



Desarrollo de un sistema de análisis de textos basado en Procesamiento de Lenguaje Natural y servicios en la nube para determinar delitos de telecomunicaciones

Motoche Macas, Byron Patricio y Villamarin Pazmiño, David Alejandro

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

Ing. Alulema Flores, Darwin Omar PhD.

18 de Agosto del 2021



Document Information

Analyzed document	Tesis_Motoche_Villamarin_Urkund.pdf (D111314089)	
Submitted	8/15/2021 11:25:00 PM	
Submitted by	Alulema Flores Darwin Omar	 <small>Primado a autorización por:</small> DARWIN OMAR ALULEMA FLORES
Submitter email	doalulema@espe.edu.ec	
Similarity	0%	
Analysis address	doalulema.espe@analysis.orkund.com	

Sources included in the report

SA	TESIS salazar'herreraurk.docx Document TESIS salazar'herreraurk.docx (D54836735)	 1
W	URL: https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/624859/Macha_tg.pdf?sequence=1&isAllowed=y Fetched: 5/30/2020 1:31:33 AM	 1
SA	TESIS PABLO JARAMILLO ABRIL 6 de febrero.docx Document TESIS PABLO JARAMILLO ABRIL 6 de febrero.docx (D63662114)	 2
SA	Asistente de voz para consulta a base de datos-TESIS.pdf Document Asistente de voz para consulta a base de datos-TESIS.pdf (D79142238)	 1
W	URL: https://1library.co/document/zwvoe5lq-implementacion-metodo-procesamiento-lenguaje-natural-voicebot-cultivo-maiz.html Fetched: 3/21/2021 6:45:20 PM	 1



**DEPARTAMENTO DE ELECTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**Desarrollo de un sistema de análisis de textos basado en Procesamiento de Lenguaje Natural y Servicios en la Nube para determinar delitos de telecomunicaciones**” fue realizado por los señores **Motoche Macas, Byron Patricio y Villamarin Pazmiño, David Alejandro**, el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 16 de agosto del 2021

Firma:



.....
Ing. Alulema Flores, Darwin Omar, PhD.

C. C 1002493334



**DEPARTAMENTO DE ELECTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Motoche Macas, Byron Patricio y Villamarin Pazmiño, David Alejandro**, con cédulas de ciudadanía n°1725205643 y 1722168240, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Desarrollo de un sistema de análisis de textos basado en Procesamiento de Lenguaje Natural y Servicios en la Nube para determinar delitos de telecomunicaciones”**, es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 16 de agosto del 2021

Firma

Motoche Macas, Byron Patricio

C.C.: 1725205643

Villamarin Pazmiño, David Alejandro

C.C.: 1722168240



**DEPARTAMENTO DE ELECTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, **Motoche Macas, Byron Patricio y Villamarin Pazmiño, David Alejandro**, con cédulas de ciudadanía n° 1725205643 y 1722168240, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “**Desarrollo de un sistema de análisis de textos basado en Procesamiento de Lenguaje Natural y Servicios en la Nube para determinar delitos de telecomunicaciones**”, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 16 de agosto del 2021

Firma

Motoche Macas, Byron Patricio

C.C.: 1725205643

Villamarin Pazmiño, David Alejandro

C.C.: 1722168240

Dedicatoria

Dedico este proyecto a mi familia, a mis padres Alberto y María Catalina, a mis hermanos Melissa y Cristhian por haberme apoyado en cada momento a lo largo de mi carrera universitaria con sus consejos, enseñanzas y palabras de aliento en los instantes precisos. Sus valores, determinación y pasión en sus actividades diarias siempre han sido un motivo más para alcanzar mis metas. Además, dedico de forma especial todo este trabajo y esfuerzo a mi pequeño yo, mi hijo Jahir, quien cambio de forma inesperada mi vida y la llena de alegrías cada día.

Motoche Macas Byron Patricio

A ustedes mi gran familia, mis padres Doris y Mauricio que siempre me han apoyado, me formaron con reglas y libertades, su soporte y motivación han sido fundamentales para ser quien soy ahora. A mi hermana Dennise que me enseñó el significado de la verdadera amistad, su apoyo y confianza han sido esenciales durante toda mi vida y me han permitido alcanzar mis metas.

Villamarin Pazmiño David Alejandro

Agradecimiento

Agradezco en primer lugar a mi madre, María Catalina, por todo el amor y apoyo a largo de esta carrera, por cada desayuno, almuerzo y cena, dándome las energías para culminar este ciclo en mi vida. A mi padre, por todo el esfuerzo y sacrificio que ha realizado para que nunca me falte nada durante mis estudios.

Agradezco de forma especial a mi hermano Cristhian que siempre ha sido un gran ejemplo a seguir y me ha ayudado siempre con sus enseñanzas. A mi hermana Melissa por motivarme a cada día siempre ser mejor y a disfrutar la vida al máximo.

Agradezco al Ing. Darwin Alulema, Director del presente proyecto de investigación, por toda su ayuda y asistencia prestada para las revisiones y el desarrollo del proyecto.

Agradezco especialmente a mi amigo y compañero de tesis, David Villamarín, por motivarme a dar cada día el máximo para completar este proyecto y por su gran esfuerzo y esmero para culminarlo de la mejor manera.

Para finalizar agradezco a mis amigos de carrera y vida universitaria a lo largo de estos siete años Gabriel, Johita, Magus, Jazmín, Raymi, David, Stalin, Carlitos, Boris y Romina. ¡Gracias Totales!

Motoche Macas Byron Patricio

Agradecimiento

Agradezco a mis padres Doris y Mauricio por todos los esfuerzos realizados para que pueda culminar esta etapa de mi vida, por sus enseñanzas, su cariño y el constante ejemplo de superación que me dan todos los días. A mi hermana que siempre me ha brindado su apoyo y ayuda a lo largo de mi vida, porque los lazos que hemos creado perduren en el tiempo, a ustedes por ser mi familia y mi soporte en las buenas y en las malas.

A mi amigo Byron por todas las noches sin dormir, su dedicación y esfuerzo, por la constante y mutua motivación, que la vida nos permita seguir creciendo juntos.

A Maria Augusta por el cariño y apoyo incondicional, por todo el amor y el tiempo juntos.

A mi tutor el ingeniero Darwin Alulema, por su apoyo y asistencia durante este tiempo, por toda la ayuda y recomendaciones realizadas en pro de este trabajo.

A los grandes amigos que la Universidad me dió, por todas las risas, buenas y malas experiencias que hemos vivido juntos, porque han hecho del camino una experiencia inolvidable.

Finalmente quiero agradecerme a mi, por todo el esfuerzo, ganas y dedicación que he puesto en este trabajo y a lo largo de la carrera, que sea muestra fehaciente que todo se puede conseguir con paciencia y trabajo duro.

Villamarin Pazmiño David Alejandro

Tabla de contenido

Urkund	2
Certificación	3
Responsabilidad de Autoría	4
Autorización de Publicación.....	5
Dedicatoria.....	6
Agradecimiento.....	7
Índice de Tablas	12
Índice de Figuras	13
Resumen	16
Abstract.....	17
Capítulo I.....	18
Introducción.....	18
Antecedentes	18
Justificación.....	19
Alcance.....	21
Objetivos.....	22
General.....	22
Específicos.....	22
Estado del Arte.....	22
Mapeo Sistemático de la Literatura.....	23
Revisión Sistemática de la Literatura.....	35
Metodología.....	48
Organización del contenido	49
Capítulo II.....	50
Marco Teórico.....	50
Normas Jurídicas.....	50
Código Orgánico Integral Penal	50
Ley Orgánica de Telecomunicaciones.....	50
Delitos de Telecomunicaciones	51
Legal Tech	69
Aplicaciones de la Tecnología en el Campo Legal.....	70

	10
Beneficios de la tecnología aplicada al Campo Legal.....	71
Ejemplos de Legal Tech en el Mundo	71
Legal Tech en el Ecuador.....	72
Cloud Computing	74
Definición	74
Características.....	75
Modelos de Servicio en la nube	77
Modelos de Implementación	80
Proveedores en la nube	82
Procesamiento del Lenguaje Natural.....	88
Definición	88
Plataformas de PLN en la Nube	89
Características de cada Plataforma.....	90
Arquitectura Orientada a Servicios.....	96
Definición	96
Características de SOA	97
Roles en SOA	98
Arquitectura de Microservicios.....	99
REST.....	102
Capítulo III	105
Desarrollo e Implementación	105
Arquitectura del Sistema	106
Backend.....	110
Base de Datos de Artículos	110
Modelos de Procesamiento de Lenguaje Natural.....	116
Asistente de Texto	147
Asistente de Voz.....	164
Base de Datos de la Aplicación Web y Móvil	168
Servicio WCF	174
Servidor	176
Frontend	177

	11
Interfaz web	179
Interfaz móvil	191
Capítulo IV.....	202
Resultados.....	202
Metodología.....	202
Métricas de los Modelos.....	202
Tiempo de respuesta	206
Pruebas de Carga y Esfuerzo.....	207
Pruebas de Usabilidad.....	209
Pruebas Cuantitativas	210
Métricas de los Modelos.....	210
Tiempo de Respuesta.....	222
Pruebas de Carga y Esfuerzo	228
Pruebas Cualitativas.....	235
Capítulo V.....	238
Conclusiones y Trabajos Futuros	238
Conclusiones	238
Trabajos Futuros	242
Referencias.....	244
Anexos.....	252

Índice de Tablas

Tabla 1 <i>Comparación de las plataformas en la nube</i>	87
Tabla 2 <i>Diferencias SOA y Arquitectura de Microservicios</i>	101
Tabla 3 <i>Métricas de cada artículo para el Modelo de AWS</i>	211
Tabla 4 <i>Métricas de cada artículo para el Modelo de Microsoft Azure</i>	213
Tabla 5 <i>Métricas de cada artículo para el Modelo de IBM Watson</i>	215
Tabla 6 <i>Resultados de las pruebas de carga de los servicios de NLP y el Asistente de Watson</i> ...	228
Tabla 7 <i>Resultados de las pruebas de esfuerzo de los servicios de NLP y el Asistente de Watson</i>	229

Índice de Figuras

Figura 1	<i>Procedimiento seguido para el SMS</i>	24
Figura 2	<i>Estudios seleccionados en las fases del SMS</i>	28
Figura 3	<i>Publicaciones por año del PLN</i>	29
Figura 4	<i>Publicaciones del PLN en Cloud</i>	30
Figura 5	<i>Publicaciones por País Fase 2</i>	31
Figura 6	<i>Proceso seguido para el SLR</i>	37
Figura 7	<i>Artículos consultados en cada una de las fases del SLR</i>	42
Figura 8	<i>Número de artículos que relacionan NLP, cloud y área legal por año</i>	43
Figura 9	<i>Porcentaje de documentos por área que relacionan NLP, cloud y el área legal</i>	44
Figura 10	<i>Fraude por clonación de celulares</i>	54
Figura 11	<i>Establecimiento del call back paso 1</i>	55
Figura 12	<i>Establecimiento del call back paso 2</i>	56
Figura 13	<i>Establecimiento del call back paso 3</i>	57
Figura 14	<i>Fraude tercer país</i>	58
Figura 15	<i>Elementos de un sistema By pass</i>	61
Figura 16	<i>Ruta legal de una llamada Internacional</i>	62
Figura 17	<i>Ruta By pass ilegal</i>	64
Figura 18	<i>LegalTech en el Ecuador: Principales nichos</i>	74
Figura 19	<i>Diferencias entre los modelos de servicio en la nube</i>	79
Figura 20	<i>Cuadrante mágico para CIPS</i>	82
Figura 21	<i>Cuota de Mercado de IaaS y PaaS en el 2019</i>	83
Figura 22	<i>Cuota de Mercado de SaaS en el 2019</i>	84
Figura 23	<i>Componentes en SOA</i>	99
Figura 24	<i>REST API</i>	103
Figura 25	<i>Principios API REST</i>	104
Figura 26	<i>Arquitectura del Sistema</i>	106
Figura 27	<i>Detalle de tabla Artículos</i>	110
Figura 28	<i>Web API Template para crear un Servicio RESTful en Visual Studio Community</i>	113
Figura 29	<i>Ejemplo de Anotaciones y Relaciones para modelos supervisados</i>	118
Figura 30	<i>Flujo de trabajo de la creación del modelo de machine learning</i>	119
Figura 31	<i>Entidades creadas en Watson Knowledge Studio</i>	120
Figura 32	<i>Relaciones creadas en Watson Knowledge Studio</i>	121
Figura 33	<i>Documentos usados en Knowledge Studio</i>	122
Figura 34	<i>Versiones del modelo generado a través del tiempo</i>	124
Figura 35	<i>Estadísticas por Entidad</i>	125
Figura 36	<i>Matriz de confusión de entidades</i>	125
Figura 37	<i>Conjunto de documentos usados para entrenar y probar el modelo</i>	126
Figura 38	<i>Módulos generales modelo de Amazon Comprehend</i>	128
Figura 39	<i>Clasificación personalizada de Amazon Comprehend</i>	130

Figura 40	<i>Amazon Comprehend análisis en tiempo real</i>	131
Figura 41	<i>Detalles del modelo clasificador de Amazon Comprehend</i>	132
Figura 42	<i>Rendimiento del clasificador de Amazon Comprehend</i>	134
Figura 43	<i>Detalles del endpoint del modelo clasificador de Amazon Comprehend</i>	135
Figura 44	<i>Ciclo de Iteración para el diseño de aplicaciones con LUIS</i>	137
Figura 45	<i>Expresión de ejemplo y anotación de entidades en LUIS</i>	139
Figura 46	<i>Métricas Modelo en Microsoft Azure</i>	142
Figura 47	<i>Predicciones por cada artículo en Microsoft Azure</i>	143
Figura 48	<i>API creado para consultar los modelos las diferentes plataformas</i>	145
Figura 49	<i>Diagrama de Watson Assistant</i>	148
Figura 50	<i>Habilitar uso de Webhooks</i>	152
Figura 51	<i>Configuración Webhook</i>	152
Figura 52	<i>Diálogo Bienvenido</i>	155
Figura 53	<i>Etiquetas del diálogo Opciones</i>	156
Figura 54	<i>Etiquetas del diálogo Buscar Delito</i>	156
Figura 55	<i>Petición del texto en diálogo Buscar Delito</i>	157
Figura 56	<i>Ejemplo de ingreso de texto en diálogo Buscar Delito</i>	158
Figura 57	<i>Diálogo Consulta Artículo</i>	159
Figura 58	<i>Diálogo Consulta Artículo</i>	160
Figura 59	<i>Diálogo Acerca de Nosotros</i>	161
Figura 60	<i>Diálogo Standby After Acerca</i>	161
Figura 61	<i>Integración del servicio de Watson Assistant</i>	162
Figura 62	<i>Personalización del asistente</i>	163
Figura 63	<i>Detalle de tabla Usuarios</i>	168
Figura 64	<i>Detalle de tabla Estadísticas</i>	170
Figura 65	<i>Detalle de tabla Encuesta</i>	172
Figura 66	<i>Detalle de tabla Pruebas Tiempos</i>	173
Figura 67	<i>Pantalla de inicio de la aplicación web</i>	179
Figura 68	<i>Pantalla de registro de la aplicación web</i>	180
Figura 69	<i>Menú lateral establecido en la Master Page</i>	182
Figura 70	<i>Página de error de la aplicación web</i>	182
Figura 71	<i>Asistente de texto en la página de Inicio</i>	183
Figura 72	<i>Asistente de Voz en página auxiliar</i>	184
Figura 73	<i>Contenido del tab Estadísticas Modelos de la página de Resultados</i>	185
Figura 74	<i>Contenido del tab Rendimiento de la página de Resultados</i>	186
Figura 75	<i>Contenido del tab Experiencia de Usuario de la página de Resultados</i>	187
Figura 76	<i>Contenido de la página Encuesta</i>	188
Figura 77	<i>Contenido de la página Ayuda con el Manual de Usuario</i>	189
Figura 78	<i>Contenido de la página Info</i>	190
Figura 79	<i>Pantalla de inicio de la aplicación móvil</i>	192
Figura 80	<i>Pantalla de registro de la aplicación móvil</i>	192
Figura 81	<i>Menú lateral establecido en la Master Detail Page de la aplicación móvil</i>	194

Figura 82	<i>Asistente de texto en aplicación móvil</i>	195
Figura 83	<i>Asistente de Voz en aplicación móvil</i>	196
Figura 84	<i>Contenido del tab Modelos de la página de Resultados de la aplicación móvil</i>	196
Figura 85	<i>Contenido del tab Rendimiento de la página de Resultados de la aplicación móvil</i>	197
Figura 86	<i>Contenido del tab UX de la página de Resultados de la aplicación móvil</i>	198
Figura 87	<i>Contenido de la página Encuesta en la aplicación móvil</i>	199
Figura 88	<i>Contenido de la página Ayuda en la aplicación móvil</i>	200
Figura 89	<i>Contenido de la página Info en la aplicación móvil</i>	201
Figura 90	<i>Matriz de confusión 2x2</i>	203
Figura 91	<i>Reducción de matriz de confusión 16x16 a 2x2</i>	205
Figura 92	<i>Matriz de confusión del Modelo en AWS</i>	210
Figura 93	<i>Matriz de confusión del Modelo en Azure</i>	213
Figura 94	<i>Matriz de confusión del modelo en IBM Watson</i>	215
Figura 95	<i>Comparativa de la exactitud de los Modelos</i>	218
Figura 96	<i>Comparativa de la precisión de los Modelos</i>	219
Figura 97	<i>Comparativa de la exhaustividad de los Modelos</i>	220
Figura 98	<i>Comparativa del F1 Score de los Modelos</i>	221
Figura 99	<i>Comparativa tiempo de respuesta del modelo en AWS en horarios de prueba</i>	222
Figura 100	<i>Comparativa tiempo de respuesta del modelo en Microsoft Azure en horarios de prueba</i>	223
Figura 101	<i>Comparativa tiempo de respuesta del modelo en IBM Watson en horarios de prueba</i>	225
Figura 102	<i>Comparativa tiempos de respuesta de los tres modelos</i>	227
Figura 103	<i>Peticiones simultáneas exitosas de los servicios de NLP y el Asistente de Watson</i>	231
Figura 104	<i>Peticiones no exitosas de los servicios de NLP y el Asistente de Watson</i>	232
Figura 105	<i>Tiempo de respuesta promedio durante las pruebas de esfuerzo</i>	233
Figura 106	<i>Resultado Encuestas SUS</i>	235
Figura 107	<i>Representación de los resultados de un SUS</i>	236

Resumen

El desarrollo tecnológico permite que diversas plataformas oferten servicios en la nube, donde destacan los servicios de inteligencia artificial como el procesamiento de lenguaje natural (PLN), la traducción de texto a voz y los asistentes virtuales. Así mismo, los delitos informáticos han experimentado un crecimiento, en Ecuador han sido presentadas 53463 denuncias sobre delitos informáticos entre 2014 y 2020. No obstante, la persecución de los delitos puede resultar compleja y retrasar a los organismos de justicia e incluso puede conducir a la prescripción de los mismos. En este trabajo, se propone una aplicación con una arquitectura orientada a servicios para el análisis de procesos judiciales mediante el uso de un asistente virtual de voz o texto, relacionado posibles delitos informáticos y de telecomunicaciones con la normativa jurídica, a través de modelos de PLN desarrollados en Amazon Web Services (AWS), Microsoft Azure e IBM Watson. Para el entrenamiento de los modelos se recopiló información de diversas fuentes como la Ley Orgánica de Telecomunicaciones, el Código Orgánico Integral Penal y diversos procesos judiciales. Los modelos entrenados han sido comparados y evaluados obteniendo que el modelo de Azure presenta los mejores resultados con un F1 Score de 0.8838, un tiempo de respuesta promedio de 251.8 ms y soporta hasta 80 peticiones simultaneas. El modelo de AWS con un F1 score de 0.8588, tiempo promedio de respuesta 327.43 ms soportando hasta 50 peticiones simultáneas. Luego, el modelo de IBM con un F1 Score de 0.7669, un tiempo promedio de 365.15 ms tolerando hasta 50 peticiones simultáneas.

Palabras Clave:

- **PROCESAMIENTO DE LENGUAJE NATURAL (PLN)**
- **DELITOS INFORMATICOS Y DE TELECOMUNICACIONES**
- **SERVICIOS EN LA NUBE**

Abstract

Technological development allows cloud platforms to offer new web services, where artificial intelligence services stand out, such as natural language processing (NLP), text-to-speech translation and virtual assistants. In addition, computer crimes have increased too, in Ecuador 53,463 complaints about computer and telecommunications crimes have been made between 2014 and 2020. Although, the prosecution of these crimes can be complex and demand a lot of time, delaying law enforcement and even leading to the prescription of them. In this work, an application with a service-oriented architecture is proposed for the analysis of judicial processes using a virtual voice or text assistant, relating possible computer and telecommunications crimes with the corresponding legal regulations, using NLP models developed on Amazon Web Services (AWS), Microsoft Azure, and IBM Watson. The information used to train the models was collected from various sources such as Ecuadorian legal regulations (LOT and COIP) and judicial processes related to computer and telecommunications crimes. The trained models have been compared and evaluated based on classification and performance metrics, obtaining that the model trained in Azure shows the best results with an F1 Score of 0.8838, an average response time of 251.8 ms and supports up to 80 requests simultaneously. Followed by the AWS model with an F1 score of 0.8588, an average response time of 327.43 ms supporting up to 50 simultaneous requests. Finally, the IBM model with an F1 Score of 0.7669, an average response time of 365.15 ms supporting up to 50 simultaneous requests.

Keywords:

- **NATURAL LANGUAGE PROCESSING (NLP)**
- **COMPUTER AND TELECOMMUNICATION CRIMES**
- **CLOUD SERVICES**

Capítulo I

Introducción

Antecedentes

El constante avance de la tecnología en el área de las Tecnologías de la Información y Comunicación (TICs) han permitido el crecimiento y desarrollo de los servicios alojados en la nube. Este paradigma permite el uso de diferentes productos de seguridad, computación, base de datos, machine learning y almacenamiento, provistos como servicios (Amazon Web Services, Inc., 2020). En este contexto, las plataformas actuales permiten integrar múltiples servicios desarrollados con distintas tecnologías, bajo el concepto de Software como Servicio (en inglés, Software as a Service).

Además, cuando SaaS se implementa en una arquitectura orientada a servicios (en inglés, Service Oriented Architecture), permite definir la utilización de servicios más robustos. La integración de estos servicios conlleva a la definición clara de los mismos, los cuales buscan brindar soporte a diferentes aplicaciones dentro de un sistema. Este tipo de arquitectura permite la creación de sistemas altamente escalables, brindando una forma bien definida de exposición e invocación de servicios (Centro Europeo de Postgrado, 2019).

Estas arquitecturas requieren un diseño estructurado, el cual permite que pequeños servicios, puedan coordinarse entre sí ofreciendo un servicio conjunto de mayor envergadura. En tal sentido existen grandes compañías dedicadas a brindar servicios en la nube, mediante las cuales se crea, administra e implementa aplicaciones a través de internet, volviéndolas mundialmente accesibles y fáciles de implementar, por ejemplo, Microsoft Azure, Amazon Web Services e IBM Watson.

Entre los servicios ofertados, se dispone de infraestructuras (almacenamiento, redes) y plataformas (bases de datos, sistemas de gestión de contenidos) entre muchos otros, sin embargo y debido a los grandes volúmenes de datos disponibles que los sistemas deben procesar actualmente, se destacan los servicios orientados al procesamiento de lenguaje natural (PLN) los cuales a través del “análisis de textos permiten extraer metadatos de contenido, como conceptos, entidades, palabras clave, categorías, sentimientos, emociones, relaciones y roles semánticos utilizando la comprensión del lenguaje natural” (IBM Watson) permitiendo así comunicar a los seres humanos con las máquinas.

Sin embargo, los delitos informáticos también se han incrementado, los cuales afectan la información y los datos como bienes jurídicos protegidos. Por otro lado, los delitos en telecomunicaciones implican actividades criminales tales como robos o hurto, fraudes, falsificaciones, perjuicios, estafa y sabotaje que afectan a usuarios, operadores de telecomunicaciones y proveedores de servicios (Llangarí Salazar , 2016). En este sentido, en Ecuador varios delitos están relacionados con las TICs y las telecomunicaciones. Destacándose principalmente el acceso no consentido a un sistema informático, telemático o de telecomunicaciones, el ataque a la integridad de sistemas informáticos, la interceptación ilegal de datos y la revelación ilegal de bases de datos (Primicias, 2019). La persecución de estos delitos puede resultar compleja y retrasar a los organismos de justicia en el tratamiento de los procesos e incluso puede conducir a la prescripción o caducidad de estos.

Justificación

En el siglo XVII el filósofo Leibniz, planteó que la jurisprudencia, debía estar sustentada en el método matemático y más específicamente en el método de la geometría euclidiana

(Hacking, 1995), basándose así en axiomas y teoremas, concluyendo que la validez de una conclusión está sostenida por su rigor lógico y no debido a evidencias empíricas, proponiendo así adoptar un método científico. Leibniz pensaba que la única forma de conseguir un sistema legal justo se obtendría reduciendo la cantidad de normas contradictorias, imprecisas y cambiantes a ciertos principios fundamentales, a través de los cuales, siguiendo un razonamiento lógico, se derivarían las reglas necesarias para resolver cualquier tipo de situaciones de la vida social (Berkowitz, 2005).

El método científico y matemático propuesto por Leibniz busca eliminar en mayor medida la arbitrariedad, incertidumbre y contradicción que existe en los océanos de normas legales, buscando así que las decisiones judiciales, bajo los parámetros de la ciencia, tengan un orden racional y respondan a criterios justos a causa de esto, recuperando la esencia ética del Derecho, buscando lo correcto en todos los casos, han transcurrido cerca de cuatro siglos y el Derecho sigue alejado de la justicia, muchas veces desconectado de criterios morales, las soluciones legales son lógicas, pero injustas, ineficientes e insatisfactorias (Escobar, 2015).

En este sentido pretendemos en nuestro proyecto aportar con una herramienta que permita analizar de forma automática procesos judiciales, por medio de un sistema basado en una arquitectura orientada a servicios y PLN que permita analizar los procesos judiciales, buscando lograr apartar en mayor medida la injusticia del Derecho, reemplazando de alguna medida la subjetividad de una persona que está sujeta a arbitrariedades o pueda su criterio verse afectado por las condiciones socioculturales en las que se encuentra, buscando así automatizar los procesos judiciales, enfocado específicamente en delitos informáticos y relacionados a las telecomunicaciones.

Este sistema busca aplicar las tecnologías actuales que se encuentran en crecimiento como son la inteligencia artificial y los servicios de computación en la nube (Cloud Computing) y aplicar las mismas en el ámbito del Derecho, ofreciendo una convergencia de tecnologías para automatizar la toma de decisiones y la resolución de problemas.

Este sistema comprende varias etapas y procesos a cumplir entre las cuales se encuentran inicialmente la búsqueda de información, referente a los procesos judiciales en el ámbito de las telecomunicaciones y los delitos informáticos obtenidos de la función judicial del Ecuador, por lo tanto, el estudio legal en este ámbito es imprescindible para el correcto desarrollo del proyecto de investigación.

Adicionalmente el uso de servicios y tecnologías en la nube, los cuales permitirán para virtualizar los servidores del sistema, así como realizar los procesos de inteligencia artificial, mediante el uso aprendizaje de máquina (machine learning) se busca entrenar al sistema, el cual permitirá analizar los cuerpos textuales que puedan generar información importante para determinar normas que se ajusten a las conductas tipificadas por los usuarios.

El sistema será generado a partir de una arquitectura basada en servicios, la cual permitirá la interoperabilidad entre aplicaciones, se busca un sistema adaptable a dos de las interfaces más comunes de las aplicaciones, las cuales son interfaces móviles y web. Finalmente se proponen varias pruebas entre las que se destacan la carga de servidores, usabilidad de las interfaces y rendimiento del sistema en general.

Alcance

El proyecto propone un sistema de análisis de textos basado en PLN a través del uso de servicios en la nube con la finalidad de determinar delitos de las telecomunicaciones, los cuales

estarán basados en los cuerpos jurídicos del Ecuador y haciendo énfasis, en las nuevas figuras jurídicas establecidas en el Código Orgánico Integral Penal y Ley Orgánica de Telecomunicaciones, debido a que estas normativas jurídicas contemplan varios artículos que hacen referencia a delitos informáticos en general y también de Telecomunicaciones.

