"/>	#PROBLEMA_ALTERNADOR		Cliente explica problema con el alternador		2 months ago
<input type="checkbox"/>	#PROBLEMA_AMORTIGUADOR		cliente presenta problemas con el amortiguador.		2 months ago

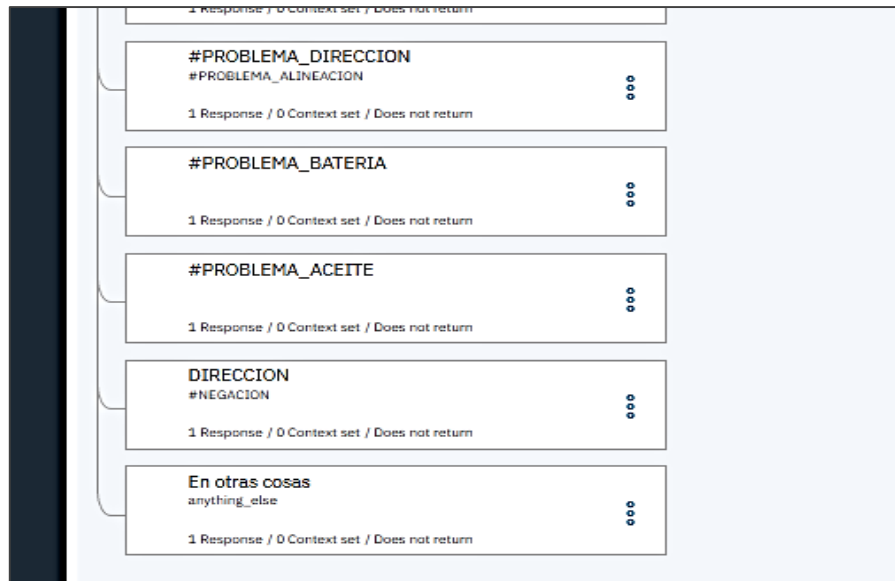
Como, por ejemplo, en la siguiente imagen podemos apreciar como el bot va a responder respecto al problema (motor), con las diversas interrogantes respecto a ello:

#PROBLEMA_MOTOR		Last modified a month ago	Download	Delete	Search
Intent name					
#PROBLEMA_MOTOR					
Name your intent to match a customer's question or goal. For example, #pay_bill or #open_account.					
Description (optional)					
cliente presenta problema con el motor, no acelera demasiado.					
Add user example					
[type a user example here]					
Add example					
<input type="checkbox"/>	User examples (7)	Added	Show only conflicts	1	
<input type="checkbox"/>	en mi auto piso a fondo el acelerador y no paso los 50 km por hora	6 months ago			
<input type="checkbox"/>	mi auto no pasa de los 50 km por hora	6 months ago			
<input type="checkbox"/>	mi motor no da	a month ago			

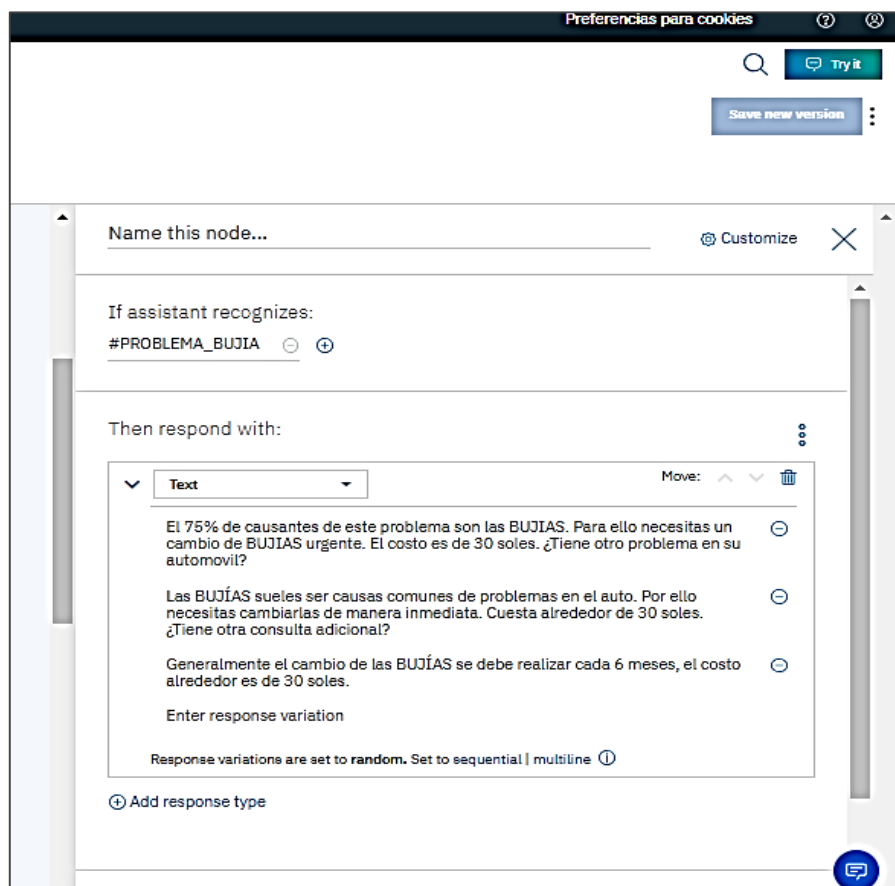
4.3.6.3.2. Creación de dialogo

En este bloque encontramos la estructura del dialogo:



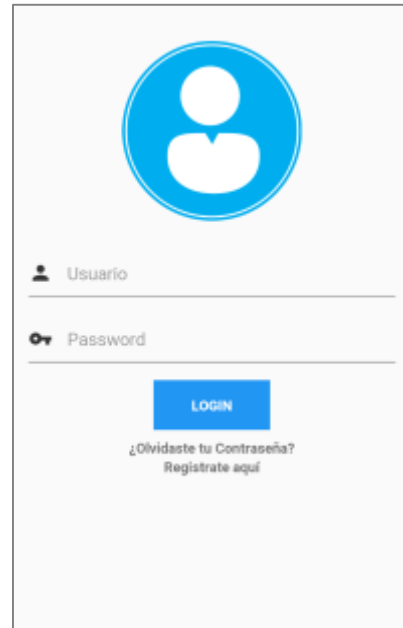
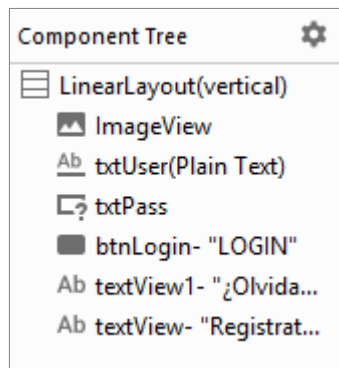