Objetivos

General

Diseñar e implementar un sistema de análisis de procesos judiciales, basado en Procesamiento de Lenguaje Natural (PLN) y Servicios en la Nube para determinar delitos en sistemas de telecomunicaciones y su relación con la normativa jurídica del Ecuador.

Específicos

- Analizar las plataformas proveedoras de servicios en la nube con inteligencia artificial y PLN.
- Entrenar modelos para PLN con los datos obtenidos de procesos judiciales relacionados a delitos informáticos y de telecomunicaciones.
- Diseñar una aplicación web y móvil con arquitectura orientada a servicios para la detección de delitos informáticos y de telecomunicaciones.
- Realizar pruebas de rendimiento, funcionamiento y usabilidad de la plataforma.

Estado del Arte

En esta sección se plantean dos bloques bien diferenciados. Por un lado, en la primera subsección se define la metodología usada para la búsqueda del estado del arte y los resultados obtenidos a través de un Mapeo Sistemático de la Literatura (en inglés, Systematic Mapping Study, SMS) y por otro lado una Revisión Sistemática de la Literatura (en inglés, Systematic

Literature Review, SLR). Buscando a través de estas técnicas brindas un mayor conocimiento del estado actual de las tecnologías a tratar en el proyecto.

Las principales diferencias entre un SLR y SMS son su amplitud y profundidad. Mientras que el SLR analiza en profundidad un número reducido de estudios primarios, en un SMS se analiza un número más amplio de estudios, pero menos detallado (Sinoara, Antunes, & Rezende, 2017).

Mapeo Sistemático de la Literatura

En el presente documento se ha explicado la importancia que tiene el PLN para comunicar a los seres humanos con las computadoras. Los avances de la computación en la nube han permitido la implementación de servicios de PLN a través de plataformas como IBM Cloud y Amazon Web Services. El PLN ofrece un amplio campo de aplicaciones como: extracción de información, respuesta a preguntas, traducción automática, síntesis de voz, comprensión del lenguaje, reconocimiento del habla, entre otras.

Por tanto, se decidió realizar un SMS para un mayor conocimiento de que se está haciendo en cuanto a técnicas y herramientas de PLN se refiere, así como sus aplicaciones y líneas de investigación. A lo largo de la presente sección se describe el SMS con la finalidad de proporcionar una base de conocimiento para desarrollar adecuadamente futuras actividades investigadoras.

Objetivos

El presente SMS servirá para conocer en qué líneas de investigación se está trabajando a nivel mundial, así como las técnicas y herramientas más utilizadas en el PLN. Además, va a servir para identificar posibles líneas en las que no se esté trabajando y sean importantes en la

búsqueda del desarrollo de un sistema de PLN para determinar delitos. Los resultados del SMS conforman un lugar de partida para futuras revisiones sistemáticas.

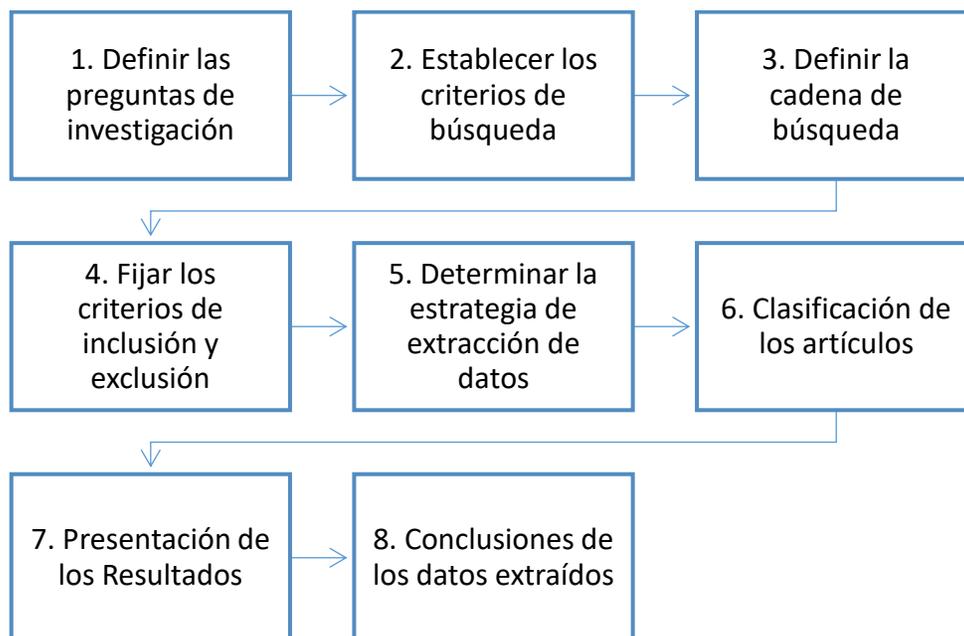
Método de Investigación

El SMS es un método utilizado para categorizar, clasificar y realizar un análisis temático dentro de un tema amplio. Además, un SMS permite obtener un mapeo de las publicaciones sobre algún tema o campo e identificar áreas que requieren el desarrollo de más estudios primarios y áreas en las que un SLR sería de gran ayuda (Petersen, Feldt, Mujtaba, & Mattsson, 2008).

Un SMS es un proceso exhaustivo que tiene como pasos principales: definir las preguntas de investigación, establecer criterios de búsqueda, la selección de estudios, la extracción de información y la presentación de los resultados del estudio. A continuación, en la figura 1 se muestra los pasos seguidos para el presente SMS (Bailey, y otros, 2007).

Figura 1

Procedimiento seguido para el SMS



Definir las preguntas de Investigación

La finalidad del presente SMS es examinar qué técnicas y herramientas están siendo utilizadas en el PLN, así como las líneas de investigación en las que se está trabajando a nivel mundial en este ámbito. Para este fin, se establecen las siguientes preguntas de investigación:

- **RQ1:** ¿Qué técnicas se usan en el procesamiento del lenguaje natural?
- **RQ2:** ¿Qué plataformas en la nube ofrecen procesamiento del lenguaje natural?
- **RQ3:** ¿Qué aplicaciones tiene el procesamiento de lenguaje natural?

Establecer los criterios de búsqueda primarios y secundarios

En el SMS se ha seleccionado como motor de búsqueda la base de datos Scopus. Esta base de datos tiene una amplia cobertura de publicaciones científicas en diversos campos de la ciencia, la ingeniería, la medicina y las ciencias sociales. Además, presenta herramientas para analizar y visualizar los resultados de la búsqueda. Scopus indexa varios catálogos de publicaciones (incluidas IEEE, ACM, Elsevier, etc.).

Al ser un primer reconocimiento de la literatura del PLN no se descarta ningún tipo de publicaciones como tesis doctorales, artículos de divulgación, páginas Web, etc., ya que el objetivo principal es identificar las técnicas existentes del PLN sin importar su origen.

Definir la cadena de búsqueda

Debido a que en un inicio se desconoce el volumen de información con la que se va a trabajar, en primer lugar, se realiza una búsqueda exploratoria del PLN para obtener una primera impresión del número de las publicaciones indexadas al motor de búsqueda Scopus.

Para elaborar la cadena de búsqueda del SMS se han establecido términos alternativos a los términos principales (en negrilla) de cada una de las preguntas de investigación mencionadas con anterioridad:

- **RQ1:**
 - **Natural language Processing** OR NLP OR Procesamiento del Lenguaje Natural OR PLN
 - **Technique** OR Method OR Procedure OR Tool OR Technology OR Algorithm OR Método OR Técnica
- **RQ2:**
 - **Cloud Computing** OR Cloud Platform OR Cloud OR Web Services
- **RQ3:**
 - **Application** OR Use OR Utilization OR Employment OR Lines of Research OR Aplicacion OR Líneas de Investigación

En base a estos términos, la cadena de búsqueda establecida para realizar el SMS fue la siguiente:

("Natural language Processing" OR NLP OR "Procesamiento del Lenguaje Natural" OR PLN) AND (Technique OR Method OR Procedure OR Tool OR Technology OR Algorithm OR Método OR Técnica) AND ("Cloud Computing" OR "Cloud Platform" OR Cloud OR "Web Services") AND (Application OR Use OR Utilization OR Employment OR "Lines of Research" OR Aplicación OR "Líneas de Investigación")

Fijar los criterios de inclusión y exclusión

Los artículos que se han tomado en consideración en la búsqueda son aquellos publicados hasta septiembre de 2020 (momento en que se culmina el SMS). Los documentos han sido seleccionados en base a criterios de inclusión y exclusión que se detallan a continuación.

Los estudios a tomar en cuenta deben cumplir con los siguientes criterios de inclusión:

- Artículos completos.
- Trabajos correspondientes a la rama "Computer Science" o "Engineering".
- Trabajos referentes a cualquier tipo de publicación (revistas, conferencias, tesis doctorales, entre otros).
- Artículos que presentan un método y/o técnica y/o aplicación relacionada al PLN y trabajos que traten sobre plataformas y/o herramientas en la nube con servicios de PLN.
- Publicaciones hasta septiembre de 2020.
- Trabajos que estén en idioma inglés o español.

Por otro lado, trabajos con al menos uno de los siguientes criterios de exclusión han sido rechazados:

- Los trabajos que no sean enfocados al PLN o aquellos en los que se nombre al PLN a modo de ejemplo sin ser el objeto principal del estudio.
- Estudios cuyas ramas no sean “Computer Science” o “Engineering”.
- Presentaciones power-point o similares.
- Estudios cuyo lenguaje no sea el inglés o el español.

Clasificación de los Artículos

La clasificación de los trabajos seleccionados se realizó en tres fases:

1. En primer lugar, se realizó la búsqueda de todos los documentos en base a la cadena de búsqueda detallada en la sección 1.5.1.5. En esta primera fase se obtuvo un total de 356 artículo, teniendo en consideración Title + Abstract + KeyWords.
2. En la segunda fase se filtró los documentos en base a los criterios de inclusión y exclusión mencionados en el Apartado 1.5.1.6. En esta fase se obtuvo 168 artículos, teniendo en cuenta Title + Abstract + KeyWords.
3. Para finalizar se ha aplicó un filtro más selectivo considerando sólo aquellos estudios de interés que mencionen métodos y/o técnicas y/o aplicaciones del PLN, así como aquellos que traten herramientas y/o plataformas en la nube con servicios de PLN. En la fase 3 se obtuvo un total de 19 artículos de interés, tomando en consideración Title + Abstract + KeyWords + Full Article.

Figura 2

Estudios seleccionados en las fases del SMS



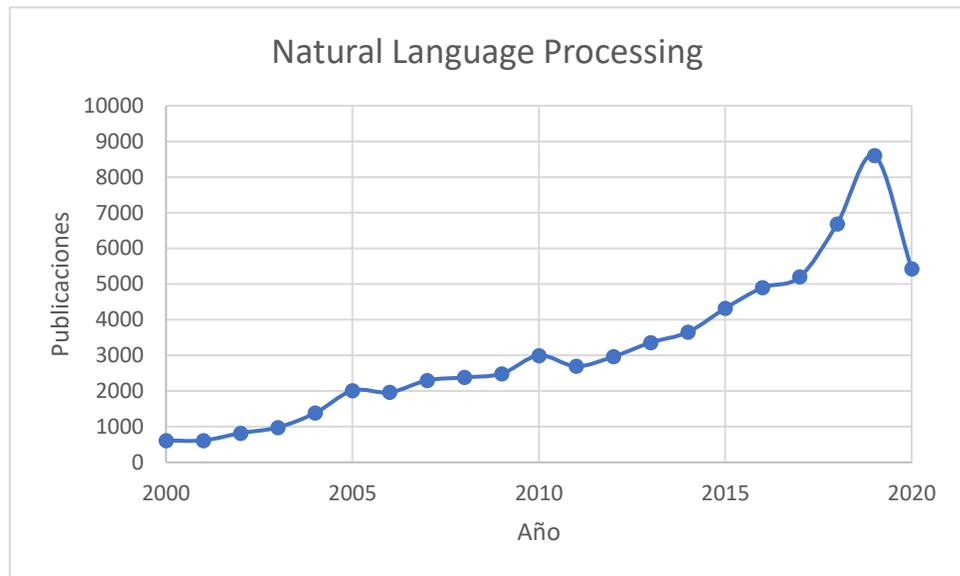
Resultados

Luego de realizar el procedimiento detallado del SMS explicado anteriormente, se ha encontrado 19 estudios de interés.

A continuación, se presenta algunos resultados relevantes obtenidos durante la búsqueda en Scopus. El primer resultado del SMS, es aquel que indica el número de publicaciones relacionadas al PLN durante los años 2000 y 2020, como se muestra en la figura se ha tenido un comportamiento creciente hasta el año 2019, llegando a superar los 8000 artículos.

Figura 3

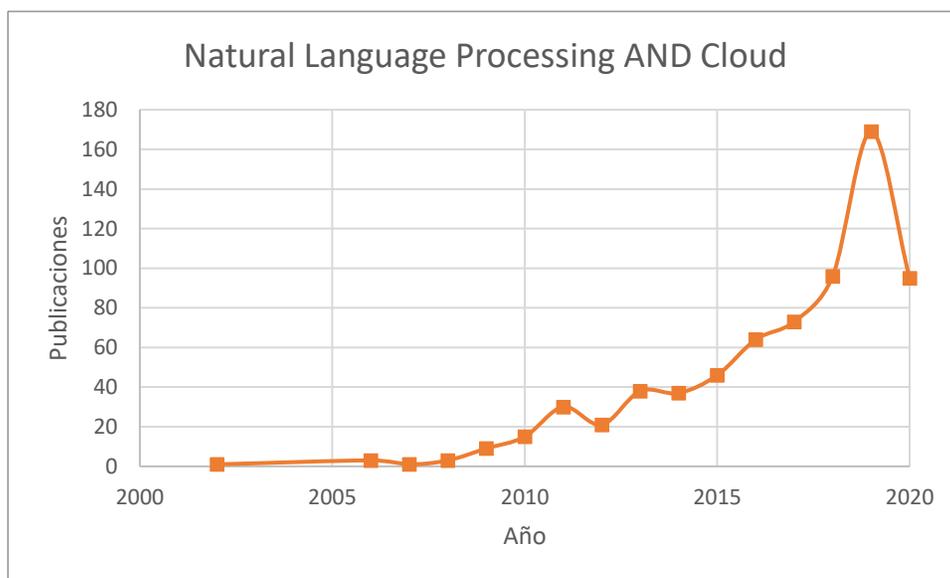
Publicaciones por año del PLN



Otro de los resultados exploratorios obtenidos durante el SMS, es el número de publicaciones relacionados al PLN y Cloud Computing por año, como se muestra en la figura, donde se puede destacar que el número de publicaciones ha ido en aumento desde el año 2010 hasta el año 2019.

Figura 4

Publicaciones del PLN en Cloud



Al aplicar los criterios expuestos previamente a la búsqueda del SMS, se ha obtenido que los tres países con mayor cantidad de publicaciones son Estados Unidos con 27 artículos, India con 23 artículos, seguido por China con 22 publicaciones, como se indica en la figura a continuación.

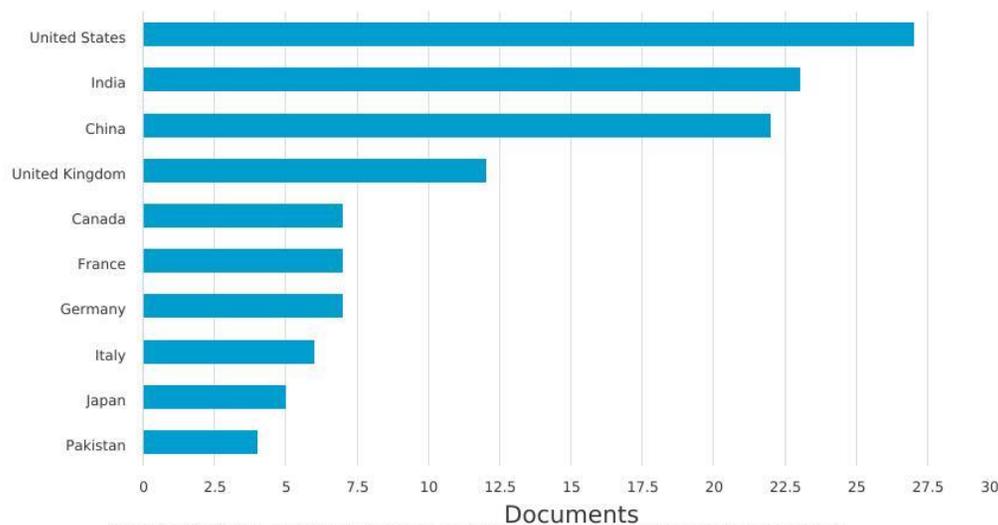
Figura 5

Publicaciones por País Fase 2

Documents by country or territory

Compare the document counts for up to 15 countries/territories.

Scopus



Copyright © 2020 Elsevier B.V. All rights reserved. Scopus® is a registered trademark of Elsevier B.V.

Nota. La figura representa los documentos por país relacionados con NLP y Cloud, (Scopus, 2020).

Conclusiones del SMS

En el presente punto se responde a las preguntas de investigación formuladas. Las cuales han sido contestadas en parte gracias a los resultados de la búsqueda y a los artículos seleccionados durante la fase 3.

RQ1: ¿Qué técnicas se usan en el procesamiento del lenguaje natural?

Según (Cambria & White, 2014), (IBM Cloud Education, 2020) y (Priyadarshini, Bagjadab, & Mishra, 2020) existen tres enfoques del PLN. El más antiguo que data desde 1960, denominado Simbólico, está basado en un conjunto de reglas escritas a mano. A fines de la década de 1980, hubo una revolución en el PLN con la introducción de algoritmos de Machine Learning y Deep Learning, ingresando al enfoque Estadístico, que se utiliza para extraer,

clasificar y etiquetar automáticamente elementos de texto y datos de voz y luego asignar una probabilidad estadística a cada posible significado de esos elementos.

El tercer enfoque está basado en Deep Learning y Redes Neuronales, las técnicas basadas en este enfoque permiten que los sistemas de PLN aprendan a la vez que trabajan, logrando así extraer un significado cada vez más preciso de grandes volúmenes de datos de voz y texto (IBM Cloud Education, 2020). En (Young, Hazarika, Poria, & Cambria, 2018), (Goldberg, 2016), (Chai & Li, 2019) y (Guo, Huang, & Pan, 2020) se describe los principales métodos de este enfoque, entre los cuales se destacan las redes neuronales convolucionales (CNNs), las redes neuronales recurrentes (RNNs) y sus variaciones (long short term memory “LSTM” y gated recurrent units “GRUs”) y las redes neuronales recursivas.

RQ2: ¿Qué plataformas en la nube ofrecen procesamiento del lenguaje natural?

Empresas como Google, Microsoft, Amazon e IBM ofrecen en sus plataformas en la nube al PLN como un servicio basado en la nube. En (Ray & Mathew, 2019) se presentan una comparación de los cuatro principales asistentes virtuales con PLN disponibles en la actualidad: Google DialogFlow, Microsoft LUIS, IBM Watson Conversation y Amazon Lex. Además de las plataformas antes mencionadas, en (Patil, Marimuthu, Nagaraja Rao, & Niranchana, 2017) se presentan dos plataformas que ofrecen PLN como un servicio, estas son Heroku y Kore.

En (Tablan, Bontcheva, Roberts, Cunningham, & Dimitrov, 2013) se presenta AnnoMarket, una plataforma abierta en la nube que permite implementar, compartir y usar componentes y recursos de PLN. AnnoMarket se centra en los recursos y servicios de análisis de texto multilingües. Por otro lado, en (Padro & Turmo, 2015) se describe TextServer, una

plataforma basada en la nube para el PLN que ofrece una variedad de servicios robustos para una amplia gama de idiomas.

RQ3: ¿Qué aplicaciones tiene el procesamiento de lenguaje natural?

Los campos de aplicación del PLN son muy extensos, en (Guo, Huang, & Pan, 2020), (Yogish, Manjunath, & Hegadi, 2019) y (Dudhabaware & Madankar, 2014) se describen algunos de ellos, como el análisis léxico, traducción automática, análisis emocional, análisis del discurso, clasificación de textos, resumen de textos, reconocimiento de entidades (NER), sistemas de respuesta a preguntas automáticos.

Entre los documentos analizados se encuentran aplicaciones del PLN enfocadas a la Ingeniería en Software. En (Yalla & Sharma, 2015) se describe la aplicación del PLN en diversas fases del Ciclo de Vida del Desarrollo de Software (SDLC). Por otro lado, en (Nazir, Butt, Anwar, & Khan Khattak, 2017) se hace una revisión de la aplicación del PLN en la Ingeniería de requisitos de software, donde se menciona que el PLN se utiliza para transformar los requisitos de software funcional en artefactos de diseño y también se utiliza para refinar las ambigüedades de los requisitos textuales iniciales.

Otro gran campo de aplicación del PLN son los asistentes virtuales y chatbots, en (El Kamali, y otros, 2020) se desarrolló un asistente virtual diseñado para ayudar a las personas mayores, que interactúa con el usuario a través de mediante un chatbot con una aplicación móvil y un objeto físico basado en la interacción vocal y la interacción tangible. Por otro lado, en (Zahour, Benlahmar, Eddaoui, Ouchra, & Hourrane, 2020), (Gupta, y otros, 2019) y (Mekni, Baani, & Sulieman, 2020) presentan investigaciones en el uso de chatbots como tutores inteligentes para apoyar el aprendizaje de los estudiantes al responder sus preguntas.

Otras Conclusiones

Además de las conclusiones en base a las preguntas de investigación del SMS, se presentan conclusiones de interés para el desarrollo del proyecto de investigación y para el SLR.

- Se confirma que los investigadores tienen gran interés en lo que se respecta a PLN y existen diferentes herramientas de cloud computing para este fin.
- El desarrollo de los últimos años de la computación en la nube ha permitido que muchas empresas de TI desarrollen plataformas de PLN natural en la nube (Padro & Turmo, 2015; Patil, Marimuthu, Nagaraja Rao, & Niranchana, 2017; Ray & Mathew, 2019). Estas plataformas en la nube se basan fundamentalmente en dos conceptos para su funcionamiento: intenciones (del inglés, intents) y entidades (del inglés, entities); el uso de estos parámetros es que pueden predefinirse y reutilizarse (Ray & Mathew, 2019).
- A pesar de excluir cualquier documento no relacionado a las ciencias de la computación y a la ingeniería, cabe mencionar que el PLN ya es utilizado en nuestra vida diaria, y con los avances en el PLN será cada vez más utilizado en campos financieros, legales, medicinales y de la salud. Por ejemplo, en el campo legal el PLN puede ayudar con búsquedas de casos, predicciones de juicios, generación automática de documentos legales y traducción de textos legales (Guo, Huang, & Pan, 2020).

Revisión Sistemática de la Literatura

Después de realizar el SMS se ha podido valorar cuáles son las líneas en las que se está trabajando en la actualidad en el sector del PLN, así como también el uso que se le ha dado a las diferentes plataformas de servicios en la nube. De los resultados obtenidos cabe destacar que no se han encontrado trabajos en los que se haga uso tanto del PLN, como de los servicios en la

nube para la generación de un sistema web de análisis de textos enfocado en el área legal, se cree conveniente investigar con mayor detalle el concepto y aplicabilidad de servicios en la nube orientado hacia esta área, teniendo en cuenta sobre todos los servicios de PLN.

A diferencia del SMS, que intenta cubrir un amplio campo sin adentrarse en detalles, el SLR cubre con mayor profundidad un tema concreto (Farshchian & Dahl, 2015).

El SLR es una investigación científica, su propósito es Integrar los resultados de la investigación de manera objetiva y sistemática. Permite realizar investigaciones empíricas sobre cuestiones de investigación específicas para lograr identificar su "estado del arte" en el campo de investigación. Para lograr este objetivo, SLR requiere el desarrollo de una serie de etapas similares a las previamente desarrolladas en SMS.

A continuación, se presenta el SLR realizado con el objetivo de presentar un conocimiento más completo sobre el tema de interés de este trabajo, un sistema de análisis de textos que use PLN, que esté enfocada al área legal y a la determinación de cualquier delito.

Objetivos

El objetivo principal es comprender qué tecnología, técnica, método o proyectos existen en la actualidad en torno a lo que se refiere a PLN para análisis de textos en el área legal, para obtener un bagaje que pueda servir para continuar con la línea de investigación.

El SLR permitirá conocer si los trabajos realizados se pueden aplicar en la actualidad a entornos reales y conocer los resultados obtenidos en el caso de que existan, además observar si estos resultados son satisfactorios para los usuarios.

Debido a la amplitud de preguntas necesarias para conocer las investigaciones de este tema, se eligió SLR para conocer las investigaciones y sistemas que han sido publicados recientemente en el ámbito legal y porque la investigación a través de SMS no es suficiente.

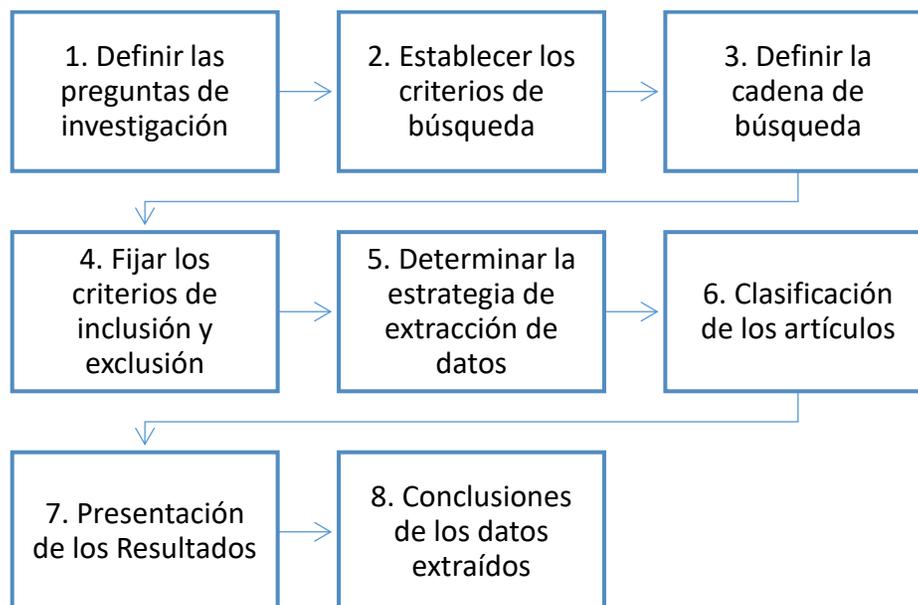
Método de Investigación

El método de investigación mediante SLR consta de tres actividades: planificación, ejecución y elaboración de informes (Kitchenham & Charters, 2007). Cada proceso se divide en varios pasos. La planificación incluye el desarrollo y la puesta en marcha de un plan de trabajo para ejecutar con éxito el SLR. Las actividades de ejecución incluyen extracción de datos, su selección e integración, convirtiéndola así en información útil. indicados en el mensaje en. Finalmente, la elaboración y explicación del evento resultado.

A continuación, se muestra el proceso similar a SMS, que seguirá los siguientes pasos:

Figura 6

Proceso seguido para el SLR



Definir las preguntas de Investigación

La finalidad del presente SLR es examinar qué técnicas y herramientas están siendo utilizadas en el PLN y el área legal. Para este fin, se sugieren las siguientes preguntas:

- **RQ1:** ¿Qué estudios usan el procesamiento del lenguaje natural enfocado en el área legal?
- **RQ2:** ¿Qué aplicaciones o herramientas realizan procesamiento del lenguaje natural enfocado en el área legal mediante el uso de servicios de cloud computing?
- **RQ3:** ¿A qué procesos del área legal está ayudando el uso del procesamiento de lenguaje natural?

Establecer los criterios de búsqueda primarios y secundarios

Los criterios serán similares a los realizados en la sección SMS, sin embargo y a diferencia del SMS, se busca mayor profundidad tecnológica, así como también aplicabilidad

real, por lo que se centra en artículos científicos y conferencias que brinden información y tengan una perspectiva legal o que sea relativo a leyes y/o delitos que involucren a las TICs.

Se utilizarán para esta búsqueda la base de datos científica Springerlink, así como también la base de datos Scopus y además la biblioteca digital IEEE Xplore, de modo que se puedan obtener una mayor cantidad de información que relacionen a los temas ya expuestos en la sección SMS.

Definir la cadena de búsqueda

Para cada una de las preguntas previamente señaladas en la sección Definir las preguntas de Investigación se determinan el término principal y los términos alternativos, basados en estos se realiza la búsqueda, gracias a la ayuda de los operadores.