Así como también las respuestas del Bot de acuerdo a cada problema que encuentre en el dialogo, por ejemplo, las posibles respuestas al problema de bujías:

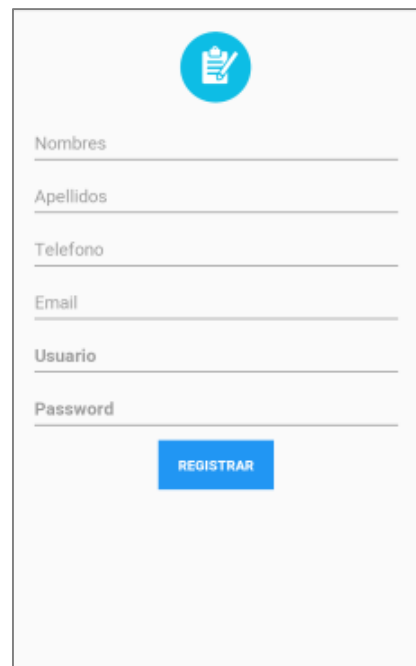
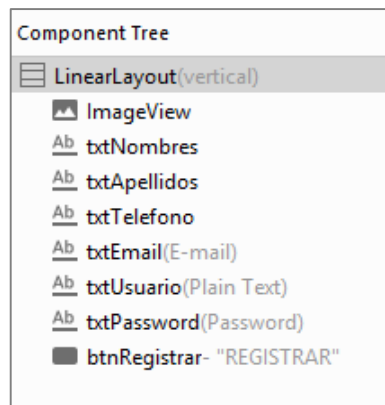


4.3.6.4. Interfaces

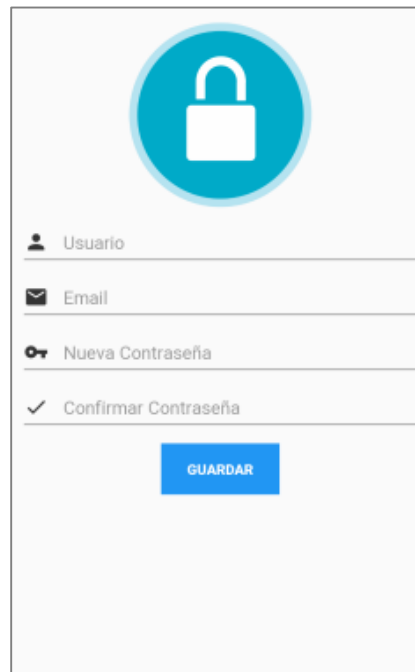
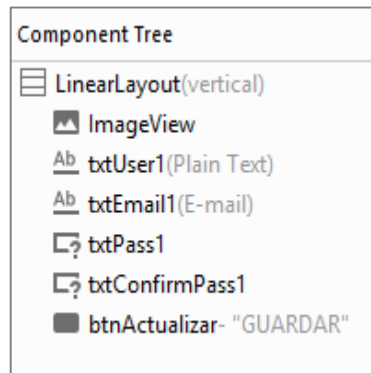
4.3.6.4.1. Interface de Login



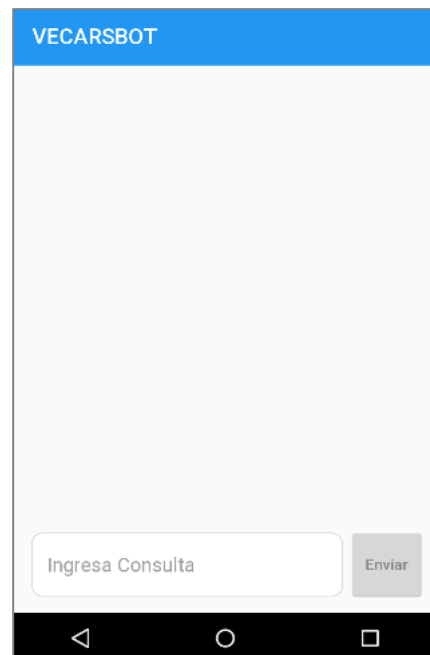
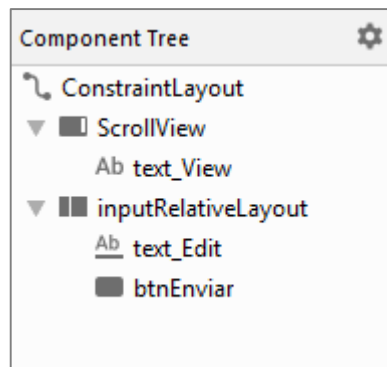
4.3.6.4.2. Interface de Registro



4.3.6.4.3. Interface de Actualizar Contraseña



4.3.6.4.4. Interface Main



4.4. Testeo y Retroalimentación:

Pruebas de Registro de Usuario	
Input	Output
“Gustavo”, “Aliaga Gálvez”, “586954”, “guhek.tweet@gmail.com”, “gustavo12”, “123”	Registro Completo
“Jorge”, “”, “5863258”, “”, “jorge23”, “aznarana12”	Campos Incompletos
“”, “”, “”, “”, “ADMIN”, “Galvez”	Campos Incompletos
“Jorge”, “Aznaran”, “5863258”, “jaznara@gmail.com”, “jaznarana”, “george12”	Registro Completo
“adb”, “”, “”, “”, “”, “”, “”	Su nombre debe comprender más de 3 caracteres.

Tabla 12. Pruebas de Registro de Usuario

Pruebas de Login de Usuario	
Input	Output
“GUSTAVO12”, “123”	Usuario Correcto
“jaznaran”, “”	Usuario Incorrecto
“”, “”	Usuario incorrecto
“Jaznarana”, “george12”	Usuario Correcto

Tabla 13. Pruebas de Login de Usuario

Pruebas de Actualizar Contraseña	
Input	Output
“gustavo12”, “guhek.tweet@gmail.com”, “abc12”, “abc12”	Registro Completo
“jaznaranaRR”, “jaznara@gmail.com”, “12345”, “12345”	Usuario no registrado
“jaznarana”, “12345”, “asdfg”	Campos no coinciden.
“”, “”, “abc”, “”	Campos Incompletos
“apcm”, “”, “12”, “12”	El campo debe comprender más de 3 caracteres.