- **RQ1**
 - Procesamiento de lenguaje natural OR natural language processing OR NPL OR PLN
 - Legal OR law OR security OR cybersecurity
- **RQ2**
 - Procesamiento de lenguaje natural OR natural language processing OR NPL OR PLN
 - Legal OR law OR security OR cybersecurity
 - Cloud Computing OR servicios en la nube OR nube OR cloud
- **RQ3**
 - Process OR procesos OR features OR tools
 - Legal OR law OR security OR cybersecurity
 - Procesamiento de lenguaje natural OR natural language processing OR NPL OR PLN

A continuación, se presenta la cadena de búsqueda para la biblioteca digital IEEE Xplore:

("All Metadata":NLP OR "All Metadata":natural language processing OR "All Metadata":PLN)
AND ("All Metadata":cloud OR "All Metadata":cloud computing OR "All Metadata":nube) AND
("All Metadata":law OR "All Metadata":legal OR "All Metadata":security)

La cadena de búsqueda para la base de datos Scopus es:

TITLE-ABS-KEY ((nlp OR "natural language processing" OR pln) AND ("cloud computing" OR
cloud OR nube) AND (law OR legal OR leyes OR cybersecurity))

Así también, la cadena de búsqueda para la base de datos Springerlink es:

(cloud AND law OR legal OR security)

Teniendo en cuenta que se combinó esta búsqueda en todo tipo de documentos y asegurando que el término NLP (natural language processing) se encuentre en el título de los mismos.

Fijar los criterios de inclusión y exclusión

En el desarrollo de la búsqueda de información, se ha tenido en cuenta toda la información recolectada hasta septiembre del 2020 (momento en el cual, se termina la revisión).

Cada documento determinado ha sido evaluado y estudiado para determinar su inclusión en el presente proyecto. Cabe mencionar que ciertas normas deben cumplirse las cuales son:

- Artículos completos.
- Trabajos pertenecientes a las ramas "Computer Science" y "Engineering"
- Trabajos publicados en revistas o conferencias.

- Estudios relacionados al PLN, cloud computing y relación con área legal.
- Publicaciones desde enero del 2010 hasta septiembre del 2020.

También, los documentos que cumplen con al menos una de las siguientes características, fueron excluidos:

- Los documentos que no traten ningún tema sobre el área legal o que sean relacionados a cloud computing.
- Los documentos introductorios, libros y presentaciones debido a su falta de detalles.
- Artículos duplicados o publicaciones realizadas en diferentes años por el mismo autor y que abarquen el mismo tema.
- Artículos no escritos en inglés.

Clasificación de los Artículos

De la misma forma que en el SMS, la extracción de datos se realizó en tres fases:

- Primero se realizó la búsqueda de todos los documentos con las cadenas de búsqueda definidas previamente. Tomando en cuenta Title + Abstract + KeyWords, se obtuvieron 96 documentos entre artículos de revista y artículos de conferencias. De los cuales 21 corresponden a Scopus, 56 a Springerlink y 19 a IEEE Xplore.
- La segunda búsqueda se realizó tomando en cuenta los criterios de inclusión y exclusión. Tomando en cuenta Title + Abstract + KeyWords, se obtuvieron 72 artículos. De los cuales 13 corresponden a Scopus, 46 a Springerlink y 13 a IEEE Xplore.
- Finalmente se aplicó un nuevo filtro donde, una vez revisados todos los documentos se consideró los que verdaderamente aportan para el tema del proyecto. Tomando en

cuenta Title + Abstract + KeyWords, se obtuvieron 38 artículos. De los cuales 7 corresponden a Scopus, 6 a Springerlink y 5 a IEEE Xplore.

Figura 7

Artículos consultados en cada una de las fases del SLR



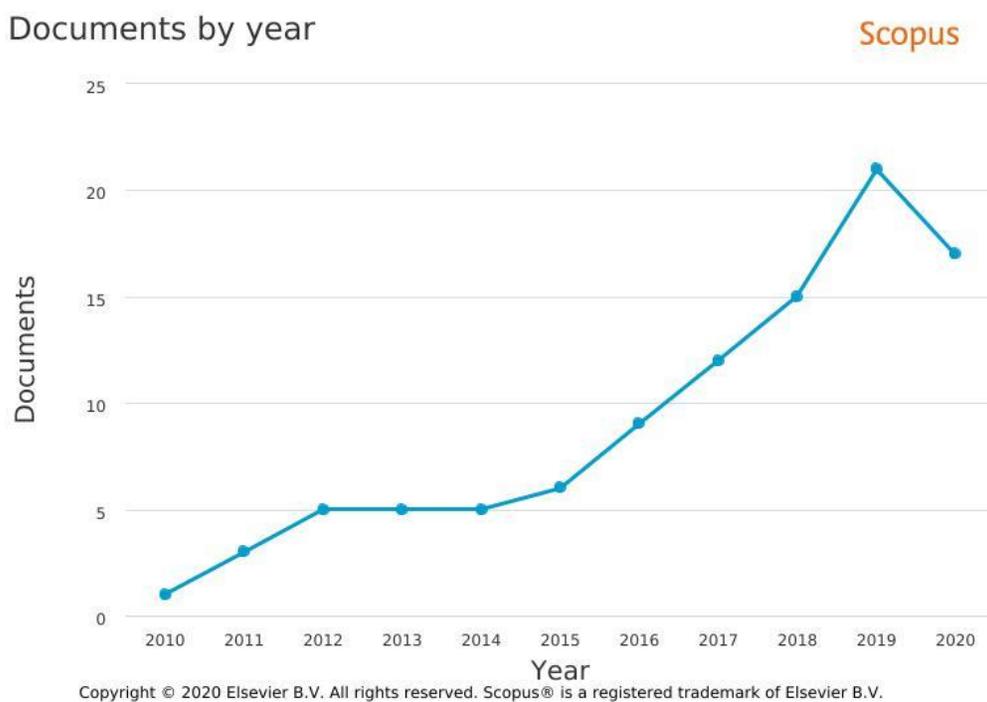
Resultados

Finalmente se han obtenido 18 trabajos de interés, los cuales cumplen con los requisitos del procedimiento previamente descrito. Se muestran a continuación algunos de los datos más significativos de la investigación planteada para el SLR.

Es necesario destacar que del estudio SLR y del mismo modo que fue descrito en la sección anterior SMS, el área del NLP (natural language processing) está creciendo, año tras año se puede visualizar el crecimiento del número de artículos y de documentos en general que abarcan esta temática, pero aún más importante para la temática específica, se demuestra gracias a la ayuda del análisis de los resultados de la búsqueda de Scopus, que tanto el NLP está siendo combinado con temas relacionados con cloud y que se enfocan en el área como una de los principales área de investigación.

Figura 8

Número de artículos que relacionan NLP, cloud y área legal por año

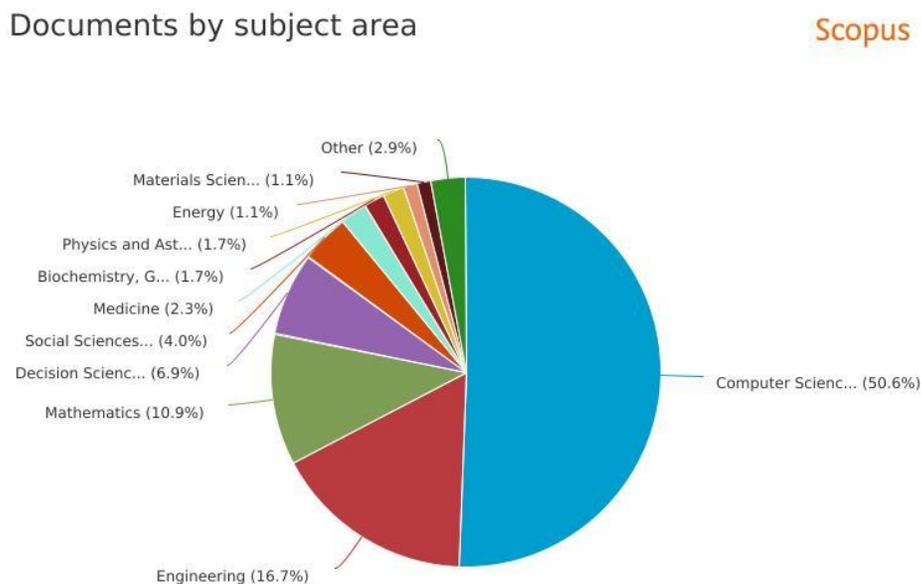


Nota. La figura representa los documentos por año relacionados con NLP, Cloud y legal, (Scopus, 2020).

Adicionalmente, es necesario tener en cuenta las áreas de aplicabilidad de este tipo de tecnologías y de esta forma y gracias al análisis de resultados de la búsqueda provisto por Scopus y teniendo en cuenta que nuevamente se ha combinado el NLP con temas relacionados con cloud y el área legal, se puede visualizar que el área de Computer Science con 50.6 % e Engineering 16.7 % poseen la mayor cantidad de artículos.

Figura 9

Porcentaje de documentos por área que relacionan NLP, cloud y el área legal



Copyright © 2020 Elsevier B.V. All rights reserved. Scopus® is a registered trademark of Elsevier B.V.

Nota. La figura representa el porcentaje de documentos por cada una de las áreas de Scopus, (Scopus, 2020).

Conclusiones del SLR

A continuación, se responden las preguntas de investigación formuladas previamente. Las cuales han sido contestadas gracias al correcto proceso de SLR, con cada uno de los documentos y artículos seleccionados durante la fase 3.

RQ1: ¿Qué estudios usan el procesamiento del lenguaje natural enfocado en el área legal?

Los principales trabajos sobre lo que se refiere a PLN y área legal han sido enfocados en varias áreas relativas sobre todo a la era digital en su mayoría entre las cuales se puede encontrar la ciberseguridad (Murcia, Moreno, Díaz, & Gómez, 2019), (Jayathilaka, Weerasinghe, & Wijesekara, 2016), (Ho, Oliveira, & Rathi, 2019) y (Sulea O, Dinu L, & A, 2015), de la misma forma y siguiendo esta línea, se encontró información de detección de ataques de phishing a través de las URLs (Buber, Diri, & Sahingoz, 2017) y detección de malware (Mathews, 2019) y (Rath, Agarwal, & Shyamasundar, 2017).

Debido a las regulaciones impuestas por muchos gobiernos, por ejemplo, la GDPR (General Data Protection Regulation) que se tiene en la Unión Europea, la protección de datos (Yang, Huang, Luo, & Yu, 2020), (Nayak & Pasumarthi, 2019) y (Keserwani & Samaddar, 2017) se ha convertido en un tema de vital importancia en estos días. La disponibilidad de datos (Yang, Huang, Luo, & Yu, 2020) y las políticas de privacidad (Mittal, Gupta, Joshi, Pearce, & Joshi, 2017) se pueden alcanzar a través del uso del NLP.

A través del NLP, es posible detectar y analizar indicadores que diversas organizaciones buscan a través de las redes sociales, para conformar redes de trata de personas (Brewster, Ingle, & Rankin, 2014).

RQ2: ¿Qué aplicaciones o herramientas realizan procesamiento del lenguaje natural enfocado en el área legal mediante el uso de servicios de cloud computing?

Existen varias herramientas, aplicaciones y sistemas que se han creado para el uso de NLP con el manejo de servicios de cloud computing y que se encuentran en mayor o menor medida enfocados en el área legal, realizando análisis de la sintaxis de diversos textos, entre los cuales es posible encontrar:

- Question and Answering system for cloud computing services: Una tarea realizada por el sistema es determinar el tipo de pregunta de entrada. Basándose en 4 tipos típicos de preguntas: "Qué", "Dónde", "Cómo" y "Cuándo". Cada una de estas preguntas tiene su propio propósito, Qué se ocupa de las definiciones, dónde se ocupa de la ubicación, cuándo se ocupa de los aspectos temporales y cómo se refiere a la descripción. La presencia de estas palabras permite identificar el tipo de pregunta que se tiene para gestionar acuerdos de nivel de servicio en la nube, que luego se utiliza para consultar a la base de conocimientos (Mittal, Gupta, Joshi, Pearce, & Joshi, 2017).
- En (Lesmo, Mazzei, Palmirani, & Radicioni, 2013) se propone un sistema para producir anotaciones automáticas de una normativa a través de la extracción y modificación de provisiones legales.
- En la actualidad existen aplicaciones web que utilizan inteligencia artificial para manejar gran parte del trabajo pesado en el área legal. Entre ellas se destacan ROSS Intelligence y LegalMation. ROSS Intelligence es impulsada por la supercomputadora Watson de IBM, que atiende consultas legales al acelerar la legislación y otros recursos. ROSS Intelligence es un software privado de investigación legal que utiliza inteligencia artificial para ayudar a miles de abogados estadounidenses a trabajar de manera más rápida e inteligente. Ross encuentra la ley más relevante en segundos utilizando inteligencia artificial de última generación (Simson, 2019).

RQ3: ¿A qué procesos del área legal está ayudando el uso del procesamiento de lenguaje natural?

Actualmente la comunidad científica se encuentra dando uso al NLP para realizar análisis de textos de manera profunda de manera que las máquinas puedan entender los

sentimientos, entidades y la lingüística en general de los seres humanos al transmitir una idea, así también se ha permitido tener una aplicabilidad legal en la cual se destacan los siguientes procesos:

- Crímenes sexuales y en especial la pornografía. C³-Sex (Murcia, Moreno, Díaz, & Gómez, 2019).
- Crimen organizado para la trata de personas (Brewster, Ingle, & Rankin, 2014).
- Regulaciones de proveedores de servicio, seguridad en internet aplicado en India para evitar la publicación de datos personales de las empresas y personas que contraten servicios (Jayathilaka, Weerasinghe, & Wijesekara, 2016).
- La privacidad digital y la protección de datos e ha convertido en un tema de suma importancia debido a la conexión global que tienen los seres humanos en la actualidad (Keserwani & Samaddar, 2017; Nayak & Pasumarthi, 2019; Rath, Agarwal, & Shyamasundar, 2017; Sulea O, Dinu L, & A, 2015; Yang, Huang, Luo, & Yu, 2020) .
- Finalmente, los procesos legales han tenido una evolución permitiendo ahora tratar acuerdos de nivel de servicio (SLA) y políticas de privacidad que se dan en los documentos legales relacionados a la contratación de servicios en la nube (Mittal, Gupta, Joshi, Pearce, & Joshi, 2017) y (De Marco, Ferrucci, Kechadi, Napoli, & Salza, 2016), así también el análisis de los distintos certificados de seguridad (Labaj, Rástočný, & Chudá, 2019) y las garantías colaterales que se puedan generar en diversos sectores, ya que estos tienen un impacto legal y económico sobre todo en la industria de la construcción.

Otras Conclusiones

Adicionalmente de las respuestas a las preguntas formuladas, también se tienen una serie de conclusiones que son de interés para el desarrollo del proyecto de investigación y además para futuros trabajos que se desprendan de este.

- Se confirma al igual que en el caso del SMS, que la comunidad científica se encuentra interesada y centra sus atenciones en lo que se refiere a NLP y se busca utilizar las diferentes herramientas de cloud computing para este propósito.
- El área legal es una de las más prometedoras en cuanto a NLP y a machine learning se refiere.
- Además, del uso del área legal, otras áreas prometedoras para la combinación de NLP y cloud computing son los sistemas IoT (Luo, Cheng, Hu, Peng, & Yao, 2020), el análisis de textos clínicos (Carrell, 2011; Mathews, 2019).

Metodología

La metodología a seguir para el desarrollo de este proyecto, es inductiva ya que se analizarán casos particulares de procesos judiciales relacionados con delitos informáticos y de las telecomunicaciones, los cuales serán extraídos desde la página web de la función judicial del Ecuador, buscando automatizar y digitalizar las normas legales que se encuentran dentro de la gaceta oficial del estado, de manera que se pueda realizar una generalización de los mismos a través de un modelo de machine learning encontrando y determinando normas que se ajusten a las conductas tipificadas ingresadas por los usuarios.

Organización del contenido

El Capítulo 2 explica el marco teórico sobre el cual se fundamenta el proyecto, describiendo cada uno de los conceptos a utilizar en el resto de los capítulos. Se introducen las definiciones de PLN, y se menciona un conjunto de herramientas de PLN populares en el desarrollo de aplicaciones. Además, se exponen los conceptos de inteligencia artificial, machine learning y cloud computing. Luego, se desarrollan los conceptos de servicios Web y de arquitectura orientada a servicios. Para finalizar, se describe algunas tecnologías de Front-End y Back-End.

El Capítulo 3 presenta el diseño del sistema propuesto, explicando cada uno de los módulos implementados en el sistema. Se detalla la sección del back-end donde se realiza una orquestación de servicios como el PLN, la interpretación de voz a texto y la síntesis de voz. Luego, se describe la sección del front-end, que dispone de una interfaz web y una interfaz móvil que invocan los servicios Web creados.

En el Capítulo 4 se realiza un análisis de las pruebas de rendimiento, funcionamiento y usabilidad de la plataforma. Las pruebas de rendimiento darán como resultado el número de solicitudes que soporta el sistema sin prolongar el tiempo de espera del usuario. Las pruebas de funcionamiento valorarán la calidad y precisión de la información otorgada por el sistema. En las pruebas de usabilidad, los usuarios evaluarán a la aplicación teniendo en cuenta la facilidad de uso, nivel de satisfacción. agrado el manejo de la aplicación.

En el Capítulo 5 se presentan las conclusiones y recomendaciones del trabajo de investigación, explicando las ventajas, limitaciones y posibles mejoras del sistema. Además, en

este capítulo se discuten diferentes líneas de trabajo futuro y aplicaciones que se desprenden de este trabajo.

Capítulo II

Marco Teórico

Normas Jurídicas

Código Orgánico Integral Penal

El Código Orgánico Integral Penal más conocido por sus siglas como COIP, es un documento de carácter legal, su propósito es homogeneizar el poder punitivo del Estado Ecuatoriano, tipificar las infracciones penales, establecer procedimientos judiciales que respeten estrictamente el debido proceso, promover la rehabilitación social de las personas condenadas y brindar una indemnización integral a las víctimas. (Ministerio de Justicia, Derechos Humanos y Cultos, 2014)

Este documento contiene una recopilación legislativa, la cual establece delitos y duración de las penas para cada uno de los mismos conforme a lo señalado por el organismo penal ecuatoriano. Consta de 730 artículos entre los cuales se encuentran los relacionados a los delitos de las telecomunicaciones de los cuales se profundizará más adelante en esta sección.

Ley Orgánica de Telecomunicaciones

La ley Orgánica de Telecomunicaciones es un documento de carácter legal, el cual Tiene como objetivo convertir el sistema general de telecomunicaciones y espectro radioeléctrico en un departamento estratégico del país de acuerdo con los principios y derechos establecidos por

la Constitución, incluyendo competencias administrativas, reglamentarias, de control y gestión dentro del territorio nacional (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2015). A través de este documento la Agencia de Regulación y Control de las telecomunicaciones, mejor conocida como ARCOTEL administra, controla y regulará todos los temas relacionados al uso y explotación del espectro radioeléctrico en el territorio ecuatoriano, ya que este constituye un bien del Estado Ecuatoriano.

Este documento contiene diferentes pautas, derechos y obligaciones que deben ser cumplidos por todas las partes que usan y explotan el espectro radioeléctrico, constituyen una herramienta diseñada para mejorar los servicios que se brindan actualmente, busca proteger y defender a suscriptores, clientes y usuarios, teniendo en cuenta las obligaciones por parte de los prestadores de servicios de telecomunicaciones, buscando así impulsar el desarrollo de la sociedad brindando acceso a las TICs en todo el territorio nacional, a través de 148 artículos.

Delitos de Telecomunicaciones

Hoy en día el sector de las telecomunicaciones ha experimentado un crecimiento y desarrollo tecnológico alrededor del mundo. Las nuevas tecnologías contempladas dentro de las TICs, contribuyen al desarrollo económico, social y mejoran la calidad de vida de las sociedades, gracias a los canales ininterrumpidos de comunicación es posible lograr una comunicación ágil y eficiente entre organizaciones, países empresas y gobiernos. Diversos sectores económicos se han visto beneficiados del crecimiento de las telecomunicaciones, como es el caso del sector de salud brindando telemedicina y apoyo a todos los ámbitos relacionados con ella; el sector comercial se ha visto impulsado en el servicio de internet, logrando así la comunicación con diversos clientes alrededor del mundo, de manera que se pueden comercializar y promocionar

productos. Si bien este desarrollo ha permitido que la sociedad obtenga enormes ventajas también, este tipo de tecnologías se han convertido en un medio donde se pueden llegar a dar diferentes fraudes, estos no solamente afectan a usuarios, sino también a los operadores y proveedores de servicio, las pérdidas económicas son abundantes (Superintendencia de Telecomunicaciones, 2011).

Un delito informático se refiere a un acto antijurídico perpetrado a través del uso de un espacio digital, pudiendo este referirse a software o a hardware, por lo tanto, un delito de las telecomunicaciones representa una acción criminal que pueda estar relacionada a actos entre los cuales se tienen principalmente el hurto, sabotaje, fraude, falsificación o estafa. Según (Llangarí Salazar , 2016) los propósitos de estos delitos se basan en los mismos que se dan en delitos informáticos (analizar posibles amenazas y vulnerabilidades dentro de diferentes sistemas informáticos o telemáticos, como también tener control de acceso a la información), pero también se hace referencia en especial a fraudes ya que afecta a todos los operadores de telecomunicaciones y prestadores de servicios en sus ingresos. Varias fuentes estiman que las diferentes empresas de telecomunicaciones pierden aproximadamente el 10% de sus ingresos debido a la falta de herramientas y procedimientos técnicos para controlar el fraude, de manera que no se aseguran los ingresos de las mismas.

Los principales delitos de telecomunicaciones son: Fraude por suscripción, clonación de teléfonos celulares, call back también conocido como llamada revertida, fraude tercer país, fraude en Roaming, robo de líneas telefónicas, by pass.

Fraude por Suscripción: se refiere a proporcionar información y documentos fraudulentos e inexactos, logrando acceso a uno de los servicios previstos por parte de una operadora, de manera que se eviten las obligaciones de pago hacia los mismos. En varios de estos casos se suplanta la identidad de un individuo para realizar este fraude. Según (Superintendencia de Telecomunicaciones, 2011) estas acciones representan una doble afectación ya que "... afecta directamente a la persona suplantada, que figurará como morosa en el buró de crédito. Además, las empresas de telecomunicaciones se verán afectadas por la pérdida de ingresos debido a la imposibilidad de cobrar el consumo realizado."

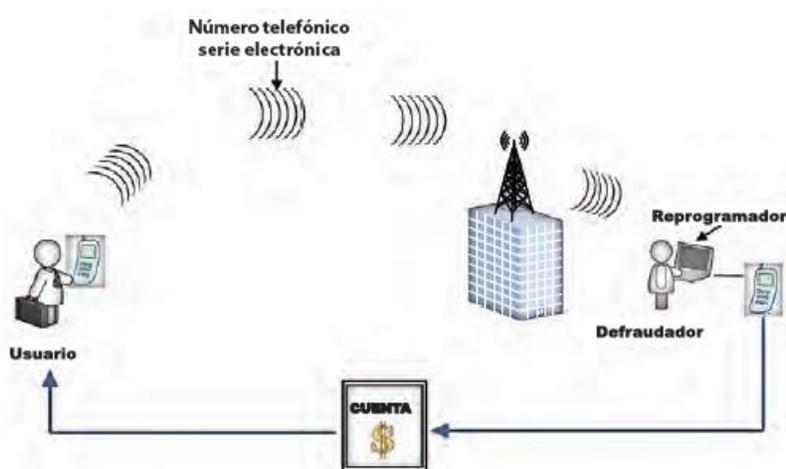
Clonación de teléfonos celulares: Este es el fenómeno de copiar números de teléfonos móviles y números de serie desde el aire para programarlos en otros dispositivos. Esto es llevado a cabo mediante un monitoreo o mediante un laboratorio, luego de llevado el dispositivo para una reparación o revisión. Los dispositivos clonados generarán facturas a suscriptores reales del servicio ya que generan tráfico telefónico. De esta forma, los dispositivos fraudulentos pueden utilizar el servicio sin pagar una tarifa. (Superintendencia de Telecomunicaciones, 2011)

El número de serie del dispositivo celular juega un papel fundamental en este tipo de delitos, pues de esta forma se genera un cargo por las llamadas realizadas desde el dispositivo clonado, lo que ocasiona que el suscriptor pierda temporalmente el servicio, esto también es evidenciado en casos de hurto y extravío de dispositivos. Existen varios factores que pueden indicar que un teléfono ha sido clonado entre los cuales destacan: caídas frecuentes de conexión, dificultades para completar llamadas y acceder al buzón de mensajes, llamadas

recibidas provenientes de números desconocidos tanto locales como internacionales y un alto valor en la factura por el servicio. (Superintendencia de Telecomunicaciones, 2011)

Figura 10

Fraude por clonación de celulares



Nota. Esquema general de un fraude por clonación de celulares, (Superintendencia de Telecomunicaciones, 2011)

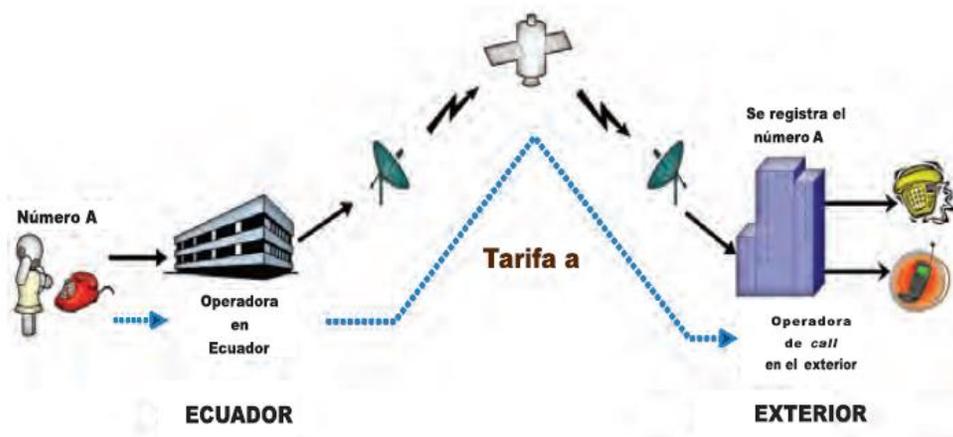
Call back: la llamada revertida o call back es el procedimiento a través del cual el origen del tráfico internacional es revertido, haciendo un denominado "disparo" hacia un número predeterminado en el exterior. No se contesta esta llamada internacional y por lo tanto no es cobrada. Mediante el uso de un equipamiento al otro lado de la línea, se identifica y guarda el número de origen de la llamada. Posteriormente a la terminación de la llamada, se reintegra la comunicación con tono del país en el exterior, gestionando de manera local, es decir fingiendo

una llamada internacional como si fuera local. Esto es realizado a través de tres pasos, a continuación, se los describe. (Superintendencia de Telecomunicaciones, 2011)

Paso 1: Una llamada al exterior es realizada desde el origen en este caso Ecuador, a través de un operador móvil legal; al no ser contestada no es facturada al usuario. Sin embargo, existe una tarifa facturada debido a la interconexión del operador local hacia el exterior, la cual finaliza la llamada hacia la empresa de call back. En el exterior, la llamada es registrada con el número procedente del origen en este caso desde Ecuador. (Superintendencia de Telecomunicaciones, 2011)

Figura 11

Establecimiento del call back paso 1

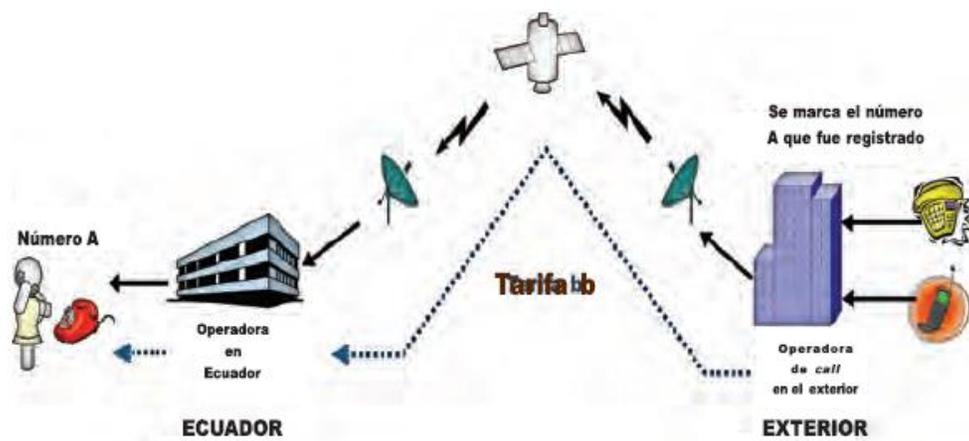


Nota. Procedimiento general de establecimiento del primer paso de call back, (Superintendencia de Telecomunicaciones, 2011).

Paso 2: La empresa encargada de revertir la llamada, conecta al número registrado por sus equipos, en otras palabras, el número A, el cual recibe tono de marcado. En este caso, se marca una tarifa b, la cual es facturada debido a la interconexión internacional, teniendo como origen el exterior y fin al Ecuador, siendo finalmente esta tarifa mucho menor a la tarifa de interconexión desde el origen Ecuador hacia el exterior, es decir la tarifa a que debió ser realmente facturada.

Figura 12

Establecimiento del call back paso 2



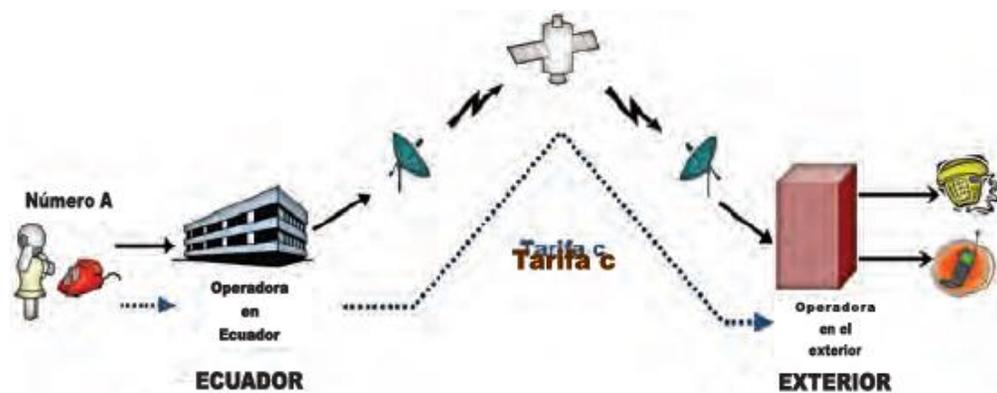
Nota. Procedimiento general de establecimiento del segundo paso de call back, (Superintendencia de Telecomunicaciones, 2011).

Paso 3: Se define una nueva tarifa c, la cual es facturada al realizar la llamada simulando que la misma estuviera en el exterior, esta tarifa, se determina por la empresa que realiza la

llamada revertida, y es diferente de acuerdo a la demanda de cada país. Es cancelada a través de tarjetas prepago o de páginas web, donde es normal que se oferten este tipo de llamadas, a través de la implementación del equipamiento adecuado en la central telefónica de tránsito internacional, es posible combatir esta clase de delito de manera óptima y eficaz, evitando así este tipo de fraudes. (Superintendencia de Telecomunicaciones, 2011)

Figura 13

Establecimiento del call back paso 3



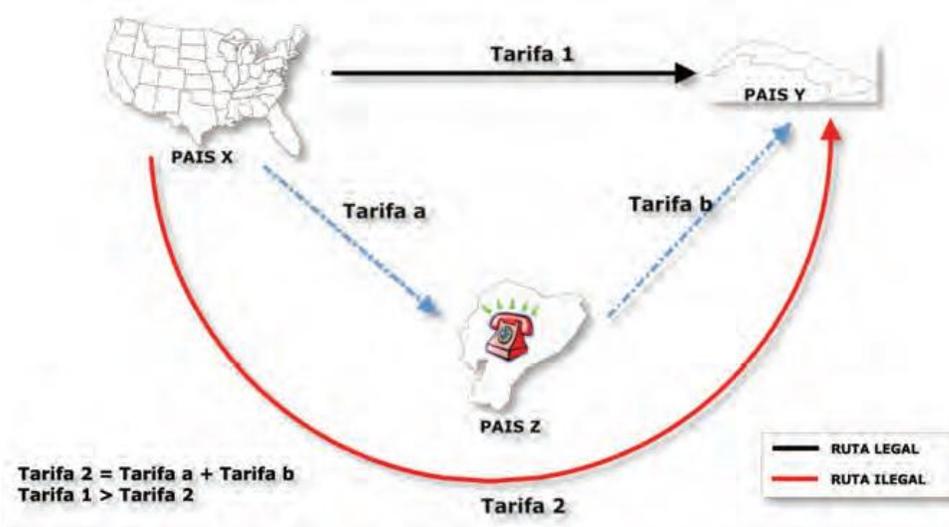
Nota. Procedimiento general de establecimiento del tercer paso de call back, (Superintendencia de Telecomunicaciones, 2011).

Fraude tercer país: este tipo de fraude consiste en que un país denominado X, origine un tráfico telefónico, el mismo que será enrutado hacia otro país, el cual no es el destino final de la llamada, este país Z realiza nuevamente un enrutamiento hasta su último destino, el país Y. Esto es realizado ya que existe discrepancia en cuanto a tarifas se refiere para estos países, buscando que el país X pague un valor inferior, realizando la redirección de la llamada hacia el país Z, previamente a lograr la comunicación con el destino final. Es justamente la diferencia de

precios existentes entre países, los que originan este delito de telecomunicaciones, así el país que ha originado el tráfico tendrá una tarifa inferior que el país de destino. (Superintendencia de Telecomunicaciones, 2011)

Figura 14

Fraude tercer país



Nota. Procedimiento general de Fraude tercer país, (Superintendencia de Telecomunicaciones, 2011).

Fraude en Roaming: Anualmente son generadas cantidades enormes de pérdidas económicas, las cuales son reportadas por operadoras de telefonía móvil, sobretodo existe una alta tasa de contribución a estas pérdidas, debido a los diferentes escenarios de roaming. (Maciá Fernández, 2008)

El roaming es un servicio mediante el cual un abonado de una determinada red móvil es capaz de recibir y realizar llamadas de voz o datos fuera de la zona geográfica de cobertura de su red principal, a través del uso de recursos de una red visitada o alojada. El fraude en roaming se refiere a la intención de no pagar por estas llamadas y/o datos transmitidos.

Para el año 2008, el fraude GSM sufría del fraude en roaming, el mismo que ascendía al 50%. Los operadores no solo perderán los ingresos no cobrados de la cuenta utilizada para acceder al servicio, sino que también sufrirán pérdidas debido a los pagos que deben realizarse a los socios de roaming. (Meza Ayala, 2008)

Los escenarios de roaming son simplemente una extensión a los escenarios de fraude convencionales, sin embargo, es necesario señalar que este tipo de fraude posee ciertas características que hacen más perjudiciales desde el punto de vista de las pérdidas que se producen para el proveedor de servicios, según (Maciá Fernández, 2008) son:

- Mayor tiempo de detección debido a que se producen en una red diferente a la del abonado.
- Mayor tiempo de reacción debido a que existen dificultades técnicas y administrativas para intentar detener estas actividades debido a la falta de control sobre los sistemas de redes visitadas.

Robo de líneas telefónicas: Este tipo de fraude se realiza en contexto interno o externo, y es realizado cuando líneas activas son alteradas y modificadas de domicilio sin autorización del suscriptor o de la empresa proveedora del servicio. Este tipo de infracción resulta común ya que

debido a la facilidad de acceso a la red externa. Este delito es perpetrado por personal de mantenimiento, instaladores de planta externa o por exfuncionarios los cuales poseen conocimiento acerca de la distribución de la red. Las líneas telefónicas son utilizadas por terceras personas y el consumo es cobrado al suscriptor del servicio. (Meza Ayala, 2008)

Este tipo de fraude no afecta solamente al proveedor de servicio debido a los daños y cambios en la infraestructura, además, supone un daño a la imagen y reputación de la compañía, por otra parte, el abonado deberá comprobar este fraude caso contrario deberá cancelar al proveedor el monto total facturado lo cual puede suponer cuantiosas pérdidas para el mismo.

By Pass: Este tipo de delito busca registrar y facturar una llamada telefónica internacional, como una llamada nacional, es decir se busca que dicha llamada sea conectada por una red telefónica establecida de forma legal en el país. Desde el punto de vista tecnológico y práctico, esto es posible debido a que dicho sistema se implementa con tres elementos básicos e indispensables: el primero un enlace internacional para recibir el tráfico originado en el exterior; una instalación de equipos de telecomunicaciones, que efectúa el procesamiento de voz en cada llamada internacional recibida; y varias líneas telefónicas utilizadas para terminar cada llamada internacional en una red de una operadora nacional. Todos estos elementos, son interconectados y juntos configuran una ruta telefónica internacional, la cual actúa de manera clandestina y paralelamente a las rutas legalmente establecidas por las operadoras autorizadas en el país. Finalmente, el Servicio Telefónico de Larga Distancia Internacional (STLDI) es prestado

sin contar con el correspondiente título habilitante entregado por la Agencia de Regulación de las Telecomunicaciones. (Superintendencia de Telecomunicaciones, 2011)

Figura 15

Elementos de un sistema By pass



Nota. Elementos y procedimiento general de un sistema By pass, (Superintendencia de Telecomunicaciones, 2011).

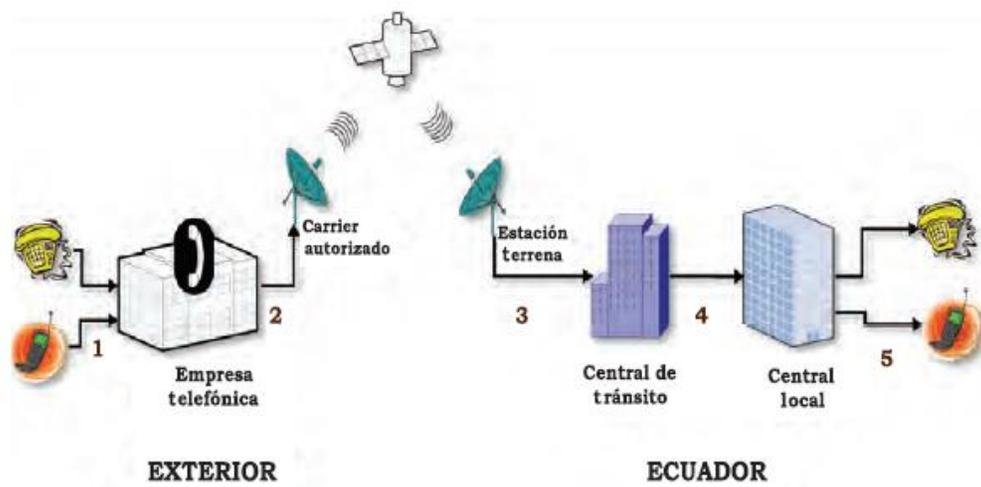
Para comprender completamente el proceso de implementación de este sistema, a continuación, se propone una ruta internacional autorizada y luego se compara con la ruta internacional ilegal denominada "by pass".

Ruta internacional autorizada

- En el extranjero, pueden realizar llamadas hacia el Ecuador a través de teléfonos tradicionales o teléfonos móviles de compañías telefónicas extranjeras autorizadas. Las llamadas pueden ser realizadas por operadores que brindan servicios telefónicos internacionales, o mediante tarjetas prepagas que legalmente brinden este servicio.
- El proveedor extranjero enruta la llamada hacia el Ecuador, la cual llega a la estación terrena de una operadora legalmente establecida y autorizada, y finalmente, es enrutada a través de la central de tránsito.
- En la central de tránsito, debido al código de acceso digitado, se encuentra reconocida legalmente y se realiza el proceso de facturación como llamada internacional, de acuerdo a las tasas internacionales de telefonía fijadas por la agencia de regulación y la operadora local.
- Posteriormente al proceso de facturación la llamada llega a la central local, donde se enruta hacia el destino final. Esta llamada solamente puede alcanzar la central local, una vez establecido un acuerdo con la operadora en el exterior, de manera que se conozca la tarifa de interconexión por ambas partes, es importante mencionar que a través de este acuerdo se establece la legalidad de la llamada internacional.
- Por último, la llamada llega al número de destino de una operadora establecida en el país, el cual puede ser convencional o móvil.

Figura 16

Ruta legal de una llamada Internacional



Nota. Procedimiento general de una llamada internacional con una ruta legal, (Superintendencia de Telecomunicaciones, 2011).

Ruta internacional ilegal denominada By pass

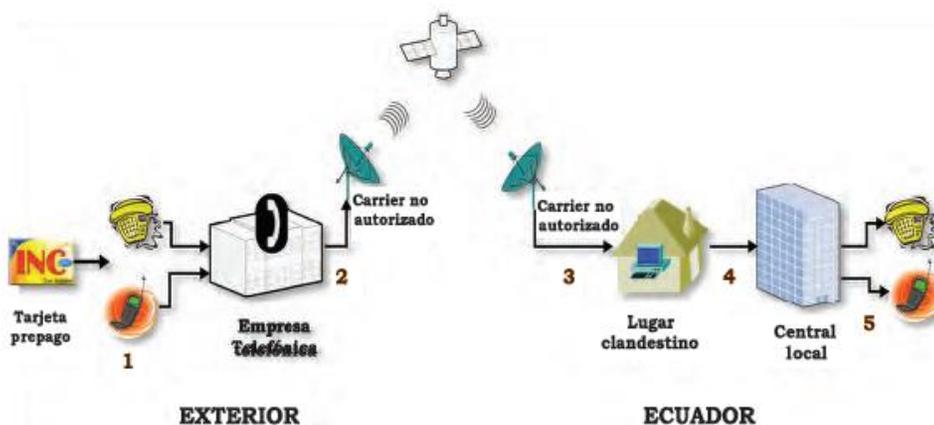
- Una llamada puede realizarse desde el exterior con destino Ecuador desde un teléfono fijo o celular. Esta llamada puede realizarse mediante el uso de tarjetas prepago no autorizadas, generalmente comercializadas en locales informales, o a través de Internet. Ya que su precio es bajo y se ofrecen grandes beneficios, las personas las adquieren sin conocimiento de la ruta ilegal usada para cursar la llamada.
- Al marcar el número de destino en el exterior, la llamada la dirige una empresa telefónica hacia un carrier 1, el cual está encargado de enrutar la llamada hacia Ecuador. Carriers en el exterior son contactados y contratados, estableciendo acuerdos bilaterales con beneficios comerciales para uso del telepuerto privado para realizar este tipo de llamadas, este acuerdo no se encuentra autorizado. Sin embargo, es necesario

mencionar que los carriers internacionales desconocen que las empresas que los contactan carecen de la correspondiente autorización en Ecuador.

- Desde este telepuerto, las llamadas son enviadas hacia el lugar clandestino en el cual operan los defraudadores, en donde se hace el enrutamiento para lograr que la llamada ingrese al Ecuador.
- Los defraudadores se encargan de cursar la llamada hacia las redes móviles nacionales una vez que han ingresado al país.
- Es posible usar líneas fijas o celulares para que la llamada internacional pueda cursar simulando ser una llamada local. Así, cada central de las operadoras telefónica autorizadas en el Ecuador, detectan estas llamadas, pero con una firma local. Logrando así que una llamada internacional sea facturada como local, consumándose el delito.

Figura 17

Ruta By pass ilegal



Nota. Procedimiento general de una llamada internacional con una ruta ilegal, (Superintendencia de Telecomunicaciones, 2011).

El fraude realizado mediante el uso de este mecanismo de by pass supone el ocultamiento de las llamadas internacionales y el objetivo es lograr que estas, sean tratadas como llamadas locales. Esta afectación como se ha mencionado previamente no solo repercute en el proveedor de servicios, también se realiza una afectación al estado el cual es el dueño del espectro radioeléctrico.

De acuerdo a las normativas jurídicas ecuatorianas señaladas previamente en esta sección, a continuación, se realiza un compendio de los artículos de interés tanto del Código Orgánico Integral Penal como de la Ley Orgánica de Telecomunicaciones que hagan referencia o se encuentren estrechamente relacionados con los delitos de las telecomunicaciones.

Respecto al Código Orgánico Integral Penal, los artículos que expresamente señalan delitos relacionados con las telecomunicaciones son:

- Artículo 188. Aprovechamiento ilícito de servicios públicos: este artículo se refiere al aprovechamiento indebido de servicios público de telecomunicaciones en beneficio propio o de terceros, ofertando y/o prestando servicios sin contar con la debida licencia o concesión.
- Artículo 190. Apropiación fraudulenta por medios electrónicos: este artículo se refiere a la utilización engañosa de sistemas de telecomunicaciones con el fin de realizar la apropiación de un bien ajeno.
- Artículo 191. Reprogramación o modificación de información de equipos terminales móviles: este artículo se refiere a la alteración de la información de identificación de los terminales móviles usados en sistemas de telecomunicaciones.
- Artículo 192. Intercambio comercialización o compra de información de equipos terminales móviles: este artículo se refiere a la venta y/o compra de bases de datos que contengan registros con información de identificación de terminales móviles.
- Artículo 193. Reemplazo de identificación de terminales móviles: este artículo se refiere a la sustitución de la identificación de terminales móviles con información diferente a la original de los mismos.
- Artículo 194. Comercialización ilícita de terminales móviles: este artículo se refiere a la venta de terminales móviles los cuales han sufrido una alteración o han violado las disposiciones de la autoridad competente ARCOTEL.
- Artículo 195. Infraestructura ilícita: este artículo se refiere a la posesión de estructuras, programas o equipos que alteren la información de identificación de un terminal móvil.
- Artículo 229. Revelación ilegal de bases de datos: este artículo se refiere a la anunciación de información registrada en una base de datos la cual forme parte de un

sistema de telecomunicaciones. Materializando la violación a la intimidad y privacidad de cada uno de los individuos.

- Artículo 230. Interceptación ilegal de datos: este artículo se refiere a la interceptación de datos informáticos en cualquiera de los puntos de un sistema informático, así como también la clonación de dispositivos electrónicos con el objetivo de obtener un provecho ya sea este propio o de terceros.
- Artículo 232. Ataque a la integridad de sistemas informáticos: este artículo se refiere al daño, alteración o destrucción de un sistema de telecomunicaciones o de una de sus partes.
- Artículo 234. Acceso no consentido a un sistema informático, telemático o de telecomunicaciones: este artículo se refiere a la entrada sin autorización de un sistema de telecomunicaciones de manera que se explote ilegítimamente este acceso.

También es necesario señalar los artículos que se encuentran en mayor o menor medida relacionados con delitos de las telecomunicaciones, los cuales en su mayoría son delitos informáticos:

- Artículo 178. Violación a la intimidad: este artículo se refiere a la revelación sin consentimiento de datos personales, mensajes, voz, audio o video.
- Artículo 179. Revelación de secreto: este artículo se refiere a la divulgación de un secreto el cual ha tenido conocimiento debido a oficio, profesión y empleo.
- Artículo 186. Estafa: se refiere a la simulación de hechos ficticios o a la desfiguración de hechos verídicos para que se realice un acto que perjudique a su patrimonio.

- Artículo 212. Suplantación de identidad: este artículo se refiere al reemplazo de identidad con el fin de obtener un beneficio para sí o para un tercero, perjudicando así a un ciudadano.
- Artículo 231. Transferencia electrónica de activo patrimonial: este artículo se refiere a la intención de manipular el funcionamiento de un programa o sistema para mediante el mismo transferirse un activo patrimonial de una persona.
- Artículo 233. Delitos contra la información pública reservada legalmente: este artículo se refiere a la destrucción, alteración o inutilización de información confidencial.

Por otra parte, de acuerdo a la Ley Orgánica de Telecomunicaciones los artículos que hacen referencia a los delitos relacionados con las telecomunicaciones son:

- Artículo 117. Infracciones de primera clase: este artículo se refiere a las infracciones de primera clase tanto para personas naturales o jurídicas y poseedores de títulos habilitantes, por ejemplo: la comercialización o utilización de equipos terminales que no hayan sido homologados; la instalación de infraestructura sin contar con la respectiva señalética y dispositivos de seguridad, respectivamente.
- Artículo 118. Infracciones de segunda clase: este artículo se refiere a las infracciones de segunda clase tanto para personas naturales o jurídicas y poseedores de títulos habilitantes, por ejemplo: obstaculizar el ejercicio de control, auditoria y vigilancia por parte de la ARTCOTEL; interrumpir de forma total o parcial el servicio sin autorización, respectivamente.

- Artículos 119. Infracciones de tercera clase: este artículo se refiere a las infracciones de tercera clase tanto para personas naturales o jurídicas y poseedores de títulos habilitantes, por ejemplo: explotación de frecuencias sin haber obtenido el título habilitante o concesión; cobrar tarifas por encima de los topes aprobados por la ARCOTEL, respectivamente.
- Artículos 120. Infracciones de cuarta clase: este artículo se refiere a las infracciones de cuarta clase para poseedores de títulos habilitantes, por ejemplo: ceder, gravar o transferir el título habilitante para la prestación de servicios.

Legal Tech

El término Legal Tech hace referencia de forma general a la aplicación de las nuevas tecnologías de la información y de software en el ámbito jurídico para facilitar, complementar y optimizar los servicios y actividades legales de abogados, jueces, procuradores y otros profesionales del sector legal (Corrales, Fenwick, & Haapio, 2019).

Según Bues y Matthaei (2017) se destacan tres categorías en Legal Tech. La primera categoría consta de herramientas de almacenamiento en la nube para facilitar el acceso a abogados y datos legales. La segunda categoría abarca herramientas de procesos de apoyo para adoptar sistemas de gestión de casos y back-office más efectivos. Estos procesos pueden variar desde la gestión de recursos humanos hasta la contabilidad, facturación, nómina y otras tareas administrativas. La tercera categoría adopta soluciones que ayudan o incluso reemplazan el asesoramiento legal de los abogados. Esta categoría incluye varias subcategorías, como la generación de contratos de forma automática, resolución de disputas en línea, análisis legal y tecnologías basadas en Blockchain.

El uso de la inteligencia artificial y machine learning en el campo legal ha aumentado en gran escala desde el 2014. Los bufetes de abogados acogen cada vez más la idea de aprovechar la información que se encuentra digitalizada y emplear las nuevas tecnologías para mejorar sus servicios y procesos. Esto se ve reflejado en las inversiones realizadas durante el 2018 en el mercado de Legal Tech, un monto estimado de \$ 3,245 millones (Chishti , 2020).

Aplicaciones de la Tecnología en el Campo Legal

En una época donde se genera día a día una gran cantidad de información de forma digital que resulta imposible de procesar por una persona. La aplicación de las tecnologías que se encuentran en auge como la inteligencia artificial y los servicios de computación en la nube ofrecen una solución para analizar, almacenar y procesar toda esta información.

Específicamente en el campo legal se plantean diversas aplicaciones y usos como:

- Almacenamiento y organización de bases de datos con una gran cantidad de casos jurídicos e información legal.
- El análisis y clasificación de documentos legales y normas jurídicas mediante técnicas del procesamiento de lenguaje natural.
- Seguimiento de las actividades realizadas por abogados u otros actores jurídicos.
- Sistemas jurídicos que analicen e interpreten una gran cantidad de datos para elaborar un veredicto que ayude en la resolución de un caso (IAT, 2020).

Beneficios de la tecnología aplicada al Campo Legal

La aplicación de nuevas tecnologías como la inteligencia artificial, el procesamiento de lenguaje natural, la computación en la nube y los servicios web en el campo legal trae consigo diversos beneficios como:

- Facilitar el estudio y la comprensión de leyes, términos y normas jurídicas.
- Disminuir el tiempo dedicado a diligencias, análisis de información y recopilación de pruebas para solventar casos con mayor agilidad.
- Organizar y recopilar la información jurídica digital disponible de forma eficaz y automática.
- Liberar a los profesionales del campo jurídico de actividades rutinarias o de oficina y puedan enfocarse en tareas específicas de valor agregado (IAT, 2020).

Ejemplos de Legal Tech en el Mundo

Los primeros sistemas y prototipos relacionados a Legal Tech surgen alrededor de los años 90. En Italia Di Giorgi, Fameli y Nannucci (1992) crearon un sistema denominado ELP (Environmental Legal Protection Advisor). Este sistema tiene la finalidad de proporcionar información y referencias a las autoridades legales en el ámbito de la protección legal del medio ambiente. ELP utiliza la búsqueda mediante palabras claves en documentos y registros integrando bases de datos de Oracle.

Ashley (2000) elaboró un sistema llamado THE CATO Program en la Universidad de Pittsburg. Este sistema tenía como finalidad la resolución de casos exponiendo los hechos, precedentes y posibles argumentos y contraargumentos aplicando inteligencia artificial.

En la actualidad entre los sistemas y plataformas relacionadas a Legal Tech se destacan Luminance, ROSS Intelligence y LegalMation. Luminance es un sistema de gestión legal alojado en la nube creado por matemáticos de la Universidad de Cambridge para ayudar a las empresas a gestionar y optimizar los procesos de revisión de documentos utilizando inteligencia artificial. Entre sus principales características se destacan la detección de anomalías, visualización de datos, agrupación de documentos y colaboración en varios idiomas (Loxam, 2018).

ROSS Intelligence es un software privado de investigación legal que utiliza la inteligencia artificial para ayudar a miles de abogados estadounidenses a trabajar de manera más rápida e inteligente. Ross encuentra la ley más relevante en segundos utilizando inteligencia artificial impulsada por la supercomputadora Watson de IBM (Simson, 2019).

LegalMation proporciona un conjunto de herramientas de Inteligencia Artificial para ayudar a los abogados a automatizar las tareas de rutina. LegalMation reunió a un equipo de expertos en la materia para utilizar IBM Watson Knowledge Studio e IBM Watson Natural Language Understanding que se ejecutan en la infraestructura de IBM Cloud para crear un modelo específico de dominio centrado en la terminología y los conceptos legales (Lau, 2020).

Legal Tech en el Ecuador

Los primeros movimientos en el Ecuador relacionados a Legal Tech surgen en el 2018, mediante el evento “Legal Innovation Ecuador” organizado por un grupo de abogados, contando con especialistas como Leonardo Hernández (Gerente General de Lexis Finder), Martín Burbano (Fundador de Inside) y Diego Álvarez (Country Manager de Biz Latin Hub). A pesar de que el

evento no tuvo posteriores ediciones, serviría de motivación para que otros profesionales apliquen las nuevas tecnologías en el campo legal (Legaltechies, 2020).

En 2019 mediante el mapeo de emprendimientos tecnológicos, Radar Tech Startup 4.0, en el Ecuador realizado por Buen Trip Hub se evidenció por primera vez en su cuarta edición el segmento correspondiente a Legal Tech, con empresas como Inside y Legal Issues (BuenTrip, 2019). Además, durante el 2019 la Universidad Espíritu Santo y el Instituto de Innovación Legal organizaron el concurso de ideas para mejorar la Justicia mediante una aplicación, #JUSTIAPPS2019, destacando las apps: Resoluto, Delictum y Ayni (Legaltechies, 2020).

Actualmente en el Ecuador existen un total de 25 empresas vinculadas al concepto de Legal Tech. Acorde al estudio realizado por Legaltechies (2020), se distinguen las siguientes categorías:

- Plataformas o Software de gestión: facilitan al abogado llevar el control de clientes, facturación, casos, etc.
- Plataformas para documentos y contratos online: son sistemas de preguntas y respuestas que tienen la finalidad de recopilar información y generar un documento o contrato.
- Plataformas de intermediación (Marketplaces jurídicos): para ofertar los servicios de un abogado y los clientes encuentren al profesional deseado. Destaca la empresa Ulpik, que cuenta con un chatbot que en base a la IA resuelve consultas legales o brinda asesoría en trámites jurídicos.
- Servicios de automatización de procesos legales: a través de nuevas tecnologías brindan la posibilidad de hacer tareas legales de forma más eficiente.

Figura 18

LegalTech en el Ecuador: Principales nichos



Nota. Principales nichos de Legaltech en Ecuador, (Legaltechies, 2020).

Cloud Computing

Definición

Varios autores y organismos de estandarización han planteado diversas definiciones de la computación en la nube, proporcionando diferentes puntos de vista en base a elementos y características fundamentales de la computación en la nube.

El Instituto Nacional de Estándares y Tecnología (NIST del inglés, National Institute of Standards and Technology) define a la computación en la nube como un modelo que facilita el acceso bajo demanda a un grupo compartido de recursos computacionales configurables, como redes, servicios, servidores, almacenamiento y aplicaciones (Mell & Grance, 2011).