Tabla 14. Pruebas de Actualizar Contraseña de Usuario

5. DISCUSIÓN DE RESULTADOS

5.1. Contrastación de Hipótesis

Se realizará la contrastación de la hipótesis analizando el resultado de los cuestionarios y observación para cada indicador establecido, utilizando el método de pre-prueba y post-prueba.

Conforme a la tabla de operacionalización de las variables, se analizaron los siguientes indicadores:

5.1.1. Grado de Satisfacción en el Diagnóstico

Se mide la satisfacción promedio (indicador cualitativo) que los clientes obtienen luego de finalizar la consulta a través del chat. Cabe resaltar que este grado de satisfacción es medido luego de que el cliente finaliza la consulta realizada.

5.1.2. Nivel de Facilidad de Uso de la Aplicación

Determina la facilidad con que el usuario puede interactuar y navegar por la aplicación de tal manera que sea eficiente su uso. El tiempo de determinación del nivel de facilidad de uso de la aplicación se mide desde el momento que el usuario ingresa a la aplicación hasta culminar su sesión.

5.1.3. Hipótesis Estadística

Hipótesis H₀: Desarrollar una aplicación móvil si dará soporte a problemas automotrices utilizando la herramienta IBM Watson para la empresa VECARS & TRUCKS S.A.C

$$H_0 \left\{ \begin{array}{l} GS_{PRE} - GS_{POST} \leq 0 \\ NF_{POST} \leq 0 \end{array} \right.$$

Hipótesis H_a: Desarrollar una aplicación móvil no dará soporte a problemas automotrices utilizando la herramienta IBM Watson para la empresa VECARS & TRUCKS S.A.C

$$H_A \left\{ \begin{array}{l} GS_{PRE} - GS_{POST} > 0 \\ NF_{POST} > 0 \end{array} \right.$$

5.1.4. Nivel de Significancia

Nivel de significancia (α) escogido para la prueba de la hipótesis es del 5%.

5.1.4.1. Muestra a Aplicar

La muestra seleccionada fue de 50 personas naturales de forma aleatoria que son además clientes dentro de la Empresa Automotriz Vecars & Trucks S.A.C ubicado en Trujillo.

Se realizó una evaluación con 50 sujetos como grupo muestral. Se realizó de forma aleatoria, en el siguiente cuadro se muestran los rangos segmentados, así como las cantidades y porcentajes correspondientes:

Edades (años)	Cantidad	Porcentaje
24 - 30	19	38%
31 - 36	13	26%
37 - 42	10	20%
43 - 50	8	16%
Total	50	100%

Tabla 15. Distribución de las Muestras

5.1.4.2. Criterios de Exclusión

Un conjunto de clientes de la empresa Vecars & Trucks, seleccionados aleatoriamente que cumplan las siguientes especificaciones:

- Tener entre 24 y 50 años.
- Poseer licencia de conducir.

5.1.4.3. Validación de la Solución

Para obtener información sobre la aplicación de la variable es necesario realizar una evaluación para obtener resultados que puedan

interpretarse de tal manera que pueda evidenciar y a su vez contrastar la validez de la hipótesis.

Se realizó encuesta para la recolección de datos con el formato de valoración del 1 al 5 siendo 1 malo y 5 Excelente, como se puede observar en la siguiente tabla con más detalle:

VALORACIÓN	ESCALA
Excelente	5
Bueno	4
Regular	3
Deficiente	2
Malo	1

Tabla 16. Formato de Valoración

- **Variable Dependiente: Mejora en el Diagnóstico de Problemas Automotrices**

Para obtener las siguientes puntuaciones fue necesario realizar una encuesta con el fin de obtener datos que puedan explotarse, la encuesta mencionada fue enfocada en base a un indicador de satisfacción que nos ayudaría a medir las respuestas de los clientes.

La evaluación como se mencionó se realizó en dos etapas PRE (etapa antes de utilizar la solución) y la POST (etapa luego de utilizar la solución), las preguntas realizadas se hicieron en base a la obtener información sobre la satisfacción del cliente y de esa manera establecer si se obtuvo una mejora en el diagnóstico por medio de las ponderaciones obtenidas de cada pregunta.

1. Califique el proceso que le toma realizar una consulta.

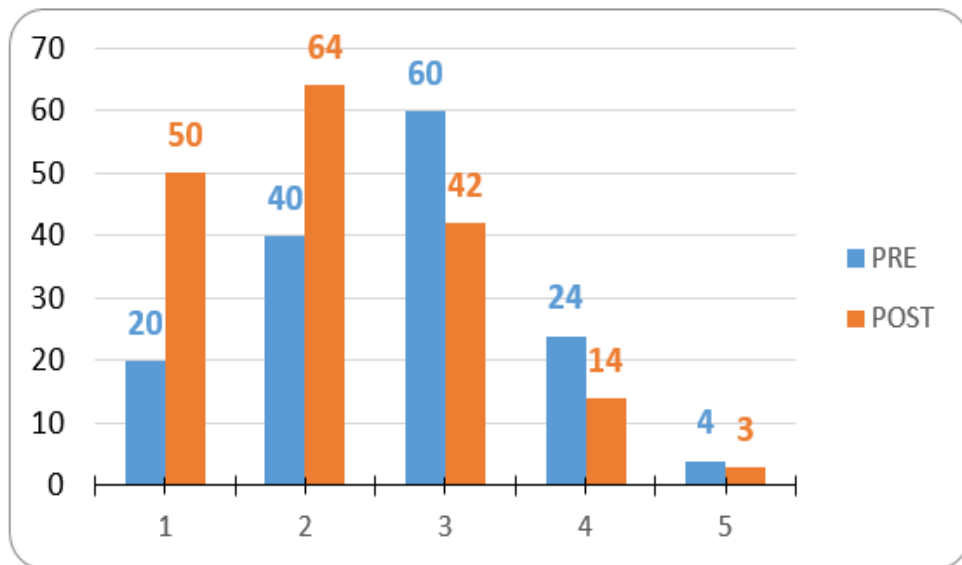


Figura 21. Resultado de Pregunta N° 1, Indicador de Satisfacción del Cliente

2. Califique diagnóstico o solución ofrecida

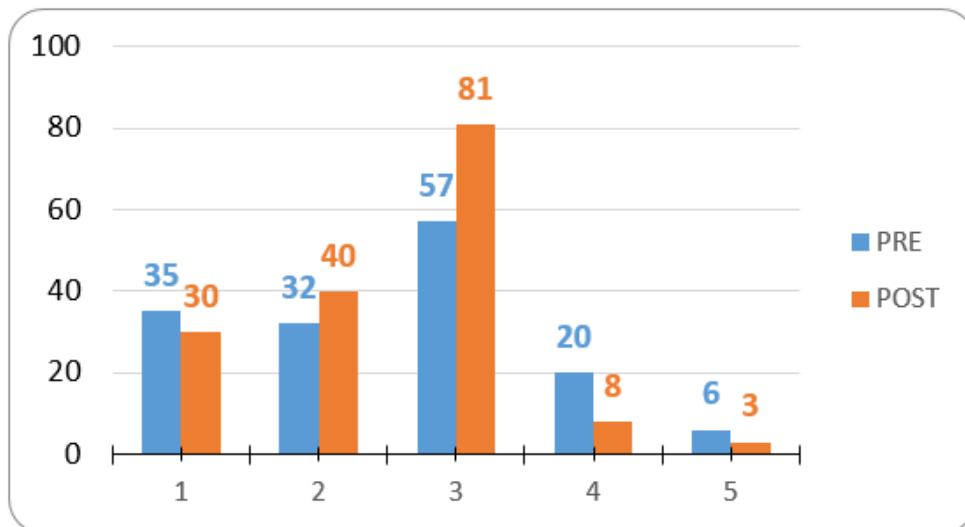


Figura 22. Resultado de Pregunta N° 2, Indicador de Satisfacción del Cliente

3. ¿Qué tan útil resultó el diagnóstico o solución brindada?