Vaquero, Rodero, Cáceres y Lindner (2009) proponen una definición que engloba varias características clave de la computación en la nube y adicionalmente presentan una visión más amplia y práctica de la misma. Describen que la computación en la nube ofrece un gran conjunto de recursos virtualizados (como infraestructura, plataformas de desarrollo y/o servicios) de fácil uso y acceso. Estos recursos pueden ser reconfigurados de forma dinámica de acuerdo a su demanda, optimizando así su uso. Usualmente la computación en la nube brinda estos recursos mediante un modelo de pago por uso, basados en acuerdos de nivel de servicios (SLA del inglés, service level agreement).

Características

Autores como Mell y Grance del NIST (2011) señalan que existen cinco características esenciales de la computación en la nube. Sin embargo, otros autores como Murugesan y Bojanova (2016), Gour (2019) y Jones (2020) destacan otras características que también son importantes al momento de hablar de la nube. A continuación, se enlistan las características de la computación en la nube:

- Autoservicio bajo demanda: permite al usuario agregar o eliminar recursos informáticos siempre que sea necesario a través de un panel de control en línea.

- **Amplio acceso a la red:** facilita el acceso al servicio desde múltiples ubicaciones, como casa u oficina, a través de diferentes dispositivos, como PCs, celulares, laptops.
- **Agrupación de recursos y múltiples inquilinos:** el proveedor de la nube agrupa los recursos informáticos para dar servicio a varios usuarios, lo cual se puede lograr mediante una máquina virtual. Los recursos se asignan en tiempo de ejecución, ayudando a aumentar la utilización y reducir costos de operación.
- **Rápida elasticidad:** los recursos asignados al usuario se pueden aumentar o disminuir rápidamente según su necesidad de forma manual o automática, desde el punto de vista del usuario las capacidades computacionales disponibles son casi ilimitadas.
- **Servicio medido:** los proveedores de la nube miden los servicios que brindan (ancho de banda, memoria, almacenamiento, etc.). El usuario solo paga por lo que consume y puede monitorear los recursos utilizados.
- **Menores costos de infraestructura:** con la computación en la nube se reduce los costos de comprar, instalar, configurar y administrar su propia infraestructura local.
- **Autorreparable:** los proveedores en la nube brindan procesos de respaldo de información, creando respaldos o copias de seguridad automáticas que pueden ser fácilmente restablecidas.
- **Responsabilidad compartida:** los proveedores en la nube se encargan de la seguridad de la infraestructura. La responsabilidad del usuario depende del servicio contratado en la nube, por ejemplo, en IaaS el usuario se encargará de que el sistema operativo utilizado este actualizado y sobre las aplicaciones que se ejecutan en el mismo.

- Agilidad y bajo tiempo de lanzamiento al mercado: el desarrollo e implementación de aplicaciones en la nube es más rápida, debido a que se empieza a trabajar antes y la inversión inicial es menor.

Modelos de Servicio en la nube

Chandrasekaran y Ananth (2016) describen que los servicios en la nube son recursos computacionales ofrecidos al usuario en forma de plataforma, software o infraestructura. Estos servicios en la nube se pueden clasificar en tres modelos principales: Software como Servicio (SaaS), Plataforma como Servicio (PaaS) e Infraestructura como Servicio (IaaS). Adicionalmente a estos modelos Vennam (2020) agrega el modelo de servicio Serverless computing.

Estos modelos de servicio en la nube se diferencian en el nivel de control que tiene el proveedor de la computación en la nube y el usuario.

Infraestructura como Servicio (IaaS)

IaaS ofrece al usuario una infraestructura con redes, almacenamiento, servidores físicos y virtuales en la nube para implementar y ejecutar software, como sistemas operativos y aplicaciones. Este modelo permite a los usuarios reducir gastos en infraestructura con un pago por uso (Mell & Grance, 2011; Murugesan & Bojanova, 2016).

En este modelo de servicio el usuario tiene control sobre los sistemas operativos, datos y aplicaciones implementadas; mientras que el proveedor de servicios en la nube se encarga de controlar los servidores físicos y virtuales, la red y el almacenamiento (Mell & Grance, 2011;

Vennam, 2020). Algunos ejemplos de servicios de IaaS son Amazon Elastic Compute Cloud (EC2), Azure IaaS, Google Compute Engine, IBM Cloud IaaS, GoGrid y FlexiScale.

Plataforma como Servicio (PaaS)

En este modelo se proporciona al usuario una plataforma en la nube que incluye infraestructura, sistema operativo, middleware y hasta herramientas para el desarrollo de aplicaciones. PaaS permite al usuario desarrollar, probar, implementar, administrar y entregar aplicaciones y servicios mucho más rápido y a menos costo que con una plataforma local (Murugesan & Bojanova, 2016; Vennam, 2020).

En PaaS, el usuario tiene control de las aplicaciones implementadas y de los datos; mientras que el proveedor en la nube administra la infraestructura de nube subyacente, es decir, la red, los servidores físicos y virtuales, el almacenamiento y los sistemas operativos (Mell & Grance, 2011). Algunos ejemplos de PaaS incluyen Google App Engine, Microsoft Azure, Amazon Web Services, IBM Cloud Foundry, OpenShift, Cloud Foundry y Salesforce.

Software como Servicio (SaaS)

En este modelo de servicio se ofrece al usuario aplicaciones que se ejecutan en una infraestructura en la nube. El usuario puede acceder a las aplicaciones desde varios dispositivos mediante una conexión a internet o una red dedicada, eliminando la necesidad de instalar y ejecutar la aplicación en el dispositivo terminal del usuario (Murugesan & Bojanova, 2016).

En SaaS, el consumidor no tiene control sobre la infraestructura de nube subyacente (red, servidores físicos y virtuales, almacenamiento, sistemas operativos, etc...) a excepción de los ajustes de configuración de la aplicación específico (Mell & Grance, 2011). Algunos ejemplos de aplicaciones bajo el modelo de SaaS son Spotify, Dropbox, Netflix, las aplicaciones Google Apps como Gmail, Drive, Classroom, Meet.

Serverless Computing

En este modelo las tareas de administración de back-end (aprovisionamiento, escalado, programación) son ejecutadas por el proveedor de la nube, permitiendo a los usuarios enfocarse en desarrollar el código y la lógica empresarial específicos para sus aplicaciones. El código de la aplicación se ejecuta por solicitud y se escala la infraestructura en base al número de solicitudes. En este modelo, los usuarios pagan únicamente por los recursos empleados cuando la aplicación está en ejecución (Vennam, 2020).

Función como Servicio (FaaS) es un subconjunto del modelo Serverless Computing. FaaS se enfoca en el paradigma de la computación impulsada por eventos en el que una función o contenedor se ejecutan solo en respuesta a determinados eventos o solicitudes (Vennam, 2020).

Figura 19

Diferencias entre los modelos de servicio en la nube

Traditional IT	IaaS	PaaS	Serverless	SaaS
Applications	Applications	Applications	Applications	Applications
Data	Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking	Networking

■ You manage ■ Provider manages

Nota. Se presentan los conceptos de cada modelo de servicio en la nube, (Vennam, 2020).

Modelos de Implementación

Además de la clasificación en base al tipo de servicio, la computación en la nube se puede clasificar en base a cómo fue implementada y a quienes la utilizan y administran en cuatro categorías: nube pública, nube privada, nube comunitaria y nube híbrida.

Nube Privada

La nube privada es utilizada por una empresa u organización para su uso exclusivo y/o el de sus socios comerciales. Esta nube puede ser una propiedad implementada y administrada por la propia organización, un tercero o de forma mixta, y puede estar localizada dentro o fuera de las instalaciones de la organización (Mell & Grance, 2011).

Nube Pública

La nube pública es la nube más utilizada y conocida, y la puede usar cualquier empresa, industria, gobierno, organizaciones o individuo. La infraestructura de la nube es propiedad y es administrada por el proveedor de servicios en la nube. La nube pública se aloja en las instalaciones del proveedor. Una nube pública se ofrece en un modelo de pago por uso; sin embargo, algunas aplicaciones en nubes públicas son accesibles gratuitamente (Murugesan & Bojanova, 2016).

Nube Comunitaria

Una nube comunitaria es utilizada por un grupo de usuarios que comparten algo en común. Su infraestructura puede ser propiedad implementada y administrada por una o más de las organizaciones de la comunidad o de forma mixta, y puede existir dentro o fuera de las instalaciones de la comunidad (Vennam, 2020).

Nube Híbrida

Una nube híbrida es una combinación de dos o más de los modelos de implementación en la nube. Usualmente, una empresa implementa sus servicios menos críticos y de bajo riesgo en una nube pública y sus aplicaciones críticas en su nube privada. Este modelo permite una obtener una seguridad en las aplicaciones críticas y con adopción de nubes públicas reducir costos y obtener más opciones de aplicación (Murugesan & Bojanova, 2016).

Proveedores en la nube

El informe de Gartner (2020) que evalúa a los proveedores de la nube tanto de IaaS y PaaS (o CIPS, del inglés Cloud infrastructure and platform services) en base a la capacidad de ejecución e integridad de la visión. Describe que el mercado está dominado en gran parte por 3 proveedores denominados líderes: Amazon Web Services, Google Cloud Platform y Microsoft Azure. A estos le siguen como jugadores de nicho: Alibaba Cloud (gran uso en China y países asiáticos), Oracle Cloud, IBM Cloud y Tencent Cloud, como se indica en la figura 20.

Figura 20

Cuadrante mágico para CIPS



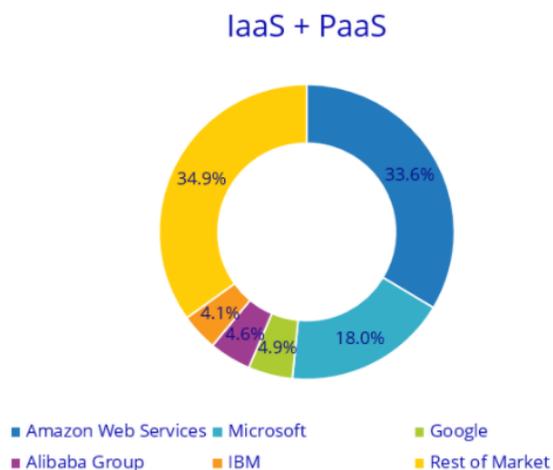
Nota. Se presentan el cuadrante mágico para IaaS y PaaS, (Gartner, 2020).

Por otro lado, la Corporación internacional de datos (2020) (IDC, del inglés International Data Corporation) afirma que el mercado mundial de servicios de nube pública (IaaS, PaaS y SaaS) creció un 26,0% en 2019 con ingresos de \$233,4 mil millones. Además, destaca que en el 2019 la cuota de mercado para SaaS es del 63.6%, mientras que para IaaS es del 21.0% y para PaaS es del 15.4%.

En la figura a continuación se muestra la cuota de mercado respecto a IaaS y PaaS, donde se puede observar que Amazon Web Services y Microsoft Azure capturaron más de la mitad de los ingresos del mercado. A su vez, en la figura 21 se observa la cuota de mercado de SaaS, donde se destaca que los 5 principales proveedores en este ámbito tienen menos del 30% y el resto del mercado más del 70%.

Figura 21

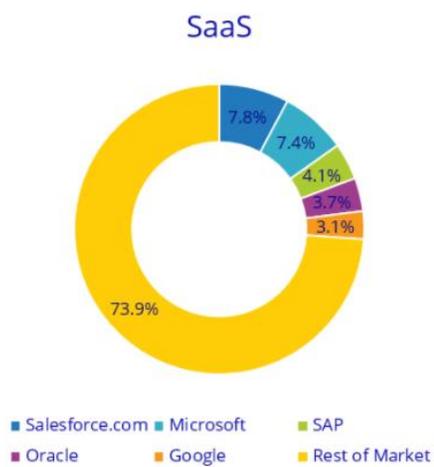
Cuota de Mercado de IaaS y PaaS en el 2019



Nota. Se presenta la cuota de mercado conformada por IaaS y PaaS en el año 2019,
(International Data Corporation, 2020)

Figura 22

Cuota de Mercado de SaaS en el 2019



Nota. Se presenta la cuota de mercado conformada por SaaS en el año 2019,
(International Data Corporation, 2020)

En las secciones siguientes, se proporciona una descripción general de los principales proveedores de servicios en la nube pública.

Amazon Web Services

Amazon Web Services (AWS) es el proveedor líder en la prestación de servicios en la nube y pone a disposición más de 175 servicios en la nube. La gama de clientes de esta plataforma abarca diversos sectores económicos, así como también tamaños empresariales, destacando diversas organizaciones del sector público de varias naciones en el mundo. Entre sus clientes se encuentran General Electric, Coca Cola, Netflix, Philips, Kellogg's, Pinterest, la NASA, Adobe, entre otros (Amazon Web Services, Inc., 2020).

Entre los servicios que AWS ofrece se encuentra el análisis, la integración de aplicaciones, realidad aumentada y realidad virtual, blockchain, aplicaciones empresariales, base de datos, herramientas para desarrolladores, Internet de las cosas, herramientas de machine learning, almacenamiento, computación, servicios multimedia, entre otros (Amazon Web Services, Inc., 2020).

Google Cloud Platform

Google Cloud Platform (GCP) es uno de los líderes en la prestación de servicios de IaaS y PaaS en la nube y ofrece más de 90 servicios en la nube. GCP al igual que AWS tiene clientes en varios sectores desde las empresas emergentes hasta compañías de presencia internacional.

Entre sus clientes se destacan Spotify, FedEx, Toyota, Nintendo, PayPal, Twitter, BBVA, Mercado Libre, entre otros (Google Cloud, 2018).

GCP ofrece una variedad de soluciones en la nube como la modernización de aplicaciones, inteligencia artificial y aprendizaje automático, plataforma de aplicaciones empresariales, migración y gestión de datos, transformación digital, modernización de infraestructura, analíticas, entre otros (Google Cloud, 2018).

Microsoft Azure

Microsoft Azure se encuentra entre los tres principales proveedores de servicios de IaaS y PaaS en la nube y dispone de más de 200 productos y servicios en la nube. Azure al ser uno de los líderes de servicios en la nube consta con clientes en todos los sectores y diversos tamaños. Entre sus clientes están Bosch, Cloud9, Renault, eBay, BMW, CenturyLink, LinkedIn, Schneider Electric, Nokia, Telefónica, Vodafone, T-Mobile, entre otros (Microsoft, 2014).

Azure pone a disposición un amplio catálogo de productos y servicios en la nube como inteligencia artificial y machine learning, analíticas, computación, contenedores, bases de datos, herramientas de desarrollo, DevOps, Internet de las cosas, redes, almacenamiento, seguridad, aplicaciones web, entre otros (Microsoft, 2014).

IBM Cloud

IBM Cloud se encuentra entre los jugadores de nicho según el cuadrante mágico de Gartner de IaaS y PaaS. Esta plataforma ofrece más de 170 productos y servicios en la nube.

Alrededor de 20 industrias cuentan con los servicios proporcionados por esta plataforma, demostrando así una gran trayectoria y experiencia. Entre sus clientes se destacan Panasonic, Whirlpool, Allianz, American Airlines, Honeywell, UBank, Japan Airlines, Creval, Bitly, entre otros (Vennam, 2020).

Entre las soluciones que IBM Cloud ofrece se encuentran la modernización de aplicaciones, inteligencia artificial, backup y recuperación ante desastres, chatbots, aplicaciones nativas en la nube, servidores dedicados, servidores privados virtuales, DevOps, computación de borde, computación de alto rendimiento, nube híbrida, integración de SaaS, sitios web y aplicaciones web, entre otros (Vennam, 2020).

Para concluir se presenta una tabla comparativa entre los principales proveedores en la nube tomando en cuenta los servicios a utilizar de las mismas y otras características.

Tabla 1

Comparación de las plataformas en la nube

Parámetro	Amazon Web Service	Google Cloud Platform	IBM Cloud	Microsoft Azure
Cuadrante mágico de Gartner (2020)	Líder	Líder	Jugador de Nicho	Líder
Cuota de Mercado según IDC (2019)	33.6%	4.9%	4.1%	18.0%
Diversos Sistemas Operativos	Si	Si	Si	Si

Número de servicios	> 175	> 90	> 170	> 200
Auto-escalable	Si	Si	Si	Si
Asistentes Virtuales	Si	Si	Si	Si
Texto a Voz	Si	Si	Si	Si
Voz a texto	Si	Si	Si	Si
PLN	Si	Si	Si	Si
Microservicios	Si	Si	Si	Si
Serverless	Si	Si	Si	Si
Computing				
Disponibilidad SLA	99.99%	99.99%	100.00%	99.9%

Nota: En la tabla se muestra la comparación en base a diferentes parámetros de las plataformas en la nube (Gupta J. , 2019).

Procesamiento del Lenguaje Natural

Definición

El procesamiento del lenguaje natural (PNL) se refiere a una rama de la informática, más específicamente, se refiere a la rama de la inteligencia artificial o IA, que tiene como objetivo permitir que las computadoras comprendan el texto y el lenguaje hablado como los humanos. PLN combina la lingüística computacional (modelado del lenguaje humano basado en reglas) con estadísticas, aprendizaje automático y modelos de aprendizaje profundo. En resumen, estas tecnologías permiten a las computadoras procesar el lenguaje humano en forma de texto o datos de voz y "comprender" su significado completo en función de las intenciones y sentimientos del hablante o del autor. (IBM Cloud Education, 2020)

El PLN es el principal actor en diversos programas que permiten la traducción de texto de un idioma a otro, así como también los programas que actúan basados en comandos hablados. Los sistemas de GPS operados por voz, así como los asistentes digitales, en especial el software de dictado de voz a texto y diversos chatbots de servicio al cliente también utilizan las bondades del PLN.

El PLN desempeña un papel cada vez más importante en las soluciones empresariales que ayudan a agilizar las operaciones comerciales, aumentar la productividad de los empleados y simplificar los procesos comerciales (IBM Cloud Education, 2020).

Plataformas de PLN en la Nube

Existen varias plataformas en el mercado que permiten realizar PLN a través de sus propias herramientas entre las cuales se destacan:

Amazon Comprehend: es un servicio de PLN proporcionado por la plataforma AWS que utiliza el aprendizaje automático para analizar textos, proporcionando APIs para la extracción de frases o palabras clave, análisis de opiniones, reconocimiento de entidades, modelado de temas y detección de idiomas integrando PLN en las aplicaciones. Las APIs usan el formato JSON, que puede ser usado por diversas aplicaciones (AWS, 2018).

Natural Language: es un servicio de PLN proporcionado por la plataforma Google Cloud que utiliza el aprendizaje automático para mostrar la estructura y el significado de los textos. Es posible la separación de información respecto a personas, lugares o entidades, permitiendo una comprensión profunda de las opiniones que pueden darse en redes sociales o las

conversaciones de los clientes de una determinada empresa. Esta herramienta permite analizar textos e integrarlos en la nube almacenamiento de documentos (Google Cloud, 2016).

Watson Natural Language Understanding: es un servicio de PLN proporcionado por la plataforma IBM Watson que analiza el texto de datos no estructurados, incluyendo páginas web, redes sociales y más. A través del kit de herramienta que realizan PLN es posible identificar conceptos, palabras clave, categorías, semántica y emociones, logrando clasificar los textos, extraer y reconocer entidades y el útil análisis de sentimientos y resumen general del texto en cuestión (IBM Cloud Education, 2020).

Language Understanding: es un servicio de PLN proporcionado por la plataforma Azure creada por Microsoft que permite crear aplicaciones capaces de comprender el lenguaje natural. Con la tecnología de enseñanza automática, es posible crear modelos de lenguaje personalizados de aprendizaje automático que interpretan los objetivos del usuario y extraen información clave de frases conversacionales, todo sin ninguna experiencia de aprendizaje automático (Microsoft Azure, 2016).

Características de cada Plataforma

Amazon Comprehend

Extracción de Frases Claves: Esta plataforma cuenta con una API que permite la extracción de frases clave, la cual genera una frase clave o tema de conversación, además adicionando una puntuación de confianza para cada una de estas frases y/o temas.

Análisis de Opiniones: Esta plataforma cuenta con una API que realiza el análisis de opiniones, es decir genera la opinión general de un texto permitiendo así que se la considere positiva, negativa, neutra o una combinación de las anteriores.

Análisis Sintáctico: Esta plataforma cuenta con una API llamada Syntax por medio de la cual se permite a los clientes analizar texto tokenizado y categorías gramaticales, e identificar etiquetas de palabras, así como también límites en el texto contemplando diferentes palabras como sustantivos y adjetivos.

Reconocimiento de Entidades: Esta plataforma cuenta con una API de reconocimiento de entidades devuelve entidades con nombre las cuales pueden ser persona o personas, lugar, ubicación entre otras y las cuales se clasifican automáticamente a partir del texto proporcionado.

Entidades Personalizadas: Esta plataforma cuenta con una API llamada Custom Entities la cual permite identificar términos específicos de su dominio. Además, gracias al uso de AutoML y con la ayuda de índices privados los cuales podrían ser listas de números o textos, se pueden entrenar modelos para localizar estos índices en bloques de textos posteriormente ingresados.

Detección de Idioma: Esta plataforma cuenta con una API que permite el reconocimiento de manera automática de texto escrito en alrededor de 100 idiomas, además genera un idioma dominante relacionado al texto ingresado el cual será acompañado de una puntuación de confianza para confirmar el lenguaje dominante.

Clasificación Personalizada: Esta plataforma cuenta con una API que permite realizar una clasificación personalizada, lo cual permite de manera sencilla crear un modelo de clasificación de texto a través del uso de etiquetas específicas previamente proporcionadas por el usuario.

Modelado de Tema: El modelado de temas identifica términos o temas relacionados de un conjunto de documentos almacenados en Amazon S3. Mediante la identificación de los temas más comunes de la colección y su agrupamiento se puede determinar qué documentos pertenecen a qué tema.

Compatibilidad con varios Idiomas: Esta plataforma cuenta con una herramienta la cual permite el análisis de textos en diferentes idiomas tales como: inglés, alemán, francés, español o italiano, lo cual permite detectar diversos idiomas y convertirlos al idioma de preferencia.

Natural Language

API REST Integrada: Esta plataforma permite acceder a los servicios de PLN mediante la API REST. Y el análisis de texto puede realizarse por petición o estar integrado con Cloud Storage.

Análisis Sintáctico: Esta plataforma permite extraer tokens y frases importantes de un texto determinado, identificando las diversas categorías gramaticales y permite adicionalmente crear árboles de análisis de dependencias de cada uno de los tokens o frases importantes previamente extraídos.

Análisis de Entidades: Esta plataforma identificar diferentes entidades dentro de un texto, por ejemplo: facturas, contratos, recibos y categorizar estas entidades por tipos como fechas, persona, organización, ubicaciones, productos entre otros.

Extracción de Entidades Personalizadas: Esta plataforma cuenta permite realizar el proceso previamente señalado como “Análisis de Entidades” pero etiquetando estas entidades con frases o palabras claves específicas del dominio propio.

Análisis de Opinión: Esta plataforma cuenta con una herramienta la cual permite conocer las diversas opiniones, actitudes y sobretodo sensaciones generales que se extraen de un bloque de texto determinado.

Análisis de Opinión Personalizado: Esta plataforma cuenta permite realizar el proceso previamente señalado como “Análisis de Opinión” teniendo en cuenta las puntuaciones de diversas opiniones específicas del dominio propio.

Clasificación de Contenido: Esta plataforma cuenta con una herramienta que permite clasificar o dividir el contenido de un bloque de texto en alrededor de 700 categorías predefinidas.

Clasificación de Contenido Personalizada: Esta plataforma cuenta permite realizar el proceso previamente señalado como “Clasificación de Contenido” teniendo en cuenta que se usan datos propios de entrenamiento a través de los cuales se crean diversas etiquetas para modelos.

Varios Idiomas: Esta plataforma analiza bloques de texto en diversos idiomas entre los que se destacan las lenguas romances, inglés, japonés, coreano, ruso, alemán y chino teniendo en cuenta que el este último puede ser simplificado o tradicional.

Modelos Personalizados: Esta plataforma permite entrenar modelos personalizados de aprendizaje automático de manera fácil.

Información sobre la Estructura Espacial: Esta plataforma usa información ofrecida gracias a la estructura y el diseño de los documentos en el formato PDF de manera que sea posible optimizar la extracción de entidades personalizada.

Compatibilidad con Conjuntos de Datos de Gran Tamaño: Esta plataforma admite alrededor de 5000 etiquetas destinadas a la clasificación y la cantidad de 1 millón de documentos de hasta 10 MB cada uno, por lo tanto, es posible acceder a casos prácticos complejos que manejan gran cantidad de datos no es un problema para la plataforma.

Watson Natural Language Understanding

Entidades: Es posible identificar personas, lugares, eventos y otros tipos de entidades mencionados en el contenido utilizando las capacidades listas para usar ofertadas por cada una de las plataformas previamente señaladas.

Categorías: Es posible categorizar con granularidad los contenidos insertados, por ejemplo, la plataforma permite una jerarquía de clasificación de cinco niveles.

Conceptos: Es posible identificar conceptos de alto nivel los cuales pueden estar referenciados directa o indirectamente en el contenido ingresado.

Emociones: Es posible extraer las diversas emociones transmitidas por frases objetivo específicas o por el documento en su conjunto, entre las emociones identificadas se encuentran: alegría, enojo, tristeza, miedo, entre otras.

Palabras clave: Es posible identificar las palabras claves o las palabras más relevantes presentes en el contenido ingresado.

Sentimientos: Es posible analizar el sentimiento hacia frases objetivo específicas y del documento en su conjunto, teniendo en cuenta la siguiente categorización: positivo, negativo o neutral.

Relaciones: Es posible comprender la relación entre dos entidades dentro del contenido ingreso e identificar el tipo de relacionamiento presente entre las mismas.

Metadata: Es posible extraer de manera ágil y rápida información relevante de un documento como autor, título, imágenes, fechas de publicación, entre otras que son importantes y brindan información sobre los datos.

Roles Semánticos: Es posible analizar oraciones en forma sujeto, acción y objeto, logrando identificar entidades y palabras clave que son sujetos u objetos de una acción.

Language Understanding

Modelos de Lenguaje Personalizados Semánticos: Es crear modelos de lenguaje personalizados específicos para cada caso de uso con herramientas de desarrollador y experiencia en el portal para simplificar el etiquetado de los bloques de texto.

Solución de Idioma Personalizada: Gracias al aprendizaje automático es posible iniciar con información sin ningún tipo de etiquetados, a partir de esta se puede entrenar diversos modelos de forma rápida e interactiva de manera que su desarrollo se vea acelerado. Esta plataforma proporciona entidades, funciones y aplicaciones prediseñadas para poner en marcha cualquier tipo de proyecto que involucre el PLN.

Soluciones Integrales de Lenguaje Natural: A través de esta plataforma es posible integrar a la perfección con Azure Cognitive Services como Text Analytics y Speech, así como Azure Bot Service para una solución conversacional de un extremo a otro.

Arquitectura Orientada a Servicios

Definición

Diversos autores y organismos de estandarización han elaborado algunas definiciones para la Arquitectura Orientada a Servicios (SOA, del inglés Service Oriented Architecture), exponiendo diferentes puntos de vista de este concepto. A continuación, se citan algunas definiciones:

El NIST define a SOA como un conjunto extensible de servicios, que se comunican entre sí, implicando el simple paso de datos o la coordinación de alguna actividad entre dos o más servicios (Singhal, Winograd, & Scarfone, 2007).

Una definición más completa es la expuesta por Mahmoud (2005), donde define que SOA es un estilo arquitectónico para crear aplicaciones de software que utilizan servicios disponibles en una red como la web, promoviendo la reutilización de software. Las aplicaciones en SOA se crean en función de los servicios. Un servicio es una implementación de una función o tarea comercial bien definida, y dichos servicios pueden ser consumidos por los clientes en diferentes aplicaciones o procesos comerciales con distintas tecnologías.

Safanov (2016) describe a SOA como uno de los enfoques más actualizados para el desarrollo de software, que se fundamenta en representar el software como un conjunto extensible de servicios (típicamente, servicios web). Estos servicios están disponibles para que los usuarios los consuman, los agreguen a su espacio de trabajo y los supervisen.

Características de SOA

Autores como Endrei y otros (2004) y Mahmoud (2005) señalan que las principales características de SOA son:

- Los servicios son componentes de software con interfaces bien definidas que son independientes de la implementación.
- Separación de la interfaz de servicio (el qué) de su implementación (el cómo).

- Los clientes que consumen los servicios no tienen que preocuparse de cómo estos servicios ejecutarán sus solicitudes.
- Los servicios realizan tareas predeterminadas y son autónomos.
- Los servicios se pueden descubrir de forma dinámica.
- Los servicios compuestos se pueden construir a partir de agregados de otros servicios.
- Promete interoperabilidad entre aplicaciones y tecnologías heterogéneas

Roles en SOA

SOA usa el paradigma publicar, buscar, vincular e invocar que se muestra en la figura 2.13, donde además se ilustra los roles de las entidades que interactúan en SOA (proveedor, consumidor, registro).

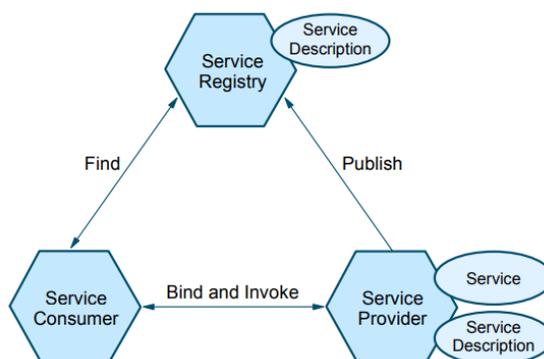
Proveedor de servicios: tiene la función de aceptar y ejecutar solicitudes de los consumidores. Además, se encarga de publicar una descripción de sus servicios junto con el contrato de interfaz en el registro de servicios (Endrei, y otros, 2004).

Consumidor de Servicios: busca el servicio necesitado en el registro de servicios que cumpla sus criterios. Si se encuentra el servicio, se vinculará al servicio y se lo invocará de acuerdo con la descripción del servicio (Endrei, y otros, 2004).

Registro de servicios o Broker: tiene la finalidad de permitir el descubrimiento de servicios al consumidor. Este registro contiene un repositorio de los servicios disponibles y permite la búsqueda de interfaces de proveedores de servicios para consumidores de servicios interesados (Endrei, y otros, 2004).

Figura 23

Componentes en SOA



Nota. Se presentan los componentes generales de SOA, (Endrei, y otros, 2004)

Arquitectura de Microservicios

La Arquitectura de Microservicios es una variante de SOA, varios autores y organismos de estandarización han planteado diferentes conceptos de los Microservicios desde diferentes perspectivas. A continuación, se describen algunas definiciones:

El NIST propone que un Microservicio es un componente básico resultado de la descomposición de una aplicación en servicios autónomos que se comunican entre sí mediante un protocolo de comunicaciones estándar y un conjunto de API establecidas, independientes de cualquier proveedor o tecnología. Los Microservicios se basan en SOA y en su mayoría son implementados en contenedores (Karmel, Chandramouli, & Iorga, 2016).

IBM Cloud Education (2019) ofrece un concepto desde la perspectiva de la computación en la nube: Los Microservicios son un modelo de arquitectura nativa de la nube en la que una sola aplicación está compuesta de varios servicios más pequeños. Estos servicios se pueden implementar de forma independiente y se pueden comunicarse entre sí.

Características de los Microservicios

Chandramouli (2019) e IBM Cloud Education (2019) señalan que las principales características y principios de la Arquitectura de Microservicios son:

- **Agilidad:** gracias a su acoplamiento flexible y modularidad permite una modificación e implementación independientes y más rápidas sin afectar a otros microservicios de la misma aplicación.
- **Escalabilidad:** los microservicios pueden escalar de forma independiente.
- **Usabilidad:** el uso de APIs bien definidas facilita la integración de varios microservicios.
- **Herramientas de automatización:** facilitan la configuración y la implementación sin errores.

Principios:

- Cada microservicio se debe administrar, escalar, actualizar e implementar de forma independiente de otros microservicios.
- Cada microservicio debe tener una única función y operar en un contexto limitado.
- Todos los microservicios deben estar diseñados para fallas y recuperaciones constantes.

- Se deben reutilizar los servicios confiables existentes (por ejemplo, bases de datos, cachés, directorios).

Comparación entre SOA y Microservicios

Karmel, Chandramouli y Iorga (2016), Chandramouli (2019) e IBM Cloud Team (2020) señalan varias de las diferencias existentes entre SOA y la Arquitectura de Microservicios en la tabla 2.

Tabla 2

Diferencias SOA y Arquitectura de Microservicios

Service-Oriented-Architecture (SOA)	Arquitectura de Microservicios
Servicios autónomos	Servicios desplegados independientemente
Comunicación entre servicios mediante un bus de servicios	Comunicación entre servicios mediante interfaces y protocolos de comunicación estándar y livianos
Dependientes del Estado (stateful) y requiere mapeo de dependencias de servicio cuando se introducen cambios	Independientes del estado (stateless) y menos frágiles cuando se introducen cambios
Tiempos de inicio-parada más largos	Tiempos de inicio-parada cortos
Construido alrededor de servicios	Construido alrededor de capacidades
Alcance empresarial	Alcance de aplicación

REST

SOA puede ser implementado mediante diversas tecnologías, entre estas, el estándar de servicios web REST (del inglés, representational state transfer). REST surge como un estilo de arquitectura derivado de varios estilos propuesto por Fielding (2000), que describe un conjunto de principios de ingeniería de software para la creación de servicios web.

Safanov (2016) menciona las empresas grandes e influyentes están utilizando cada vez más servicios web basados en REST debido a su rapidez y a su facilidad de uso. Principalmente, en el área de la computación en la nube para organizar de forma eficiente los servicios web en la nube.

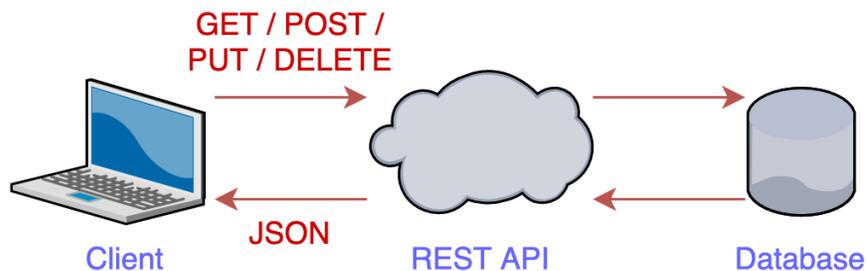
Funcionamiento API REST

En un sistema con API REST, el cliente envía una solicitud al servidor utilizando los métodos HTTP (GET, POST, PUT y DELETE) para realizar funciones de base de datos estándar como leer, crear, actualizar y eliminar e indicando un recurso identificado por un URI (identificador de recursos uniforme). Una vez procesada la solicitud, el servidor envía una respuesta o representación del recurso utilizando formatos como JSON, XML, HTML, entre otros, como se indica en la figura 24 (Safanov, 2016; IBM Cloud Education, 2019).

Adicionalmente, en las peticiones enviadas por el cliente se incluye otra información importante como: metadatos, autorizaciones, URIs, almacenamiento en caché, cookies y más mediante los encabezados y parámetros (Doglio, 2015).

Figura 24

REST API



Nota. Se presentan los métodos generales de un REST API y la comunicación que tiene con una base de datos y un cliente, (Doglio, 2015).

Principios API REST

Las API REST se implementan mediante cualquier lenguaje de programación y admiten una variedad de formatos de datos como JSON o XML. Sin embargo, deben alinearse con los seis principios de diseño de REST (Doglio, 2015; IBM Cloud Education, 2021):

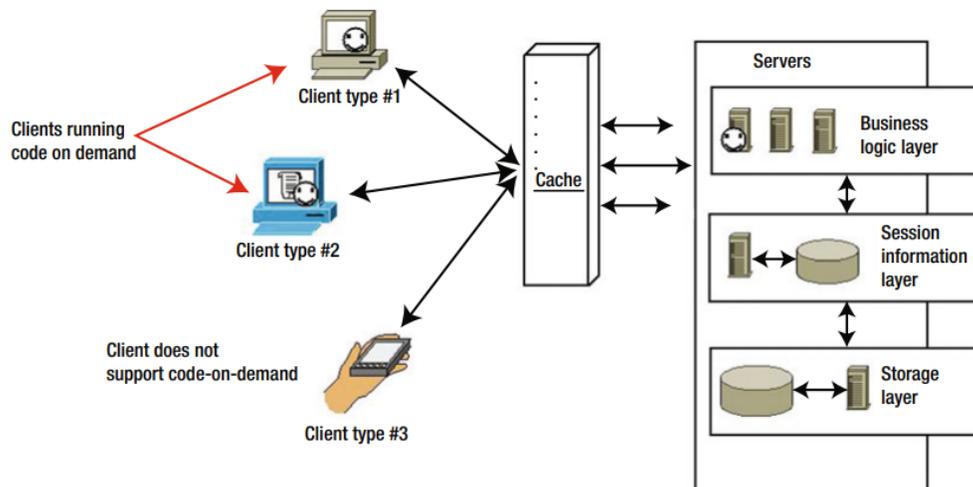
- **Interfaz uniforme:** cada recurso debe tratarse mediante un único identificador (URI), los recursos se manipulan a través de diferentes representaciones diferentes sin cambiar nunca su identificador. La comunicación debe ser mediante mensajes autodescriptivos, que pueden incluir metadatos con detalles adicionales. La representación del estado de un recurso incluye enlaces a recursos relacionados.
- **Sistema en capas:** las solicitudes y respuestas pasan por diferentes capas, por tanto, las API REST deben diseñarse de modo que ni el cliente ni el servidor puedan saber si se

comunican con la aplicación final o con un intermediario. Los intermediarios comúnmente se usan para seguridad, almacenamiento en caché y equilibrio de carga.

- **Servidor – Cliente:** las aplicaciones cliente y servidor tienen que ser independientes entre sí y pueden ser implementadas en diferentes lenguajes de programación. La única información que debe conocer la aplicación cliente es el URI del recurso solicitado.
- **Sin-Estado:** no se requiere que un servidor web almacene el estado de las aplicaciones cliente. Por tanto, los clientes deben enviar toda la información necesaria para cada petición con el servidor web. Las aplicaciones de servidor no deben almacenar información relacionada con la solicitud de los clientes.
- **Almacenamiento en Caché:** cuando sea posible, los recursos deben almacenarse en caché en el lado del cliente o del servidor. El caché permite mejorar el rendimiento en el lado del cliente, mientras aumenta la escalabilidad en el lado del servidor.
- **Código bajo demanda:** Las API REST generalmente envían recursos estáticos, pero en ciertos casos, las respuestas también pueden contener código ejecutable. En estos casos, el código solo debe ejecutarse bajo demanda.

Figura 25

Principios API REST



Nota. Se presentan los principios generales de un API REST, (Doglio, 2015).

Capítulo III

Desarrollo e Implementación

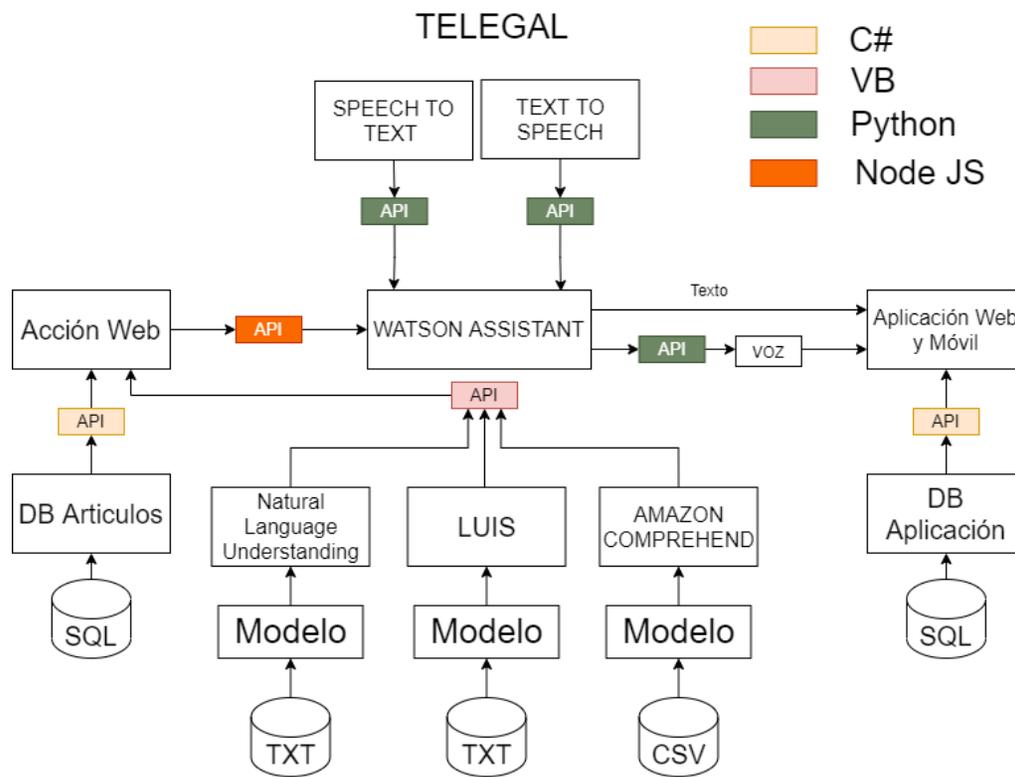
En el presente capítulo se presenta el desarrollo e implementación del sistema como tal, iniciando por el análisis de la arquitectura del sistema, donde se detallan los servicios creados para el backend como cada uno de los modelos para las diferentes plataformas de servicios de computación en la nube, así como también las diferentes APIs que permiten la interconexión y comunicación entre los modelos, integrando todo esto en un asistente virtual el cual puede ser consumido a través del uso de texto o por medio de comandos de voz. Finalmente, se detallan los componentes del frontend del proyecto donde se creó una interfaz gráfica la cual permite a los usuarios consumir los diferentes asistentes creados a través de una aplicación web y una aplicación móvil.

Arquitectura del Sistema

A continuación, se presenta un diagrama detallado con cada uno de los módulos que componen la arquitectura del sistema la cual está basada en una arquitectura orientada a los servicios, permitiendo así que las secciones que conforman el backend y el frontend puedan funcionar de manera independiente y aislada, es importante señalar que el sistema planteado se encuentra completamente implementado en la nube, los modelos y servicios se consumen desde servidores remotos, así mismo la aplicación web y móvil se encuentran disponibles a través del internet.

Figura 26

Arquitectura del Sistema



Se iniciará el análisis de la arquitectura del sistema por sus componentes

fundamentales, que son los modelos de machine learning desarrollados en tres plataformas cloud, los cuales permiten realizar el procesamiento de lenguaje natural orientados a encontrar la relación entre los delitos de telecomunicaciones planteados en el Capítulo 2 y un texto de interés específico.

- Natural Language Understanding:** este modelo corresponde a los servicios cloud brindados por IBM Watson, a través de sus servicios Knowledge Studio y Natural Language Understanding es posible encontrar relaciones e intenciones presentes en un texto ingresado, la forma de entrenar a este modelo es mediante el uso de archivos de texto plano que son transformados a objetos JSON con las palabras de interés.

- **LUIS:** este modelo corresponde a los servicios cloud brindados por Microsoft Azure, a través de este servicio de PLN es posible crear un asistente virtual, el cual reconoce intenciones presentes en un texto ingresado, la forma de entrenar a este modelo es mediante el uso de archivos de texto plano y una interfaz disponible para el ingreso de casos específicos.
- **Amazon Comprehend:** este modelo corresponde a los servicios cloud brindados por Amazon Web Services, a través de su servicio Amazon Comprehend, es posible crear un clasificador y encontrar intenciones presentes en un texto ingresado, la forma de entrenar a este modelo es mediante el uso de archivos del tipo CSV.

Todos estos modelos pueden ser consumidos a través del uso de diferentes lenguajes de programación para este caso, se ha utilizado el lenguaje Visual Basic, creando un Web API en el framework de .NET es posible tener un control centralizado del consumo de cada modelo.

El sistema posee dos bases de datos relacionales del tipo SQL (Structured Query Language) completamente aisladas, las cuales tienen propósitos diferentes los cuales son:

- **DB Artículos:** esta base de datos posee los artículos relacionados directamente con las telecomunicaciones y sus diversos delitos, los cuales están establecidos tanto en las leyes COIP y LOT, siendo en total 15 artículos.
- **DB Aplicación:** esta base de datos posee toda la información que se maneja tanto en la aplicación web y móvil, los datos son almacenados de manera permanente.

Cada una de estas bases de datos, poseen Web APIs para su consumo las cuales han sido desarrolladas en el lenguaje de programación C# en el framework de .NET

Otro aspecto fundamental en el sistema es la presencia del asistente virtual, el cual ha sido realizado gracias al servicio de Watson denominado Watson Assistant, a través de este asistente es posible comunicar todos los aspectos previamente descritos haciendo uso de las Web APIs, tanto el API de los modelos, como el API del DB Artículos han logrado ser acoplados dentro de una Web Action, utilidad presente en la nube de IBM, mediante la cual se genera un a URL que permite invocar esta acción desde cualquier aplicación, este Web Action como tal ha sido desarrollada en el lenguaje de programación Node.js

Así mismo los servicios de Speech to Text y también Text to Speech han sido acoplados al servicio de Watson Assistant permitiendo tener disponibles tanto un asistente virtual comandado por texto, como un asistente virtual comandado por voz, los diferentes APIs que utilizan estos dos servicios han sido simplemente utilizados y están desarrollados en el lenguaje de programación Python.

Una vez integrados los servicios al Watson Assistant, el asistente puede ser consumido mediante el uso del lenguaje de programación JavaScript, de manera que puede ser integrado a cualquier aplicación basada en web, de esta forma a través de la aplicación web y la aplicación móvil es posible tener una instancia del asistente virtual. Finalmente, el Web API de la base de datos de la aplicación se acopla con el asistente y la aplicación web desarrollada usando el framework de .NET, y también se acopla a la aplicación móvil desarrollada usando la plataforma para .NET denominada Xamarin.

Backend

A continuación, se detallan con mayor profundidad ciertos módulos orientados a satisfacer la lógica del sistema, así como también las diferentes conexiones con las bases de datos previamente descritas y sobre todo el consumo de los modelos de las diferentes plataformas orientados a analizar textos con la finalidad de determinar delitos de las telecomunicaciones que puedan cometerse en estos textos específicos.

Base de Datos de Artículos

La base de datos a la que se refiere esta sección es la DB Artículos, explicada previamente, la cual posee los artículos relacionados estrechamente con las telecomunicaciones presentes tanto en el Código Orgánico Integral Penal Ecuatoriano (COIP) y la Ley Orgánica de Telecomunicaciones (LOT), en cuanto a lo que se refiere a los diversos delitos que podrían llegar a cometerse por parte de cualquier ciudadano y/o empresa pública o privada, en total los artículos seleccionados son los siguientes, para el caso del COIP se tiene en cuenta los artículos 188, 190, 191, 192, 193, 194, 195, 229, 230, 232, 234. Y para la LOT los artículos 117, 118, 119, 120.

Se ha definido para el efecto la tabla denominada "Articulos", la cual posee 6 columnas entras las cuales se tiene:

Figura 27

Detalle de tabla Articulos

Articulos
Id: SMALLINT NOT NULL [PK]
Ley: VARCHAR(4) NOT NULL
Articulo: SMALLINT NOT NULL
Titulo: VARCHAR(256) NOT NULL
Detalle: VARCHAR(2048) NOT NULL
Condena: VARCHAR(256) NOT NULL

- **Id** columna del tipo SMALLINT, a la cual se le definió el atributo de identidad de 1 y 1, lo que significa que genera valores secuenciales automáticos iniciados en el valor 1, e incrementa este valor cada vez que un registro es ingresado a la tabla, esta columna ha sido definida como clave principal de la tabla (PK).
- **Ley** columna del tipo VARCHAR de longitud 4, la cual indica que se puede presentar una serie de caracteres de longitud variable entre 1 y 4, donde se realiza la diferenciación de la ley de la cual proviene el artículo, en este caso puede ser "COIP" o "LOT".
- **Articulo** columna del tipo SMALLINT, la cual representa el número del artículo ya sea tanto para el "COIP" como para la "LOT".
- **Título** columna del tipo VARCHAR de longitud 256, la cual indica que se puede presentar una serie de caracteres de longitud variable entre 1 y 256, donde se almacenan los títulos de cada uno de los artículos insertados.
- **Detalle** columna del tipo VARCHAR de longitud 2048, la cual indica que se puede presentar una serie de caracteres de longitud variable entre 1 y 2048, donde se almacena el detalle de cada uno de los artículos en su totalidad, con cada una de sus secciones y diferentes puntos de consideración, esta información fue extraída de los documentos oficiales de la función judicial del Ecuador.

- **Condena** columna del tipo VARCHAR de longitud 256, donde se almacenan la condena o la multa dependiendo del artículo en cuestión ya que el COIP presenta condenas de privación de la libertad variables, mientras que la LOT presenta multas que deberán ser canceladas por parte de los ciudadanos o empresas prestadoras de servicios, estas multas son un porcentaje de los montos referenciales.

Es importante señalar que esta tabla fue diseñada pensando en una posible expansión del número de artículos y de las diversas leyes que podrían ser consideradas, así mismo el atributo NOT NULL presente en la figura previa, significa que la columna es de carácter obligatorio, es decir, para este caso todas las columnas deben ser insertadas con información válida.

API

Para efectos de uso de la base de datos previamente creada, ya que se trata de una base de datos en la cual solamente se realiza la consulta de la información de los artículos de las diferentes leyes, se creó un servicio web del tipo RESTful utilizando el framework Web API de .NET, la cual es una plataforma ideal para crear aplicaciones del tipo REST que comuniquen varios componentes de software en este caso una base de datos con una aplicación web, esta plataforma utiliza HTTP para realizar estas comunicaciones, por lo cual es posible construir servicios RESTful que pueden llegar a una amplia gama de clientes, entre los que se encuentran los navegadores y los dispositivos móviles englobando así el alcance de este proyecto.

Existen diferentes formatos de datos soportados por el framework Web API entre los principales se encuentran XML (Extensible Markup Language) y JSON (JavaScript Object

Notation), para este y todos los casos de servicios web usados dentro de esta aplicación, se usó el formato JSON el cual es sencillo e ideal para el intercambio de dato; Web API se presenta como una plataforma de desarrollo de servicios web de forma ágil, fácil de mantener y escalable.

Como se detalló en el capítulo 2, existen diversos métodos en lo que se refiere a HTTP, de los cuales los más comunes y utilizados son: GET, PUT, POST, DELETE, a partir de este contexto, se creó un servicio web RESTful utilizando el framework Web API disponible en Microsoft Visual Studio Community usando el lenguaje de programación C#.

Figura 28

Web API Template para crear un Servicio RESTful en Visual Studio Community

Create a new ASP.NET Web Application

Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication
No Authentication
[Change](#)

Add folders & core references

Web Forms
 MVC
 Web API

Advanced

Configure for HTTPS
 Docker support
(Requires [Docker Desktop](#))
 Also create a project for unit tests
WebApplication2.Tests

[Back](#) [Create](#)

Ya que se trabaja con SQL, es necesario crear una cadena de conexión que permitirá acceder al servidor de base de datos, el cual puede estar ser local o remoto, una vez establecida la conexión es posible mediante el uso de dos queries de SQL brindar funcionalidad hacia los endpoints del controlador del Web API. Para este efecto los queries utilizados son:

```
Select * from Articulos
```

El cual permite consultar toda la información alojada dentro de la tabla Articulos de la base de datos del servidor.

```
Select * from Articulos where articulo = ' ' + articulo + ' '
```

El cual permite consultar la información específica de un registro de la tabla Articulos de la base de datos del servidor, para lo cual es necesario definir una variable denominada “articulo” la cual contendrá el número del artículo del que se desea obtener información.

Para transformar el detalle obtenido del servidor de base de datos en un objeto JSON es necesario crear una clase con el modelo recibido, de forma que se pueda mapear cada una de las columnas de la base de datos con un valor dentro del tipo de datos JSON.

```
public class ArticlesModel
{
    public int id { get; set; }
    public string ley { get; set; }
    public int articulo { get; set; }
    public string titulo { get; set; }
    public string detalle { get; set; }
    public string condena { get; set; }
}
```

Este modelo cumple con los tipos de datos de la definición de cada una de las columnas de la tabla "Articulos".

Para obtener un endpoint con cualquiera de los métodos HTTP, es necesario crear un controlador que albergue estas funciones, este controlador extiende de la clase ApiController de System.Web.Http la cual maneja las solicitudes HTTP entrantes y envía las respuestas al cliente que las llamó.

Cada uno de los queries previamente creados será destinado a un endpoint del controlador en este caso se tiene dos endpoints que usan el verbo GET, el primero permite obtener el detalle de todos los artículos que se encuentran almacenados en la base de datos, mientras que el segundo solamente consultará un artículo a la vez, por lo tanto los endpoints realizan un llenado de un objeto del tipo DataTable proveniente del import System.Data con la

respuesta de su query SQL y transformará esta respuesta en una lista del tipo ArticlesModel, con este objeto creado, la respuesta es transformada y retorna un objeto JSON válido hacia el cliente, de esta forma se obtiene un objeto del tipo JsonResult a la salida de los endpoints del Web API con la información resultante de la consulta proveniente del servidor de base de datos.

Es necesario configurar el Web API para que trabaje con el tipo de datos JSON, ya que por defecto la configuración de este tipo de servicio web es XML. Una vez comprobado el funcionamiento de manera local, es posible realizar una construcción del Web API, de manera que se compila el código creado y se obtienen los archivos resultantes, los cuales podrán ser ubicados en un servidor remoto para ser accedidos y consumidos a través del internet.

Modelos de Procesamiento de Lenguaje Natural

Datos de Entrenamiento

Para la creación del CORPUS se realizó una recopilación de diversas fuentes de información referentes a delitos de telecomunicaciones en el Ecuador entre las que se destacan: el detalle extendido de 11 artículos de la Ley Orgánica de Telecomunicaciones (LOT) y de 4 artículos del Código Orgánico Integral Penal (COIP); las noticias de los principales medios de información digital del país permitieron obtener algunos casos comprobados de delitos de telecomunicaciones y finalmente algunos procesos judiciales disponibles en página web de la Función Judicial del Ecuador. Los artículos seleccionados para el entrenamiento de la LOT y el COIP se encuentran listados en la sección anterior.

Una vez analizados cada uno de los artículos vinculados a los delitos de las telecomunicaciones y delitos informáticos establecidos en el COIP y la LOT, se realizó diversos

ejemplos utilizando las palabras claves y verbos que se relacionan con cada artículo para poner entrenar el modelo.

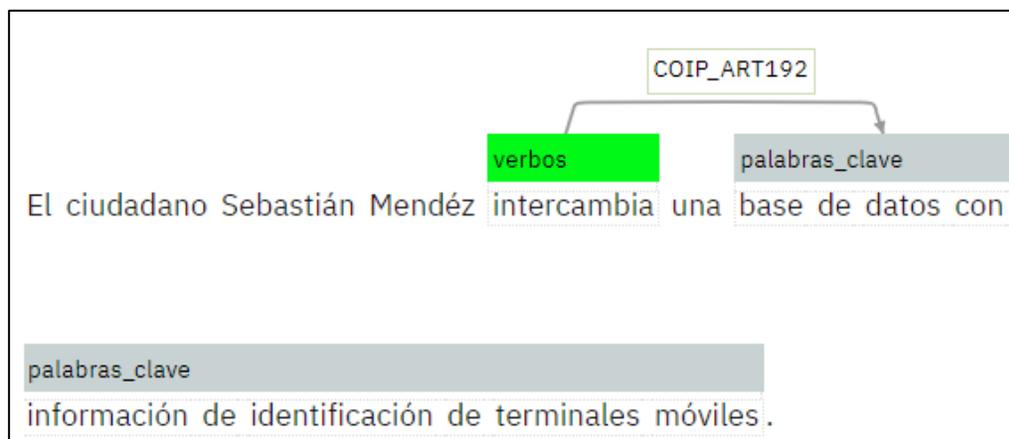
Es importante señalar que los modelos de machine learning construidos y entrenados de cada una de las plataformas se pueden clasificar de la siguiente manera:

- **Modelos Supervisados:** Dentro de estos modelos luego de realizar un preprocesamiento de la información, la misma es etiquetada y se realizan anotaciones con la información que se quiere destacar, así mismo es posible relacionar las entidades o palabras claves permitiendo así tener un mayor control sobre el modelo y su comportamiento como herramienta predictiva, los modelos creados mediante el uso de las plataformas IBM Watson y Microsoft Azure son de este tipo.
- **Modelos No Supervisados:** Dentro de estos modelos posterior a realizar el preprocesamiento de la información, la misma se encuentra en su mayoría sin etiquetas, de manera que a este tipo de modelos no se le realizan anotaciones ni se relacionan sus entidades o palabras claves, el modelo creado mediante la plataforma AWS es de este tipo.