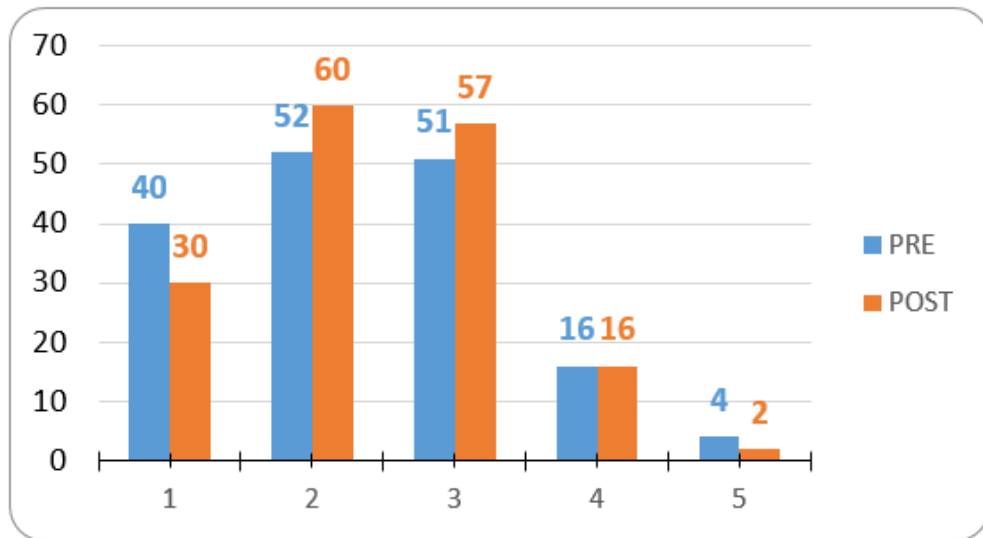


Figura 23. Resultado de Pregunta N° 3, Indicador de Satisfacción del Cliente

4. Califique el desempeño del asistente al momento de dar el diagnóstico.

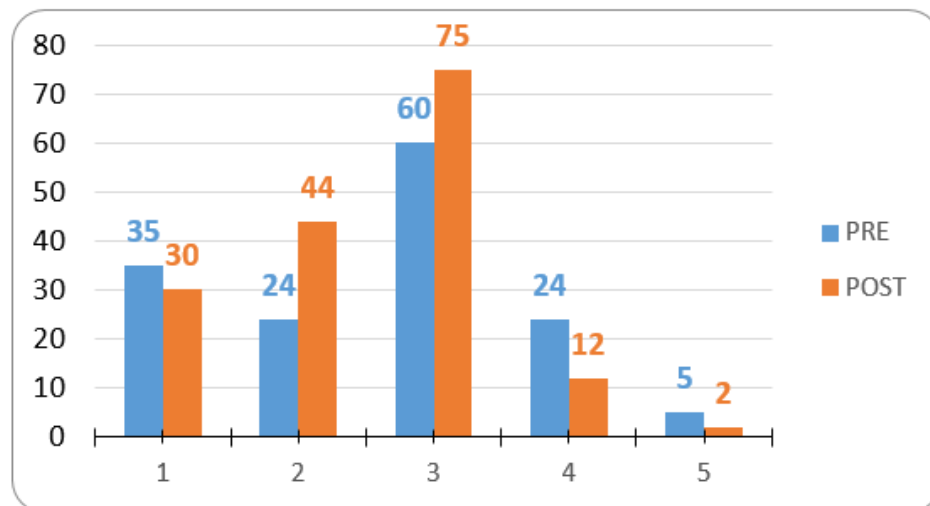


Figura 24. Resultado de Pregunta N° 4, Indicador de Satisfacción del Cliente

5. Como califica el desempeño del asistente al ofrecer una solución, explicó de manera eficiente?

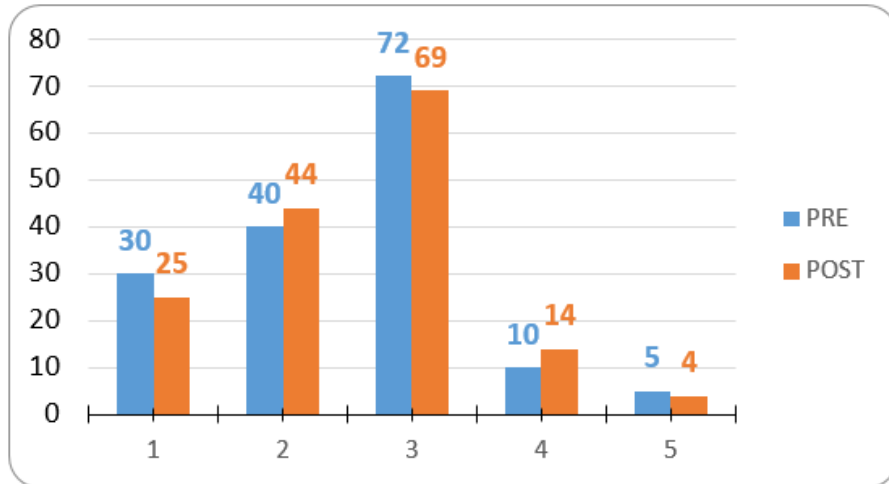


Figura 25. Resultado de Pregunta N° 5, Indicador de Satisfacción del Cliente

- **VARIABLE INDEPENDIENTE: Aplicación Móvil basado en IBM Watson**

Para obtener las siguientes puntuaciones se realizó con el fin de recoger información que pueda explotarse, se enfocó en un indicador en relación a la Variable independiente: La Facilidad de Uso.

Para el indicador de la Variable Independiente: Facilidad de uso se realizó una encuesta en base a Juicio de Expertos para de esta manera obtener los datos correspondientes.

1. Considera efectivo la consulta virtual para la solución del problema

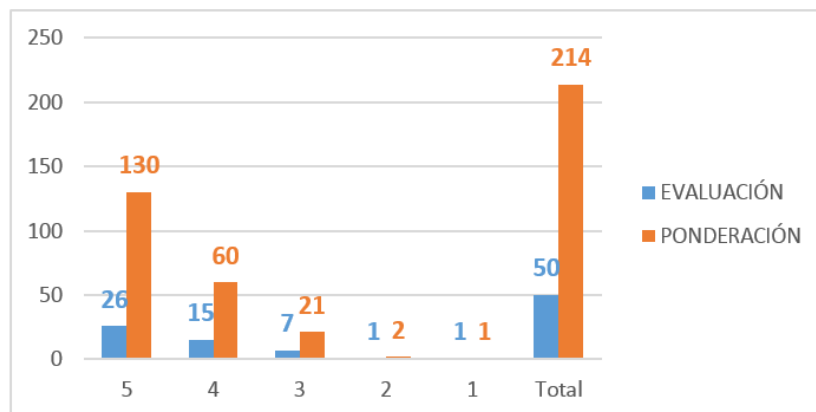


Figura 26. Resultado de Pregunta N° 1, Indicador de Nivel de Facilidad de Uso.

2. Para usted es facil navegar dentro de la aplicacion a nivel especifico

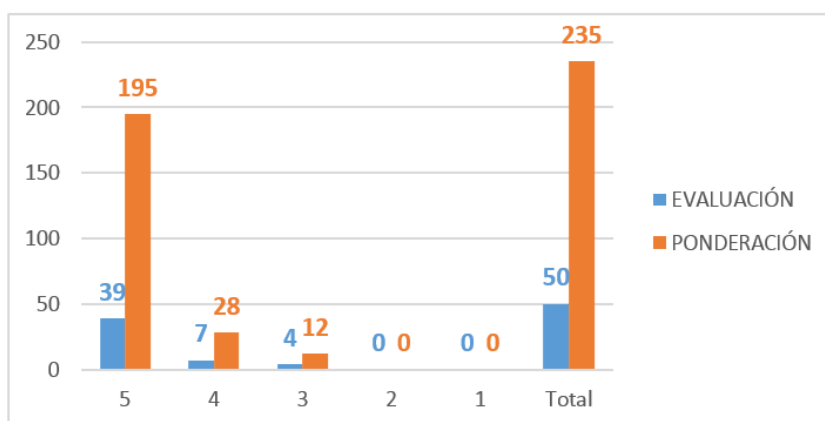


Figura 27. Resultado de Pregunta N° 2, Indicador de Nivel de Facilidad de Uso

3. La aplicacion ofrece guias o instrucciones para manejar facil la aplicacion

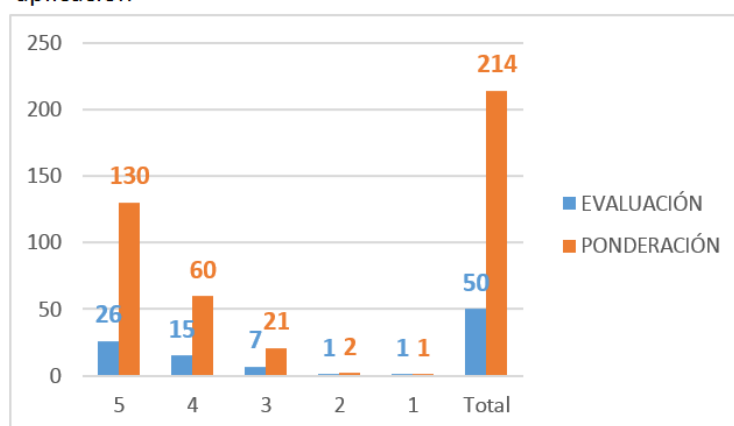


Figura 28. Resultado de Pregunta N° 3, Indicador de Nivel de Facilidad de Uso

4. Los factores tiempo y resultado fueron mejorados con la aplicación.

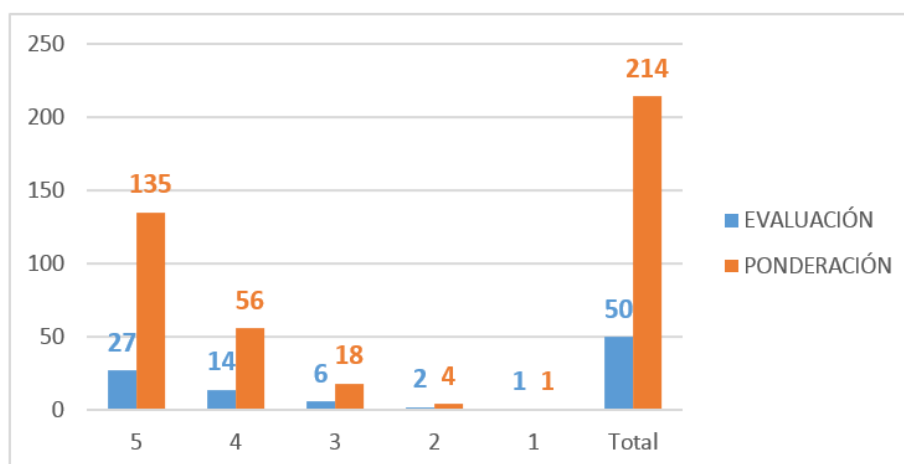


Figura 29. Resultado de Pregunta N° 4, Indicador de Nivel de Facilidad de Uso

5.1.4.4. Análisis de los Resultados obtenidos

De los resultados obtenidos para cada variable a través de sus indicadores se realizó una matriz que resume la recolección de información.