Para el entrenamiento de los modelos supervisados se siguió el siguiente patrón: una vez obtenido el texto de las diversas fuentes de información, se determinan cada uno de los verbos que satisfagan el sentido y la coherencia del mismo, una vez determinados los verbos, es posible complementarlos gracias a la presencia de palabras clave dentro de los diversos textos, etiquetando de esta manera a la información extraída, logrando así una mayor confianza debido a la intervención humana en el anotado y etiquetado de información relevante.

Figura 29

Ejemplo de Anotaciones y Relaciones para modelos supervisados



La cantidad de información que fue introducida a cada uno de los modelos varía dependiendo de los mismos por lo cual será detallada a profundidad en las siguientes secciones, así como también los resultados obtenidos posterior al entrenamiento de cada uno de los modelos.

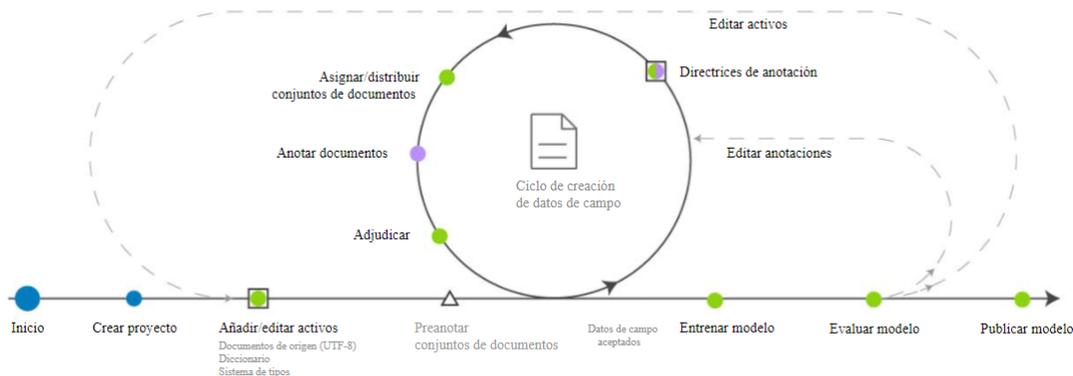
IBM Watson

La plataforma de servicios cloud IBM Watson permite a través de su servicio Watson Knowledge Studio, la creación de un modelo de aprendizaje automático, el cual brinda la posibilidad de entrenar un modelo de manera personalizada con el objetivo final de identificar entidades y relaciones que puedan resultar de interés, en diferentes textos o documentos de manera general. El modelo ofrecido por esta plataforma es de tipo supervisado.

A continuación, se presenta el diagrama general para crear un modelo de machine learning usando Knowledge Studio.

Figura 30

Flujo de trabajo de la creación del modelo de machine learning



Nota. Se presentan el flujo de trabajo para crear un modelo de machine learning en la plataforma IBM Watson,(IBM Cloud, 2020).

De forma general los pasos para crear y mejorar el modelo son los siguientes:

- **Creación del proyecto:** Es posible crear un espacio de trabajo en la plataforma, donde se definirán todo tipo de recursos usados para crear y mejorar el modelo, donde los más importantes son:
 - **Definición de entidades:** A través del uso de entidades, es posible determinar la información importante y relevante para el usuario. En este caso se han usado cuatro entidades. Las cuales se detallan a continuación:

Figura 31

Entidades creadas en Watson Knowledge Studio

<input type="checkbox"/>	Entity Type Name
<input type="checkbox"/>	finalidad
<input type="checkbox"/>	localizacion
<input type="checkbox"/>	palabras_clave
<input type="checkbox"/>	verbos

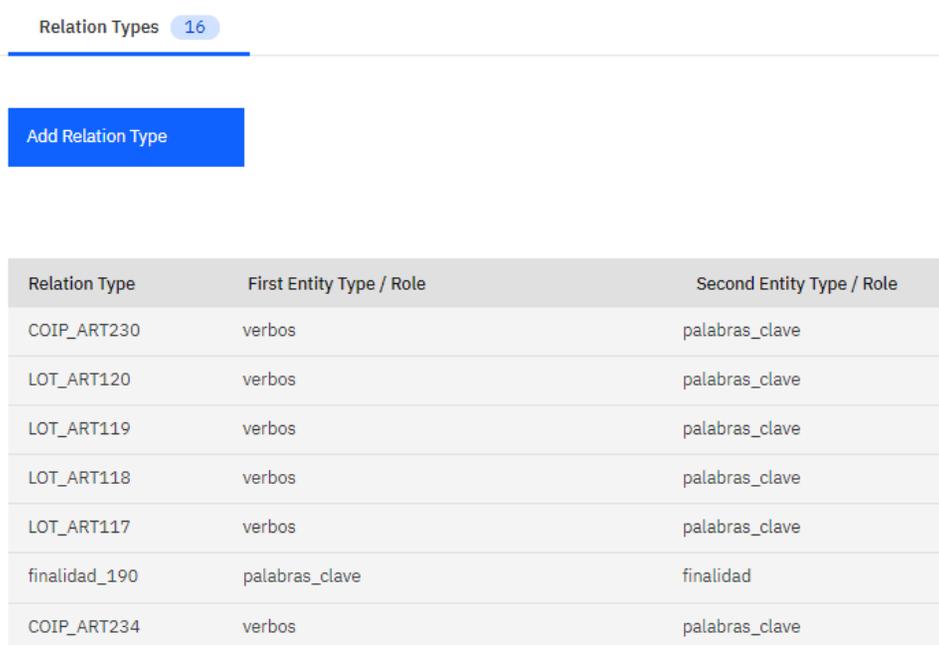
- **Finalidad:** Es usado para reconocer las diferentes intenciones que tienen ciertos artículos tanto del COIP como de la LOT, entre las que destacan el artículo 190 del COIP.
 - **Localización:** A través de esta entidad es posible determinar el nombre de alrededor de 200 ciudades del Ecuador, donde se reconocerán en documentos o textos ingresados.
 - **Palabras Clave:** Esta entidad permite determinar las palabras clave que tiene cada uno de los artículos del COIP y la LOT.
 - **Verbos:** Esta entidad al igual que la previa, permite determinar los verbos presentes en cada uno de los artículos del COIP y la LOT.
- **Definición de relaciones:** a través del uso de relaciones es posible encontrar y comparar dos o más entidades, de manera que solamente cuando se cumplan

ciertas condiciones, se podrá determinar la relación existente entre las mismas.

En este caso se han definido 16 relaciones, por ejemplo:

Figura 32

Relaciones creadas en Watson Knowledge Studio



Relation Type	First Entity Type / Role	Second Entity Type / Role
COIP_ART230	verbos	palabras_clave
LOT_ART120	verbos	palabras_clave
LOT_ART119	verbos	palabras_clave
LOT_ART118	verbos	palabras_clave
LOT_ART117	verbos	palabras_clave
finalidad_190	palabras_clave	finalidad
COIP_ART234	verbos	palabras_clave

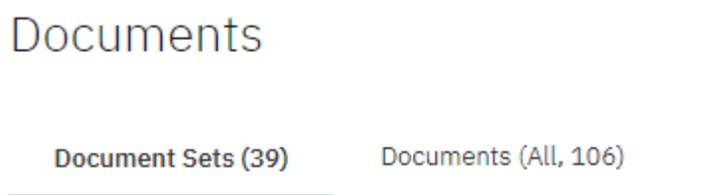
De manera general, se definió una relación por cada artículo del COIP Y LOT donde se enlazan las entidades verbos y palabras clave, pudiendo así determinar el artículo en cuestión, adicionalmente fue creada una relación denominada finalidad_190 que enlaza las palabras clave y la finalidad de este artículo.

- **Documentos:** es fundamental para el funcionamiento del modelo, la creación de un corpus donde se encuentren diversos ejemplos y casos que determinen y sean

representativos del contenido, por lo tanto, es necesario definir (en este caso, conductas tipificadas) y diversos ejemplos para cada uno de los artículos del COIP y de la LOT. Se generaron 39 conjuntos de documentos con 106 archivos de datos.

Figura 33

Documentos usados en Knowledge Studio



- **Diccionarios:** ya que el modelo generado por este servicio es del tipo supervisado, es posible anotar texto, esto se puede realizar de manera automática a través del uso de diccionarios, los cuales pueden ser generados de forma manual o a través de archivos de texto, a través de los mismos se generarán etiquetas personalizadas y de interés dentro de los textos cargados, realizando más sencilla las tareas de anotación. Fueron generados diccionarios para cada artículo buscando anotar los verbos y las palabras clave de cada uno y adicionalmente se creó un diccionario con alrededor de 200 ciudades del Ecuador.
- **Anotación:** Es posible generar diversos grupos de trabajo con el fin de asignar tareas de anotación, estableciendo reglas para que sean seguidas por todo el equipo, como fue descrito previamente esto es posible realizarlo de manera automática con el uso de

diccionarios o de forma manual a través del uso del editor de datos de la plataforma, básicamente se identifican entidades y relaciones de interés dentro del contenido, las cuales son marcadas con etiquetas.

- **Creación y entrenamiento del modelo:** Una vez finalizada la anotación de cada uno de los documentos con las entidades y relaciones de interés, se puede crear un modelo de machine learning definiendo los conjuntos de documentos que se desea usar para su entrenamiento y adicionalmente el porcentaje de estos documentos que se desea tomar como datos de entrenamiento, esto para evaluar diferentes estadísticas del modelo.
- **Publicar el modelo:** Una vez finalizado el entrenamiento del modelo y generadas las estadísticas del mismo es posible exportar y desplegar el mismo hacia otras aplicaciones de Watson. En este caso el modelo será desplegado con el servicio Watson Natural Language Understanding que permite usar PLN, de esta forma es posible analizar diversas características de una entrada de texto o documento, en este caso se usará para determinar las relaciones que puedan existir en estas entradas y por lo tanto reconocer si una conducta tipificada ingresada tiene relación con uno de los artículos del COIP o LOT. Ofreciendo como resultado a la petición un porcentaje de confianza de la predicción realizada por el modelo.

Una vez entrenado el modelo, la plataforma brinda información acerca del rendimiento y desempeño del mismo:

Fueron creadas diversas versiones del modelo a través del tiempo, donde se puede observar que se han realizado varias iteraciones de entrenamiento, inicialmente se tuvo un modelo de bajo rendimiento, hasta llegar a la versión 4.2 donde la precisión y exhaustividad del modelo para determinar relaciones y entidades es cercana a 1.

Figura 34

Versiones del modelo generado a través del tiempo



Nota. Se presentan las estadísticas de cada una de las entidades.

Donde:

- **Precisión:** indica un porcentaje de etiquetas de los datos tomados como prueba a los cuales el modelo ha realizado una predicción correcta.
- **Exhaustividad:** indica un porcentaje de las etiquetas correctas en un texto o documento dado que el modelo es capaz de predecir.
- **Puntuación macro F1:** representa una media armónica de las métricas de precisión y exhaustividad que se han obtenido previamente.

Figura 35*Estadísticas por Entidad*

Entity Types	F1	Precision	Recall	% of Total Annotations	% of Corpus Density (by number of words)	% of Documents that Contain This Type
finalidad	1	1	1	7% (149/2090)	3% (149/4287)	20% (1/5)
localizacion	0.92	1	0.85	10% (208/2090)	5% (208/4287)	100% (5/5)
palabras_clave	0.99	0.98	0.99	57% (1214/2090)	28% (1214/4287)	100% (5/5)
verbos	1	1	1	25% (519/2090)	12% (519/4287)	100% (5/5)
Overall Statistics	0.97	0.99	0.95	100% (2090/2090)	49% (2090/4287)	100% (5/5)

Nota. Estos valores han sido obtenidos en base a la matriz de confusión del modelo, esta permite de manera sencilla verificar el funcionamiento de un modelo de machine learning, la misma es presentada a continuación.

Figura 36*Matriz de confusión de entidades*

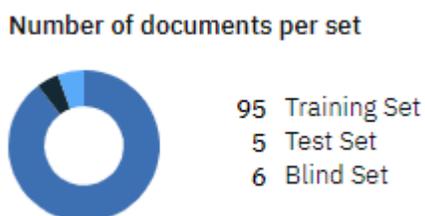
Test Set ▼ Confusion Matrix ▼ Entity Types Relation Types

Entity Types	O	finalidad	localizacion	palabras_clave	verbos	Total
O	2191	0	0	6	0	2197
finalidad	0	149	0	0	0	149
localizacion	30	0	178	0	0	208
palabras_clave	11	0	0	1203	0	1214
verbos	0	0	0	0	519	519
Total	2232	149	178	1209	519	4287

Es importante señalar que todas estas estadísticas están relacionados a los datos de entrenamiento usados para testear el modelo. Donde fueron usados 5 sets de documentos, así mismo 95 sets de documentos fueron usados para entrenar el modelo y adicionalmente 6 sets de documentos denominados ciegos, usados para probar periódicamente el modelo después de varias iteraciones, con la intención de mejorar el mismo.

Figura 37

Conjunto de documentos usados para entrenar y probar el modelo



Como fue descrito en el capítulo anterior el servicio de IBM Watson, Natural Language Understanding permite encontrar conceptos, emociones, entidades, palabras clave y diferentes relaciones dentro de un texto o documento, ya que se desea realizar esto desde otra aplicación es necesario definir un endpoint en una zona geográfica especificada y a través de este, se identificará una URL que creará una instancia del servicio basado en el modelo personalizado previamente creado.

Es posible consumir este endpoint a través de las siguientes herramientas y lenguajes de programación: Curl, Java, Node JS, Python, Go, .NET, Ruby, Swift y Unity. Se usará .NET debido a que es usado por varios módulos del proyecto, permitiendo así una fácil integración.

Se deberá crear un objeto del tipo `IamAuthenticator` donde se establecen las credenciales para el acceso al endpoint, en este caso se usará el `APIKEY` del servicio. A través del objeto del servicio de Natural Language Understanding del tipo `NaturalLanguageUnderstandingService` el cual solicita como parámetros la versión del servicio y las credenciales de autenticación previamente instanciadas. La propiedad `ServiceUrl` de este objeto permite asignar la URL del endpoint creado en la plataforma.

A través de la función `Analyze` del objeto del servicio, es posible establecer el código del modelo creado por Knowledge Studio y adicionalmente el query o texto que se desea analizar, logrando así completar la petición donde se concatena el modelo creado y el servicio de PLN de la plataforma, se retornará una respuesta del tipo JSON la cual será deserializada para encontrar las relaciones presentes en el texto, de existir.

Dentro del objeto de respuesta se encuentra el nombre de la relación y una puntuación de la confianza que tiene el modelo al realizar la predicción.

Amazon Web Services

La plataforma Amazon Web Services posee los siguientes módulos de manera general, los cuales permiten la creación de un modelo de machine learning basado en datos específicos

de un área particular de interés, creando así un modelo del tipo no supervisado, el mismo que se comporta como clasificador.

Figura 38

Módulos generales modelo de Amazon Comprehend



Nota. Representación de módulos generales de Amazon Comprehend, (AWS, 2018).

A través del uso del servicio Amazon S3 el cual permite almacenar todo tipo de información en la nube de AWS mediante el uso de una interfaz web, es posible insertar información en una instancia de este servicio, la información deberá ser del tipo CSV, mediante una colección de archivos será posible entrenar al modelo, estos archivos como fue mencionado previamente poseen la recopilación de los artículos de las leyes COIP y LOT, así como también noticias encontradas en medios digitales y finalmente varios ejemplos de casos extraídos de la página web de la Función Judicial del Ecuador.

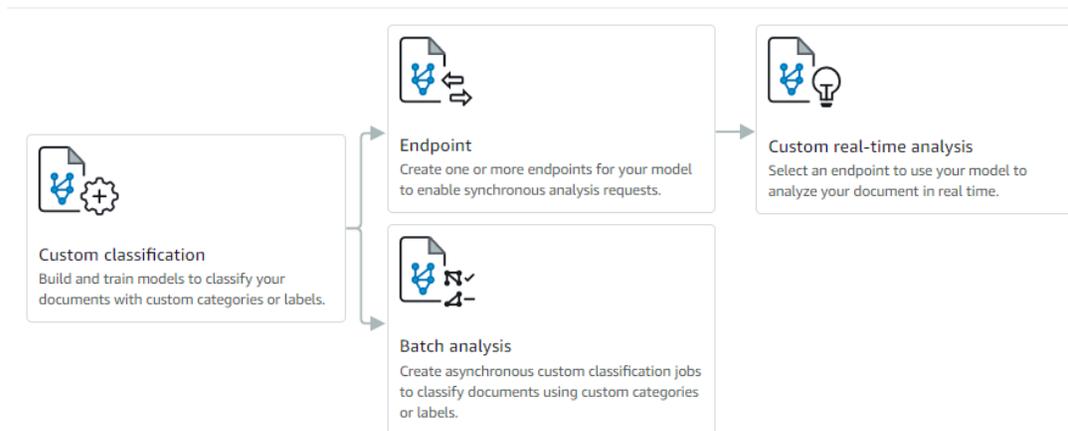
A continuación, esta colección será usada por el servicio de Amazon Comprehend el cual es capaz de procesar el texto usando NLP, de manera que se pueden extraer frases claves, entidades y sentimientos entre otras cosas, las cuales fueron detalladas en el capítulo 2. En este caso fue creado un modelo basado en la asignación latente de Dirichlet para determinar los artículos de las leyes COIP y LOT que pudieran estar relacionados con una conducta tipificada. Esto se realiza examinando la conducta tipificada de esta manera es posible determinar el contexto y el significado de cada una de las palabras, con frecuencia es posible encontrar algunas palabras relacionadas en un contexto específico, por lo cual se puede determinar un artículo específico, cada palabra aporta un porcentaje que ayuda a definir este artículo.

El modelo clasificador de Amazon Comprehend creado, utiliza un modelo de red neuronal profunda de etiquetado de secuencias de vanguardia propio que alimenta el mismo servicio de entidades de detección de Amazon Comprehend para entrenar modelos del tipo reconocedor de entidades personalizadas, así como también clasificadores personalizados. (AWS, 2021)

La clasificación personalizada de Amazon Comprehend permite crear modelos propios mediante dos sencillos pasos, el primer paso es entrenar el clasificador de manera que se reconozcan las clases (en este caso, los artículos de las leyes COIP y LOT) que son de utilidad y beneficio, después deberán cargarse documentos sin ningún tipo de etiqueta para comprobar la funcionalidad del modelo en su clasificación. El esquema general de una clasificación personalizada es el siguiente:

Figura 39

Clasificación personalizada de Amazon Comprehend



Nota. Se presentan los diferentes módulos de la clasificación personalizada de Amazon Comprehend, (AWS, 2021).

- **Clasificación personalizada:** A través del uso de la clasificación personalizada opción provista por la plataforma, es posible construir y entrenar modelos que clasificarán documentos (en este caso, conductas tipificadas) con categorías o etiquetas personalizadas y de interés.
- **Endpoint:** Es posible crear uno o más endpoints para el modelo de clasificación creado de manera que se puedan clasificar los documentos y se puedan también permitir diversas solicitudes de análisis síncrono.
- **Análisis de Lotes:** Es posible crear trabajos asincrónicos de clasificación personalizada de documentos usando categorías o etiquetas personalizadas y de interés.
- **Análisis en tiempo real:** Es posible seleccionar un endpoint previamente creado para usar el modelo de clasificación y analizar todo tipo de documentos de manera real.

Figura 40*Amazon Comprehend análisis en tiempo real***Real-time analysis with built-in models**

Use Amazon Comprehend built-in models to analyze content in real time.

**Custom real-time analysis with endpoint**

Create an endpoint for your custom model to [classify documents](#) or [detect custom entities](#) in real time.

Nota. Se presentan los diferentes módulos de la clasificación personalizada de Amazon Comprehend, (AWS, 2021).

Mediante la plataforma de Amazon Comprehend es posible realizar análisis en tiempo real por medio de dos opciones:

- **Análisis mediante modelos incorporados:** esta opción permite usar los modelos que se tiene por defecto en la plataforma, para analizar un contenido en tiempo real.
- **Análisis mediante endpoints:** esta opción permite crear un endpoint el cual puede ser consumido a través de ciertos lenguajes de programación, el cual será un modelo personalizado del tipo clasificador (este caso) o reconocedor de entidades en tiempo real.

Es importante señalar que el API de AWS puede ser utilizado con uno de los siguientes lenguajes de programación: .NET, C++, Go, Java 2, JavaScript, PHP V3, Python y Ruby V3.

Para este caso y ya que varios módulos del sistema utilizan el framework de .NET, se usó el Software Developer Kit (SDK) de .NET una vez que el modelo ha sido entrenado definiendo el bucket de Amazon S3 donde se encuentran disponibles los archivos del tipo CSV, con la información de las etiquetas o clases que se desean clasificar. La plataforma brinda ciertos detalles entre los cuales se encuentra:

- **Número de etiquetas:** es el número de etiquetas o clases que se desean determinar a partir del modelo clasificador.
- **Número de documentos entrenados:** el número de documentos con los cuales se ha entrenado al modelo clasificador.
- **Número de documentos testeados:** el número de documentos con los cuales se ha probado el modelo clasificador.
- **Lenguaje:** el lenguaje que usa el modelo clasificador.
- **ARN:** el nombre de los recursos de Amazon es un identificador único el cual establece de manera exclusiva cada uno de los recursos que se proveen en la plataforma.
- **Ubicación de los datos de entrada:** es la referencia hacia el bucket de Amazon S3 donde se encuentran los documentos usados para entrenamiento y testeo del modelo clasificador.

Figura 41

Detalles del modelo clasificador de Amazon Comprehend

Classifier details			
Name	ARN	Number of labels	Input data location
ArticlesClassifier	arn:aws:comprehend:us-east-2:29139251166:document-classifier/ArticlesClassifier 	16	s3://dojo-text-records 
Status	Training started	Number of trained documents	Language
 Trained	6/5/2021 22:16:31	2513	Spanish
Mode	Training ended	Number of test documents	Classifier encryption
Multi-class	6/5/2021 22:46:49	279	-

Una vez entrenado el clasificador, la plataforma brinda información acerca del rendimiento y desempeño del mismo, los detalles que presentan son:

- **Precisión:** indica un porcentaje de etiquetas o clases de los datos tomados como prueba a los cuales el modelo ha realizado una clasificación de manera correcta.
- **Macro precisión:** representa una medida del beneficio y de la utilidad que poseen los resultados brindados por el clasificador frente a los datos que han sido tomados como pruebas. Donde se ha tomado el número total de documentos y textos clasificados de manera correcta y se los divide entre el número total de clasificadores de la clase.
- **Macro exhaustividad:** indica un porcentaje de las categorías o etiquetas correctas en un texto o documento dado que el modelo es capaz de predecir, todo esto basado en los datos tomados como pruebas. Donde se ha tomado la media de todas las puntuaciones de exhaustividad para cada etiqueta disponible.
- **Puntuación macro F1:** representa una media armónica de las métricas de precisión y exhaustividad que se han obtenido previamente.

- **Pérdidas de ataques hash:** represente una parte de las etiquetas o clases del clasificador que se han predicho de manera errónea.
- **Micro precisión:** similar a la macro precisión, se diferencian en que se basa en la puntuación global de cada una de las puntuaciones de precisión agregadas.
- **Micro exhaustividad:** similar a la macro exhaustividad, su diferencia es, en lugar de realizar el promedio de las puntuaciones de todas las etiquetas, se toman en cuenta las puntuaciones de exhaustividad agregadas.
- **Puntuación micro F1:** representa una media armónica de las métricas de micro precisión y de micro exhaustividad que se han obtenido previamente.

Figura 42

Rendimiento del clasificador de Amazon Comprehend

Classifier performance Info			
Accuracy	Precision	Recall	F1 score
0.9964	0.9833	0.9778	0.97
Hamming loss	Micro precision	Micro recall	Micro F1 score
0.0036	0.9964	0.9964	0.9964

De forma general obtener métricas cercanas a 1, como es el caso del clasificador entrenado significa que el mismo será eficaz en cada caso de uso, sin embargo, es necesario tener en cuenta que los datos de prueba usados para conseguir las métricas de rendimiento del modelo, fueron obtenidos en su mayoría por datos usados en el entrenamiento del modelo, de

manera que el mismo podrá tener puntos de fallo que serán analizados a mayor profundidad en el siguiente capítulo.

Posterior a la creación y entrenamiento del modelo clasificador, es posible generar un Endpoint de este clasificador, el cual será la puerta de entrada hacia la conexión del modelo con el mundo exterior, de manera que pueda ser consumido mediante el uso de cualquier lenguaje de programación. A través de este endpoint la plataforma realiza el cargo dependiendo de la cantidad de unidades de inferencia con las que sea aprovisionado, cada unidad representa un rendimiento de 100 caracteres.

Figura 43

Detalles del endpoint del modelo clasificador de Amazon Comprehend

Endpoint details		
Name	ARN	Creation time
articlesEndPoint	arn:aws:comprehend:us-east-2:229139251166:document-classifier-endpoint/articlesEndPoint	24/5/2021 21:22:49
Status	Inference units Pricing information	Last modified time
🟢 Ready	1	24/5/2021 21:22:49

Al igual que en el caso del clasificador, la plataforma brinda ciertos detalles acerca del endpoint de los cuales se puede resaltar las unidades de inferencia seleccionadas, y el ARN que será importante para consumir el modelo a través de un lenguaje de programación.

Para consumir el endpoint del modelo clasificador creado mediante Amazon Comprehend a través de .NET es necesario crear un objeto del tipo `Amazon.Runtime.BasicAWSCredentials` el cual posee la información del `accessKey` y `secretKey` que son datos básicos de autenticación para el uso de los servicios de la plataforma.

Se usará un objeto del tipo `Amazon.Comprehend.AmazonComprehendClient` el cual solicita como parámetros el objeto de credenciales previamente instanciado y además un objeto que indique la región donde se encuentra alojada el endpoint, en este caso `Amazon.RegionEndpoint.USEast2`

Se usará un objeto del tipo `Amazon.Comprehend.Model.ClassifyDocumentRequest` el cual tendrá como parámetros el ARN del endpoint y el texto que se desea ingresar al clasificador, el cual realiza una petición al servidor mediante el uso del objeto cliente previamente instanciado.

A través de una respuesta asincrónica, es posible obtener la respuesta del endpoint, gracias al método `ClassifyDocumentAsync` al cual tendrá como parámetro la petición previamente instanciada, y los métodos `GetAwaiter().GetResult()` de esta forma se obtendrá un objeto de respuesta del tipo `Amazon.Comprehend.Model.ClassifyDocumentResponse`, dentro del cual se tiene una lista de objetos del tipo `Amazon.Comprehend.Model.DocumentClass` los cuales tendrán información acerca de las etiquetas o clases de respuesta de nuestro modelo (en este caso, los artículos de las leyes COIP y LOT), cada uno de estos objetos ofrece la posibilidad de encontrar el nombre de la etiqueta así como también un valor del tipo puntuación que simboliza la medida de confianza que tiene el modelo sobre la clasificación realizada.

Microsoft Azure

El servicio web que ofrece el procesamiento de lenguaje natural en la plataforma de Microsoft Azure, es Language Understanding (LUIS). Al igual que el servicio de NLP de IBM Watson, LUIS permite la creación de un modelo de aprendizaje automático supervisado para la extracción de información de interés y tiene la capacidad de identificar lo que quiere el usuario o el tema relacionado del que habla.

Para crear y entrenar el modelo de aprendizaje automático supervisado de LUIS se siguió el ciclo de iteración indicado a continuación, aplicado a la información recopilada que se utilizó para entrenar los modelos en IBM Watson y AWS.

Figura 44

Ciclo de Iteración para el diseño de aplicaciones con LUIS



Nota. Se presenta el ciclo iterativo para el diseño y creación de aplicaciones de NLP con LUIS (Microsoft Azure, 2016).

Crear el esquema de aplicación o modelo de LUIS

El esquema de la aplicación debe de ser específico del dominio de aplicación para poder extraer información e identificar la intención del usuario, para el presente caso de estudio el dominio está conformado por el CORPUS descrito anteriormente. Además, durante este paso se identifica que pedirá el usuario o el tema relacionado del texto que se desea analizar (intención), en este caso, cada intención creada está relacionada estrechamente con cada uno de los once artículos del Código Integral Penal (COIP) y los cuatro Ley Orgánica de Telecomunicaciones (LOT).

Posteriormente se crean las entidades que aportarán datos y ayudarán al modelo de LUIS para identificar cada intención. Para el presente caso se creó una entidad del tipo Machine Learned (ML) por cada artículo del COIP y de la LOT, estas entidades a su vez están compuesta por dos subentidades más pequeñas denominadas verbos y palabras clave que ayudarán a identificar delitos relacionados con los artículos del COIP y de la LOT.