En esta matriz se puede observar las evaluaciones realizadas para ambas variables y la diferencia obtenida entre la evaluación PRE y POST para cada uno de sus indicadores, además de ello se puede obtener la diferencia entre ponderaciones para obtener una mayor claridad de la situación actual.

		VARIABLE DEPENDIENTE			VARIABLE INDEPENDIENTE
		Grado de Satisfacción en el Diagnóstico			Nivel de Facilidad de Uso
ESCALA	PRE	POST	DIFERENCIA	EVALUACIÓN	
Excelente	160	165	-5	615	
Bueno	188	252	-64	196	
Regular	300	324	-24	60	
Deficiente	94	64	30	10	
Malo	24	14	10	3	

Tabla 17. Tabla de Ponderación de Valores Obtenidos.

Para la **Variable Dependiente**, se puede demostrar que los valores son favorecedores luego de haber realizado la evaluación POST con la solución planteada, dando mayor presencia de cambios en la valoración Regular, Bueno y Excelente siendo Bueno el que mayor diferencia de cambio tiene.

Para obtener estos resultados hemos realizado un promedio de cuantas personas han mejorado en cada valoración de la encuesta de realizada a través del siguiente cuadro:

ESCALA	VALORES OBTENIDOS		RESULTADO PORCENTUAL (%)		DIFERENCIA	PROMEDIO
	PRE	POST	PRE	POST		
5	4	10	8	20	12	9,6
4	10	16	20	32	12	
3	20	14	40	28	12	
2	12	7	24	14	10	
1	4	3	8	6	2	
Total	50	50	100	100		

Tabla 18. Resultados de la Variable Dependiente en el Item 1

En este cuadro hecho para la pregunta número 1 de la encuesta; podemos observar que existe una diferencia marcada (5 personas) en los clientes que cambiaron de opinión respecto al proceso que toma realizar la consulta, eso significa que existe un 9.6% de eficacia respecto a esta variable en este ítem.

Realizando el mismo procedimiento para el total de las preguntas se logró tener que un **7%** de manera general como ventaja respecto a la connotación antes de la aplicación, esto quiere decir que la aplicación tuvo un impacto positivo en la empresa.

Respecto a **la Variable Independiente**, se puede considerar una diferencia común respecto a la variable dependiente, siendo el Nivel de Facilidad de Uso el indicador de evaluación se pueden observar la variación que existe dentro de la evaluación realizada utilizando Juicio de expertos se puede apreciar el gran margen de distancia que presentan las valoraciones positivas (Regular, Bueno y Excelente) respecto a las Evaluaciones Negativas (Deficiente y Malo) siendo la diferencia más cercana de 50 puntos entre la distancia de Deficiente a Regular, llegando a obtener resultados de 196 y 615 como valoración de Bueno y Excelente respectivamente.

ESCALA	VALORES OBTENIDOS	PORCENTAJE (%)	VALORACION
	POST	POST	
5	26	52	82
4	15	30	
3	7	14	18
2	1	2	
1	1	2	
Total	50	100	100

Tabla 19. Resultados de la variable independiente en el ítem 1

En este cuadro podemos observar que en el ámbito de facilidad de uso en el ítem de si se considera efectivo la consulta virtual para la solución del problema del cliente hay un 82% el cual piensa que está en un promedio de bueno y excelente, eso significa que hemos tomado en cuenta los principios básicos de usabilidad, navegabilidad, accesibilidad, etc.

De la misma manera continuamos el procedimiento en los ítems restantes teniendo como resultado:

ESCALA	PONDERACION (%)
5	86
4	
3	14
2	
1	
Total	100

Tabla 20. Resultados finales de la variable independiente

Donde podemos observar que un 86% de los clientes sienten y piensan que la aplicación reúne los requisitos necesarios para ser una app fácil de usar.

De esta manera se puede apreciar la mejora significativa obtenida tras el uso de la propuesta de solución.

6. CONCLUSIONES

- Se llevó a cabo una investigación bibliográfica, así como la búsqueda de evidencias de soluciones que guardaban relación con proyecto a desarrollar, encontrando 5 evidencia divididas en 3 Nacionales y 2 Internacionales con el fin de tener un conocimiento más claro de nuestras fortalezas y debilidades.
- El alcance se definió en base a un análisis de requerimientos obteniendo así 7 requerimientos funcionales y 5 requerimientos no funcionales, estableciendo de esta manera el desarrollo en conjunto con la metodología de desarrollo de software ICONIX, un prototipo de aplicación móvil que sirva como un asistente de diagnóstico de soluciones móviles para la empresa.
- Respecto a la Implementación de la Aplicación móvil, la evaluación de la funcionalidad la tenemos al momento de realizar la prueba del prototipo en la muestra, el diagnostico de los posibles fallos lo podemos detectar y sobre todo las soluciones avalados por la encuesta de satisfacción de cliente con un aumento de 9.6%; la viabilidad la evaluamos en los recursos usados; la cuenta de IBM Watson estuvo disponible al 100% y el Android Studio no fue un inconveniente acompañado de ICONIX.
- Después de probar el prototipo, contrastamos que nuestra hipótesis nula no se llegó a cumplir, debido a que la resta de el GS(pre) y el GR(post) es menor a "0" ($GS_{pre}(766) - GS_{post}(819) = -53$), lo cual nos lleva a concluir que la aplicación si nos permite mejorar el diagnostico de problemas automotrices en la empresa VECARS & TRUCKS S.A.C.

7. RECOMENDACIONES

- Se recomienda contar con una versión y cuenta licenciada de IBM, ya que te permite exportar mayor cantidad de datos y contar una versión beta de importación de datos al IBM y facilitar el direccionamiento de información.
- Debido a la dificultad entre la interacción de IBM Watson con Android, se recomienda utilizarse de manera nativa si es el caso de requerirse generar reportes o trabajar con diálogos condicionales que sean más complejos.
- Se recomienda identificar una escala de versiones de compatibilidad al momento de realizar cualquier solución móvil con el fin de tener la posibilidad de abarcar dispositivos móviles de menor gama por versiones del sistema operativo.

8. REFERENCIA BIBLIOGRÁFICAS

(s.f.). *El ComercioQ*.

Acens_Sinfony 3 Framework. (3 de Marzo de 2014). <https://www.acens.com/>. Obtenido de <https://www.acens.com/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf>

Anda_Lucía_es_Digital. (2017). <https://www.significados.com>. Obtenido de <https://www.significados.com/chat/>, 2017

Android Developer. (23 de Octubre de 2018). <https://developer.android.com>. Obtenido de <https://developer.android.com/guide/platform?hl=es>

Arauz. (1998). *Técnicas de Inteligencia Artificial*.

Carrillo Calderón, M. E. (2017). *Agentes virtuales con capacidades cognitivas utilizando IBM Watson*. Madrid.

Castellano Casas, R. (2001).

CASTELLANOS, L. (2012). *Desarrollo de Sistemas de Información*. Editorial Académica Española 1era Edición.

Castillo, L. (2012). <https://conociendogithub.readthedocs.io/en/latest/data/introduccion/>. Obtenido de <https://conociendogithub.readthedocs.io/en/latest/data/introduccion/>

COMPANYRED. (17 de 11 de 2016).

<https://programacionparatodosite.wordpress.com/2016/11/17/que-son-los-frameworks/>. Obtenido de <https://programacionparatodosite.wordpress.com/2016/11/17/que-son-los-frameworks/>

Daniel Bourcier. (2003). *Inteligencia Artificial y Derecho*. UOC.

Darío Propato, J., & Marisa Panizzi, L. T. (2017). *Propuesta de un Modelo de Proceso para Resolver*. Moron.

Developers_Society_Chatfuel. (10 de Agosto de 2015). <https://chatfuel.com/>. Obtenido de <https://chatfuel.com/about-us.html>

Diario TI. (26 de Julio de 2017). <https://diarioti.com/ripley-lanza-eva-asistente-virtual-para-retail-habilitada-por-ibm-watson/105141>. Obtenido de <https://diarioti.com/ripley-lanza-eva-asistente-virtual-para-retail-habilitada-por-ibm-watson/105141>

Digital_GUIde_1&1. (17 de Agosto de 2017). <https://www.1and1.es/digitalguide/paginas-web/disenio-web/ui-que-es-una-interfaz-de-usuario/>. Obtenido de <https://www.1and1.es/digitalguide/paginas-web/disenio-web/ui-que-es-una-interfaz-de-usuario/>

Eyzaguirre Alberca, E. B., & Placencia Roman, C. C. (2018). *Modelo de implementación de tecnología cognitiva sobre intereses de consumo para el sector bancario en Cloud Computing*. Lima.

Fileim, R. (2 de Mayo de 2017). <https://sg.com.mx/>. Obtenido de <https://sg.com.mx/revista/17/sqlite-la-base-datos-embedida>

Fritz Piaget, J. (1868-1980). La Teoría Constructiva de Jean Piaget. En J. Fritz Piaget.

Gastelum Lopez, P. (2007). Ingeniería de Sistemas. En P. Gastelum Lopez.

GESTION. (9 de Octubre de 2017). *GESTION*. Obtenido de <https://gestion.pe/tendencias/management-empleo/aplicaciones-moviles-peru-han-incrementado-63-productividad-empresarial-220422>

GITBOOKS_AjGallego. (2010). <https://ajgallego.gitbooks.io/introduccion-a-android/content/sesion01-apuntes.html>. Obtenido de <https://ajgallego.gitbooks.io/introduccion-a-android/content/sesion01-apuntes.html>

- GITBOOKS_AJGALLEGO. (2010). <https://ajgallego.gitbooks.io/introduccion-a-android/content/sesion01-apuntes.html>. Obtenido de <https://ajgallego.gitbooks.io/introduccion-a-android/content/sesion01-apuntes.html>
- Goleman, D. (1996). *La Inteligencia Emocional*. Vergara.
- Haj-Salej, A. (5 de Marzo de 2017). <https://www.revistagq.com>. Obtenido de <https://www.revistagq.com/noticias/tecnologia/articulos/que-son-exactamente-los-bots-y-como-funcionan/25633>
- Hidalgo, Francisco José. (4 de Octubre de 2016). <https://www.whatsnew.com/>. Obtenido de <https://www.whatsnew.com/2016/10/04/motion-ai-lanza-su-integracion-en-smooch-para-crear-bots-visualmente-y-llevarlos-a-diferentes-canales/>
- Horna Gutiérrez, D. C., & Rodríguez Marquina, F. L. (2016). *Implementación de un Sistema de Información de análisis predictivo para la toma de decisiones en el proceso de Atención médica del Hospital Víctor Lazarte usando la Herramienta de Cognos BI de IBM*. Trujillo.
- Howard, G. E. (2000). *La Teoría de las Inteligencias Múltiples de Gardner*. Paídos. <https://www.blog.andaluciaesdigital.es/chatbots-inteligencia-artificial-para-empresas/>, 2. (2017). <https://www.blog.andaluciaesdigital.es/chatbots-inteligencia-artificial-para-empresas/>, 2017. Obtenido de <https://www.blog.andaluciaesdigital.es/chatbots-inteligencia-artificial-para-empresas/>, 2017
- Ian Sommerville. (2005). *Ingeniería del Software 7ma Edición*. Madrid: PEARSON EDUCATION S.A. Obtenido de <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>: <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>
- IBM Cloud Documents. (26 de Enero de 2018). <https://console.bluemix.net>. Obtenido de <https://console.bluemix.net/docs/services/assistant/index.html#acerca-de>
- LA REPUBLICA. (7 de Abril de 2017). Obtenido de <https://larepublica.pe/tecnologia/856919-arturito-el-bot-del-bcp-tu-servicio>
- Lepage Chumpitaz, C. (2016). *Aplicaciones actuales de la inteligencia artificial y su uso con la tecnología IBM Watson*. Lima.
- MEDIUM_ADM. (23 de Marzo de 2018). <https://medium.com>. Obtenido de <https://medium.com/@activechat/dialogflow-api-ai-to-chatfuel-integration-easy-as-1-2-3-with-integrator-729798e21b8>
- Mendoza Riofrío, M. (05 de Junio de 2017). Al menos 20 grandes empresas locales ya trabajan proyectos de inteligencia artificial. *El Comercio*.
- Merino, M. (12 de Julio de 2014). <http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>. Obtenido de <http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>
- Moncada, A., & Reyes, D. (2015). <http://semanaeconomica.com/caso/watson-de-ibm-la-inteligencia-de-los-negocios-esta-en-la-data/>. Obtenido de <http://semanaeconomica.com/caso/watson-de-ibm-la-inteligencia-de-los-negocios-esta-en-la-data/>
- Muñoz de Frutos, A. (27 de Agosto de 2016). <https://computerhoy.com>. Obtenido de <https://computerhoy.com/noticias/software/que-es-chatbot-50012>
- Oxford_Dictionary. (2016). <https://es.oxforddictionaries.com>. Obtenido de <https://es.oxforddictionaries.com/definicion/artificial>
- Plasencia Prado, C. E. (18 de Diciembre de 2017). *Devcode*. Obtenido de <https://devcode.la/blog/que-es-sql/>

- Ramirez Meson, J. (24 de Noviembre de 2016). <https://www.eoi.es>. Obtenido de <https://www.eoi.es/blogs/redinnovacionEOI/2016/11/24/cognitive-systems-ibm-watson/>
- Ramos, A. I., Herrera, J. A., & Ramirez, M. S. (2009). *Desarrollo de habilidades cognitivas con aprendizaje móvil: un estudio de casos*. MONTERREY.
- Real Academia de la Lengua Española, R. (7 de Marzo de 2007). <https://dle.rae.es>. Obtenido de <https://dle.rae.es/srv/search?m=30&w=artificial>
- Rodriguez, M. (27 de Marzo de 2017). <http://observatorio-ia.com/aplicaciones-de-watson-en-el-sector-sanitario>. Obtenido de <http://observatorio-ia.com/aplicaciones-de-watson-en-el-sector-sanitario>
- Saya Comunicaciones S.A.C. (7 de Julio de 2017). <https://cioperu.pe/articulo/23862/centrum-catolica-implementa-ia-de-ibm-watson/>. Obtenido de <https://cioperu.pe/articulo/23862/centrum-catolica-implementa-ia-de-ibm-watson/>
- Semionóvich Vygotski, L. (1989). *Interacción entre Aprendizaje y Desarrollo*. SIGNIFICADOS. (2017). <https://www.significados.com/chat/>, 2017. Obtenido de <https://www.significados.com/chat/>, 2017
- Sommerville, Ian. (2005). *Ingeniería del Software 7ma Edición*. Madrid: PEARSON EDUCATION S.A. Obtenido de <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>: <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>
- Stupia, Pamela. (12 de Mayo de 2015). <https://www.itsitio.com>. Obtenido de <https://www.itsitio.com/us/que-es-bluemix-de-ibm-2/?mdx-ed=Sm9YRkgvcmtVTHF5MWY2MGgyUWxidz09>
- Suja, A. (13 de Abril de 2018). <https://ecommerce-platforms.com/>. Obtenido de <https://ecommerce-platforms.com/es/interviews/botsify-interview>
- Vanegas, H., Marin, A., & Vargas, W. (2007). <http://soda.ustadistancia.edu.co>. Obtenido de http://soda.ustadistancia.edu.co/enlinea/Actividades_Chat-WEB/concepto_de_chat.html
- Volaya, Hub. (Abril de 2016). <http://volaya.github.io>. Obtenido de http://volaya.github.io/libro-sig/chapters/Bases_datos.html
- Wikibooks_Anon. (22 de Febrero de 2016). https://es.wikibooks.org/wiki/Las_Interfaces_de_Usuario/Cap%C3%ADtulo_1:_Introduccion_a_las_Interfaces_de_Usuario. Obtenido de https://es.wikibooks.org/wiki/Las_Interfaces_de_Usuario/Cap%C3%ADtulo_1:_Introduccion_a_las_Interfaces_de_Usuario

ANEXOS

ENCUESTA PARA EVALUAR SATISFACCIÓN DEL CLIENTE

Estimado usuario: El siguiente documento tiene como fin, darnos una idea de su opinión respecto a la Aplicación lanzada por VECARS & TRUCKS S.A.C, para poder mejorar su experiencia con nosotros. Gracias por su tiempo.

***Obligatorio**

INSTRUCCIONES: Marcar con una X dentro de la casilla que usted crear correspondiente para cada una de las preguntas propuestas.

La valoración es en rango del 1 al 5, siendo 1 malo y 5 excelente.

1. Califique el proceso que le toma realizar una consulta. *

Marca solo un óvalo.

1 2 3 4 5

2. Califique diagnóstico o solución ofrecida. * Marca solo

un óvalo.

1 2 3 4 5

3. ¿Qué tan útil resultó el diagnóstico o solución brindada? * Marca solo un óvalo.

1 2 3 4 5

4. Califique el desempeño del asistente al momento de dar el diagnóstico. * Marca solo un óvalo.

1 2 3 4 5

5. ¿Cómo califica el desempeño del asistente al ofrecer una solución, explicó de manera eficiente? *

Marca solo un óvalo.

1 2 3 4 5

6. Qué considera que podría mejorarse dentro de la Aplicación?

NIVEL DE FACILIDAD DE USO

Estimado usuario: El siguiente documento tiene como fin, darnos una idea de su opinión respecto a la Aplicación lanzada por VECARS & TRUCKS S.A.C, para poder mejorar su experiencia con nosotros. Gracias por su tiempo.

INSTRUCCIONES: Marcar con una X dentro de la casilla que usted crear correspondiente para cada una de las preguntas propuestas.

La valoración es en rango del 1 al 5, siendo 1 malo y 5 excelente.

7. Considera efectivo la consulta virtual para la solución del problema *Marca solo un óvalo.*

1 2 3 4 5

8. Para usted es fácil navegar dentro de la aplicación a nivel específico *Marca solo un óvalo.*

1 2 3 4 5

9. La aplicación ofrece guías o instrucciones para manejar fácil la aplicación *Marca solo un óvalo.*

1 2 3 4 5