Añadir ejemplos de entrenamiento

Las expresiones de ejemplo son necesarias para entrenar el modelo en LUIS y deben abarcar todas las intenciones y entidades creadas en el paso anterior. Adicionalmente, las expresiones de ejemplo deben ser lo más parecidas posibles a las entradas que realizaría el usuario a la aplicación en tiempo de ejecución.

Debido a que en LUIS el modelo de aprendizaje automático es del tipo supervisado, es necesario realizar anotaciones, sobre las expresiones de ejemplo ingresadas, que estén relacionadas con las entidades creadas para identificar cada intención y así facilitar el reconocimiento de LUIS.

A continuación, se puede observar una expresión de ejemplo por la intención relacionada al artículo 194 utilizada durante el entrenamiento y las anotaciones realizadas sobre la misma. Se puede observar que la entidad ART194 se encuentra compuesta de dos subentidades: la primera subentidad hace referencia a la acciones o verbos definidos en cada artículo de la LOT o el COIP, mientras que la segunda entidad hace referencia a las palabras clave que complementan el sentido de cada artículo.

Figura 45

Expresión de ejemplo y anotación de entidades en LUIS

carlos castillo vendia de manera ilegal dispositivos moviles en la ciudad de santo
Verbos Palabras Clave
ART194
domingo .

Entrenamiento y publicación del Modelo de LUIS

El entrenamiento consiste en enseñar al modelo creado de LUIS a reconocer las intenciones y entidades creadas para su extracción e identificación. El entrenamiento se debe ejecutar cuando se han realizado modificaciones en el modelo, es decir, se añadieron, editaron o eliminaron entidades, intenciones o expresiones de ejemplo.

Para el entrenamiento del modelo es importante tener en consideración algunas recomendaciones de Microsoft Azure:

- Se debe tener al menos 30 expresiones de ejemplo por cada intención.
- Se debe etiquetar correctamente las entidades en las expresiones de ejemplo de cada intención para evitar futuros conflictos.
- Las expresiones de ejemplo deben de ser variadas para aumentar la eficacia del modelo.
- Se puede añadir expresiones que no estén relacionadas con el dominio de la aplicación y asignarlas a la intención None.
- El entrenamiento puede utilizar una pequeña parte de las expresiones de ejemplo o todas las expresiones. Para utilizar todas las expresiones se debe desactivar la opción no determinística.
- Se puede activar una opción que permite que las tildes no afecten el puntaje de predicción.
- El entrenamiento es un proceso iterativo, una vez entrenado se debe probar el modelo con expresiones de ejemplo para validar que las entidades e intenciones sean reconocidas de forma correcta, caso contrario, hacer las modificaciones necesarias en los ejemplos que generen conflictos en LUIS, volver a entrenar y probar si se han resuelto los conflictos.
- Se puede publicar el modelo en dos ranuras, ensayo y/o producción, obteniendo dos versiones del modelo.

En base a estas recomendaciones se entrenó al modelo creado en LUIS de forma iterativa, con la información recopilada en el CORPUS descrito anteriormente. Se utilizó

alrededor de un total de 1500 expresiones de ejemplo, aproximadamente 100 expresiones de ejemplo por cada intención creada.

La publicación consiste en que el servicio sea accesible a través del internet mediante un endpoint. Al finalizar el entrenamiento y tener pruebas satisfactorias del reconocimiento de intenciones del modelo, se procede a la publicación del Modelo en LUIS a través de un endpoint. En LUIS existen dos ranuras para realizar la publicación:

- **Ranura de ensayo:** es útil para un ambiente de pruebas antes de ser implementado en producción.
- **Ranura de Producción:** es el área de producción donde se encuentra la versión operativa de la aplicación.

Para la publicación del modelo se utilizó la ranura de producción y se obtuvo un endpoint para utilizar el modelo a través de internet: tesis-nlp.cognitiveservices.azure.com

Endpoints HTTPs

Con el endpoint obtenido se puede realizar diversas peticiones HTTP dependiendo de lo que se desea realizar. Es posible realizar desde una petición HTTP POST para crear un modelo, una petición HTTP DELETE para eliminar el modelo, una petición HTTP GET para consultar información del modelo, entre otras.

La petición HTTP que nos interesa para realizar una consulta al modelo que se creó es la del tipo HTTP GET, específicamente la "Predict - Get published slot predictions". El URL para realizar este tipo de peticiones es de la forma:

```
https://{endpoint}/luis/prediction/v3.0/apps/{appId}/slots/{slotName}
/predict?subscription-key={subscription-
key}&query={query} [&verbose] [&log] [&show-all-intents]
```

Donde los parámetros para realizar la consulta son:

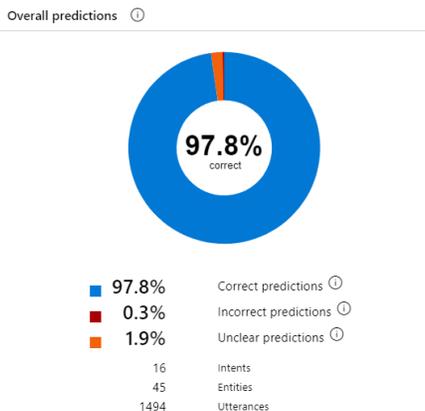
- **endpoint:** Punto de Conexión del modelo
- **appId:** ID de la aplicación o modelo
- **slotName:** La ranura que se utilizará en la predicción (production/ staging)
- **subscription-key:** Clave primaria para utilizar el servicio
- **query:** Consulta a realizar (hasta 500 caracteres)

Métricas del Modelo

Una vez entrenado el modelo creado en LUIS se puede visualizar las métricas del modelo de predicción. Estas métricas se pueden obtener en la pestaña DASHBOARD del panel de gestión de LUIS.

Figura 46

Métricas Modelo en Microsoft Azure

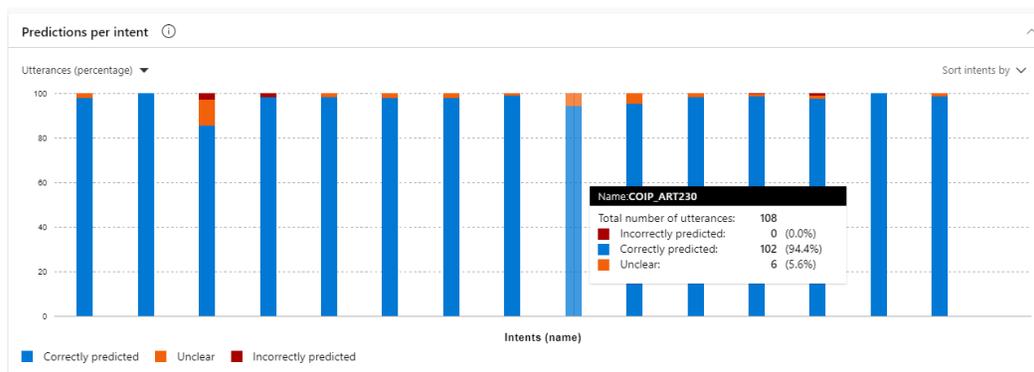


- **Predicciones Correctas:** La intención predicha coincide con la intención actual de la expresión de ejemplo.
- **Predicciones Incorrectas:** La intención predicha no coincide con la intención actual de la expresión de ejemplo.
- **Predicciones Poco claras:** Las dos puntuaciones más altas predichas para una expresión de ejemplo tienen puntuaciones muy cercanas.

Adicionalmente también se puede obtener los resultados de forma individual de cada intención.

Figura 47

Predicciones por cada artículo en Microsoft Azure



Es posible consumir este endpoint a través de las siguientes herramientas y lenguajes de programación: Curl, Java, Node JS, Python, Go, .NET, Ruby, Swift y Unity. De la misma forma que en los dos casos anteriores se usará .NET para consumir este endpoint.

Al contrario de los modelos previos, este servicio permite realizar la autenticación con un query parameter asignado al parámetro `subscription-key` de esta forma se puede realizar la consulta de forma sencilla, será necesario un objeto del tipo `WebClient`

Adicionalmente será usado el parámetro `query` para establecer el texto que se desea analizar, la respuesta de esta petición será del tipo JSON y es obtenida a través del método `DownloadString`, del objeto `web client` creado, al cual se establecerá la URL con los parámetros previamente señalados. Una vez deserializada la respuesta es posible encontrar los datos de la predicción, con las etiquetas definidas las cuales señalarán un artículo y una medida de confianza de esta predicción.

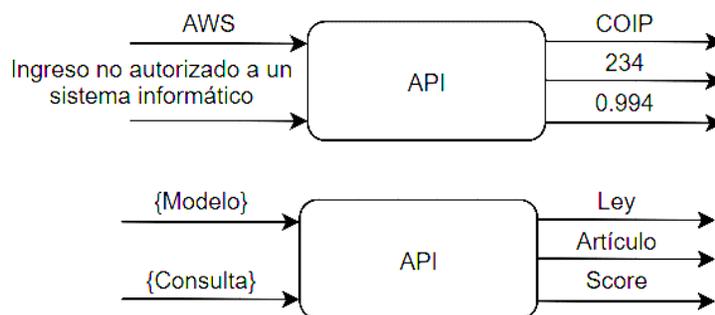
API

De la misma forma que fue creada el servicio web para el uso y manejo de la base de datos, se creó un servicio web del tipo RESTful basado en el framework de .NET Web API y el

lenguaje de programación VB para el manejo de las peticiones hacia cada uno de los modelos entrenados en las diferentes plataformas (AWS, Azure y IBM Watson). Con la creación de estos servicios web se simplificó y normalizó los parámetros de entrada y salida al consumirlos los servicios a través de un query string con el método GET, diferenciado las consultas hacia cada modelo simplemente por su ruta y obteniendo como respuesta un objeto JSON normalizado que contiene la ley, el artículo y un score de exactitud relacionados con el texto ingresado como parámetro de la consulta a través del query string, como se indica a continuación.

Figura 48

API creado para consultar los modelos las diferentes plataformas



La ruta creada de manera general es la siguiente:

`Servicios/Pruebas/api/{modelo}?query={texto}`

Donde modelo tiene las siguientes posibilidades:

- azure para el modelo de Microsoft Azure
- aws para el modelo de Amazon Web Services
- watson para el modelo de IBM Watson

Donde texto se refiere a la conducta tipificada de la cual se desea conocer el artículo de las diferentes leyes que puedan estar relacionados con el mismo, es importante señalar que de ser incorrecta la consulta y que el modelo seleccionado no pueda relacionar el texto con algún artículo, se retornará un objeto JSON con la respuesta "None" tanto para ley como para artículo y el score será del 100% esto para diferenciar los resultados que han podido ser predichos de los cuales no.

Para efectos del API se crearon tres controladores uno para cada modelo:

- AwsController
- AzureController
- WatsonController

Dentro de los cuales básicamente han sido consumidos los modelos entrenados de acuerdo a las especificaciones descritas en las secciones anteriores para cada plataforma, como detalle adicional se creó el objeto JSON de respuesta el cual tiene la misma forma para todos los casos.

```
New JObject(New JProperty("ley", ley), New JProperty("articulo",  
articulo), New JProperty("score", score))
```

Donde ley, articulo y score se refieren a las variables obtenida de la respuesta del modelo seleccionado.

El método GET del endpoint tiene la misma forma para cada uno de los modelos, solamente se diferencian en la función que obtiene la respuesta la cual se refiere al modelo en cuestión pudiendo ser: `GetAwsResponse`, `GetAzureResponse`, `GetWatsonResponse`

```
Public Function GetValue(ByVal query As String) As JsonResult

    Dim response = GetAwsResponse(query)

    Dim jsonObject As New JsonResult()

    jsonObject.JsonRequestBehavior =
JsonRequestBehavior.AllowGet

    jsonObject.Data = response

    Return jsonObject

End Function
```

Donde se declara como respuesta el tipo de objeto `JsonResult` de la clase `System.Web.Mvc.JsonResult` y el resultado de cada una de las funciones se asigna al parámetro `data` de este objeto, así mismo es necesario configurar el API para que retorne en sus respuestas objetos JSON ya que por defecto retorna objetos del tipo XML.

Asistente de Texto

El asistente de texto es un poderoso asistente virtual el cual posee inteligencia artificial y proporciona respuestas rápidas, consistentes y precisas a través de cualquier canal, dispositivo

o aplicación, se encuentra disponible a través del servicio de IBM Watson denominado Watson Assistant, el cual permite crear cualquier tipo de flujo conversacional y en este caso, permitirá a los usuarios interactuar para obtener una respuesta frente a una conducta tipificada ingresada en el mismo, a continuación se revisarán conceptos y procedimientos seguidos para obtener un chatbot donde se reúnan todos los APIs y módulos previamente detallados.

Figura 49

Diagrama de Watson Assistant



Nota. Se presenta el diagrama general del servicio Watson Assistant, (IBM Cloud , 2020).

El diagrama muestra el flujo de una conversación entre el cliente y el asistente, existen los siguientes módulos:

- **Canales:** son los medios a través de los cuales el cliente realiza la interacción, entre los cuales se tiene el chat web incorporado mediante el uso de JavaScript a cualquier sitio web y una aplicación personalizada del tipo móvil.
- **Asistente:** el encargado en comunicar el cliente y sus requerimientos con los diálogos previamente establecidos.

- **Diálogos:** a través del uso de intenciones, entidades y el ingreso de información por parte del cliente y webhooks es posible redireccionar y dirigir de manera adecuada el flujo de la plática establecida entre cliente y asistente.
- **Búsqueda:** si previamente una interacción del cliente no ha sido respondida, se inicia una búsqueda la misma que intenta detecta las respuestas fundamentado en una base de conocimientos predefinidos.
- **Agente:** es posible realizar interacciones e integraciones con agentes en tiempo real vía telefónica, brindado el asistente esta opción de ser configurada.

Es importante señalar que los dos últimos módulos han sido excluidos ya que no forman parte de los objetivos planteados del presente proyecto. A continuación, se especifican todos los detalles técnicos del funcionamiento del asistente de texto que permiten su funcionamiento.

Intenciones

Las intenciones se refieren a todos los motivos, propósitos o un fin en particular que se ingresa a través de la entrada del cliente, entre las principales intenciones se encuentran el responder a una pregunta, la misma que da lugar a cierto flujo definido a través del uso de diálogos, una vez que la respuesta ingresada por el usuario ha sido reconocida.

Para este proyecto han sido creadas siete intenciones, las cuales se detallan con su motivo a continuación:

- **Acerca De:** información acerca del proyecto.
- **Buscar Delito:** relacionar una conducta tipificada con un delito de las telecomunicaciones.

- **Condena:** información relacionada a la condena de un artículo previamente seleccionado.
- **Consulta Artículo:** buscar un artículo relacionado a las telecomunicaciones del COIP y la LOT.
- **Detalle:** información relacionada al detalle del artículo previamente seleccionado.
- **Gracias:** reconocer la voluntad del usuario de no saber más sobre el artículo previamente solucionado.
- **Ley:** información relacionada a la ley proveniente del artículo previamente seleccionado.

Entidades

Las entidades al igual que las intenciones se refieren a todo tipo de información importante y relevante para el usuario, las cuales constituyen valores requeridos para los propósitos de ciertos flujos del asistente. Las entidades pueden ser verbos, sustantivos entre otros, reconocer esta información a través de la entrada de usuario es necesario ya que se pueden crear respuestas más útiles y direccionadas.

Para este proyecto han sido creadas dos entidades y adicionalmente se ha utilizado una entidad que tiene el asistente por defecto, las cuales se detallan con su motivo a continuación:

- **Modelo:** modelo de machine learning que se desea consultar, las opciones son AWS, Azure, Watson.
- **Texto:** reconocer la conducta tipificada ingresada por el usuario, cuando se encuentra en el diálogo Buscar Delito.

- **Sys-number:** extrae números expresados como dígitos o escritos como números, necesario para consultar el artículo de telecomunicaciones que el usuario requiera.

Webhook

Un webhook es un mecanismo por medio del cual se permite realizar una llamada hacia un programa externo basado en ciertas condiciones y flujos conversacionales que ocurren mientras se usa el servicio del asistente de Watson, es usado como una característica especial que poseen todos los diálogos, solamente debe habilitarse para iniciar su uso. A través del uso de las variables de contexto es posible recolectar datos específicos por parte del usuario, el webhook usa una petición del tipo HTTP POST hacia una URL determinada en la configuración del mismo y usa las variables de contexto con la información proporcionada.

La URL que recibe la información actúa como una acción de oyente, es decir estará a la espera de ser llamado y realizará cierta acción específica con la información suministrada, luego, se realiza un procesamiento de datos y finalmente se retorna una respuesta.

Mediante el uso del webhook es posible realizar las siguientes acciones de acuerdo a la documentación oficial (IBM Cloud , 2020):

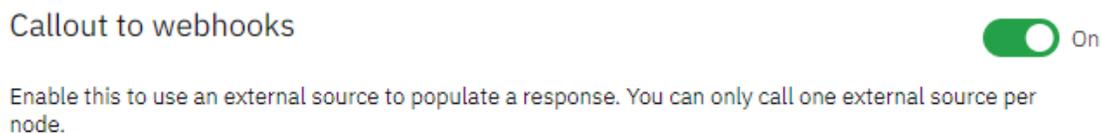
- Validar información del usuario, la cual ha sido recolectada y requerida.
- Interactuar con un servicio web externo para obtener determinada información.
- Enviar peticiones hacia una aplicación externa de manera que se completen los flujos y las operaciones que el usuario ha solicitado y realizado.
- Accionar una notificación del tipo Short Message Service. (SMS)
- Desencadenar una acción web de las funciones de IBM Watson.

Para el proyecto ha sido usada la opción de usar una acción web de IBM Watson, como fue mencionado previamente es posible definir la URL del webhook y esta ser utilizada por uno o varios diálogos.

Es necesario habilitar la opción en el diálogo para poder hacer uso de los webhooks.

Figura 50

Habilitar uso de Webhooks



Luego es necesario configurar el mismo, estableciendo la URL y es posible adicionar contenido al encabezado de la petición HTTP POST de manera que se cree una capa de seguridad debido a la autenticación, sin embargo, no es un requisito, por lo tanto, no se ha utilizado esta funcionalidad. Así, se ha ingresado la URL de la acción web creada.

Figura 51

Configuración Webhook

Webhooks

A webhook is a mechanism that allows you to call out to an external program based on events in your dialog.

Webhook setup

Specify the request URL for an external API you want to be able to invoke from dialog nodes. Watson will call this URL when configured to do so from a dialog node. [Learn more](#)

URL

```
https://us-south.functions.appdomain.cloud/api/v1/web/tesisbyrdav%40g
```

Headers

Add HTTP headers for authorization or any other parameters required for invoking the webhook.

Header name	Header value
-------------	--------------

[Add header](#) [Add authorization](#)

La acción web es un servicio de IBM Watson el cual permite invocar una pequeña cantidad de código o establecer una respuesta automática la cual es generada justo después de un evento.

Se ha creado una URL que puede ser usada desde cualquier aplicación web, la cual es generada de manera asincrónica, para este caso se ha escrito un pequeño script usando el lenguaje de programación JavaScript y el tipo de entorno ha sido Node JS, el nombre asignado al script es “consulta_art”.

Este script permite concatenar la funcionalidad de consultar artículos de las diferentes leyes, como también buscar delitos dada una conducta tipificada, esta distinción se realiza a través de un parámetro llamado tipo, el cual tendrá dos valores, los cuales son:

- consulta_art: usado para consultar artículo.
- consulta_texto: usado para determinar artículo.

Se ha usado la librería Request-Promise la cual permite realizar una petición del tipo HTTP hacia cualquier cliente, y retorna una promesa como respuesta. Se han establecido los servicios web creados previamente para cada caso de la variable tipo, como opción la librería permite retornar un objeto JSON, dentro de la promesa, por lo cual a la salida de esta acción se tendrá siempre un JSON, aunque el modelo o artículo seleccionado no retornen datos, se ha establecido un objeto vacío para este caso especial.

Par el caso que la variable tipo sea:

- consulta_art: es necesario proporcionar un artículo en forma numérica.
- consulta_texto: es necesario proporcionar uno de los modelos (aws, azure, watson) y una conducta tipificada la cual será enviada a través del query string de la petición, e identificada a través de la entidad texto.

Esta opción ha sido usada en los diálogos: Buscar Delito y Consulta Artículo que se detallan a continuación.

Diálogos

Como fue mencionado previamente a través del uso del diálogo es posible definir un flujo establecido en una conversación por parte del asistente, el mismo reconoce y usa cada una de las intenciones identificadas desde la entrada por parte del usuario, reconociendo y enmarcando cada uno de sus contextos de forma que se cree una interacción bilateral siendo el objetivo brindar una respuesta de utilidad para el usuario.

En el presente proyecto se han creado nueve diálogos, los cuales se detallan a continuación con cada uno de sus flujos, de manera que sea posible entender el funcionamiento del asistente en su totalidad.

Bienvenido

Diálogo de bienvenida que permite saludar al usuario y dar una breve introducción sobre el asistente.

Figura 52

Diálogo Bienvenido

Hola bienvenido, soy tu asistente, te ayudare a encontrar un artículo del Código Integral Penal (COIP) y la Ley Orgánica de Telecomunicaciones (LOT) relacionado con las telecomunicaciones. También puedo encontrar la relación de un delito con un artículo de estas leyes.

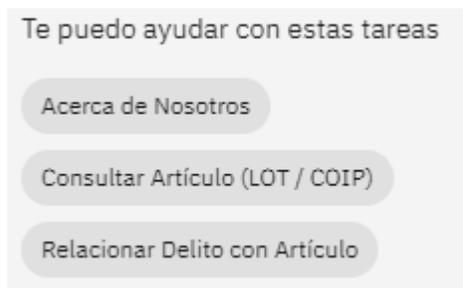
Opciones

Ofrece una lista de tres etiquetas las cuales se comportan como botones y son:

- Acerca de Nosotros: inicia el flujo conversacional del diálogo “Acerca de Nosotros”
- Consultar Artículo: inicia el flujo conversacional del diálogo “Consulta Artículo”
- Relacionar Delito con Artículo: inicia el flujo conversacional del diálogo: “Buscar Delito”

Figura 53

Etiquetas del diálogo Opciones



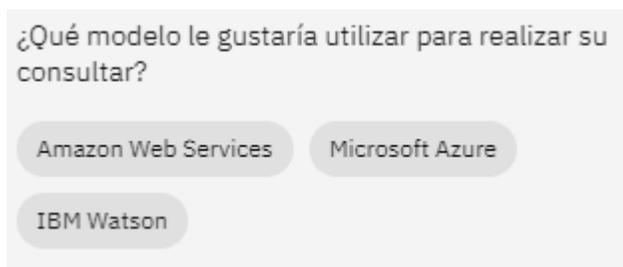
Buscar Delito

Permite relacionar una conducta tipificada ingresada por el usuario, con un artículo relacionado a las telecomunicaciones tanto del COIP o la LOT, para lo cual es necesario que se elija a través del uso de etiquetas entre:

- **Amazon Web Services:** elegirá el modelo de Amazon Comprehend para la consulta.
- **Microsoft Azure:** elegirá el modelo de LUIS para la consulta.
- **IBM Watson:** elegirá el modelo de Natural Language Understanding para la consulta.

Figura 54

Etiquetas del diálogo Buscar Delito



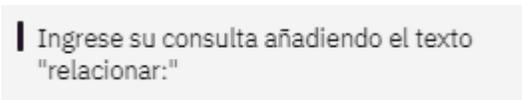
Además, es necesario que se ingrese un texto de consulta, el cual es entendido a través de la entidad “texto”, para que el asistente reconozca el texto es necesario ingresarlo de la siguiente manera:

relacionar: {texto a ingresar}

El asistente al solicitar el texto de consulta, informa al usuario del ingreso adicional de “relacionar:”

Figura 55

Petición del texto en diálogo Buscar Delito



La forma de ingresar el texto a consultar tiene la siguiente estructura:

Sujeto + verbo + palabras clave + locación

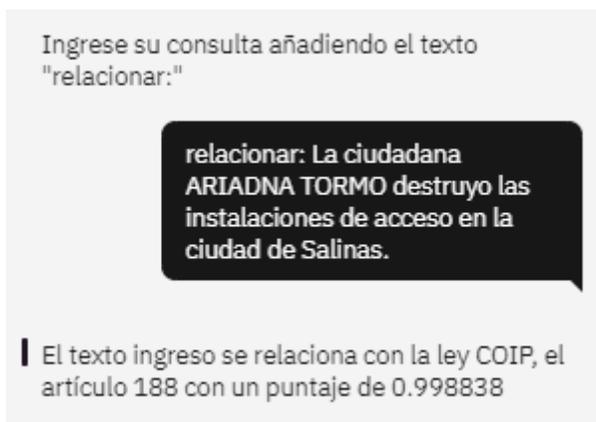
Donde:

- Sujeto puede referirse a un nombre propio, o a una palabra de descripción (señor, señora, joven, etc.)
- Verbo puede estar conjugado en cualquier tiempo.
- Palabras clave se refiere a un identificador del delito (instalaciones de telecomunicaciones, dispositivos móviles, sistemas telemáticos, equipos terminales, servicio de radiodifusión, etc.)
- Locación se refiere a un espacio físico (calle, radio base, edificio) o a una ciudad del Ecuador

Por ejemplo:

Figura 56

Ejemplo de ingreso de texto en diálogo Buscar Delito



Con esta información establecida se realiza una consulta usando el webhook detallado en la sección anterior, el cual tendrá como contenido del parámetro tipo la frase “consulta_texto”, consultando así el API de modelos.

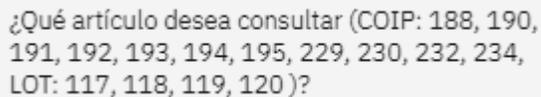
Una vez que el artículo ha sido identificado es posible iniciar el diálogo “Consulta Artículo After” o retornar a “Opciones”.

Consulta Artículo

Permite consultar un artículo relacionado con las telecomunicaciones tanto del COIP como la LOT, para lo cual es necesario ingresar un número de artículo, el cual será entendido a través del uso de la entidad sys-number, esta información será enviada al webhook, el cual tendrá como contenido del parámetro tipo la frase “consulta_art”, consultando así el API de artículos.

Figura 57

Diálogo Consulta Artículo



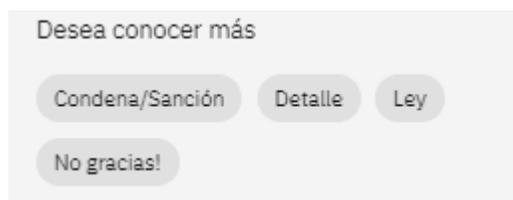
¿Qué artículo desea consultar (COIP: 188, 190, 191, 192, 193, 194, 195, 229, 230, 232, 234, LOT: 117, 118, 119, 120)?

Una vez es consultada la información es posible obtener mayor detalle acerca del artículo en cuestión mediante el uso de etiquetas (Condena/Sanción, Detalle, Ley, No gracias), las mismas realizarán una consulta hacia el objeto JSON de retorno del API para mostrar la

información seleccionada por la etiqueta haciendo uso de las intenciones Condena, Detalle y Ley; la etiqueta No gracias inicia el diálogo Opciones, cualquier otra etiqueta iniciará el diálogo “Consulta Artículo”, sin necesidad de ingresar el número del artículo generando un bucle hasta que el usuario seleccione la etiqueta No gracias.

Figura 58

Diálogo Consulta Artículo



Consulta Artículo After

Este diálogo ofrece las mismas funcionalidades que el diálogo “Consulta Artículo”, simplemente fue creado para complementar la funcionalidad del diálogo “Buscar Delito”, ya que permite consultar Condena/Sanción, Detalle, Ley del artículo que ha sido predicho por el modelo previamente seleccionado, además es generado el bucle hacia “Consulta Artículo After” hasta que la etiqueta No gracias inicie el diálogo de “Opciones”.

Acerca de Nosotros

Permite conocer a los integrantes, tutor y un breve resumen del proyecto, una vez mostrada esta información, se inicia el diálogo “Standby After Acerca”.

Figura 59*Diálogo Acerca de Nosotros*

Somos un proyecto de tesis desarrollado por Byron Motoche y David Villamarin con la tutoría de Darwin Alulema, el cual integra procesamiento de lenguaje natural y servicios del tipo cloud, enfocado en encontrar delitos relacionados a los artículos de telecomunicaciones del Código Integral Penal (COIP) y la Ley Orgánica de Telecomunicaciones (LOT).

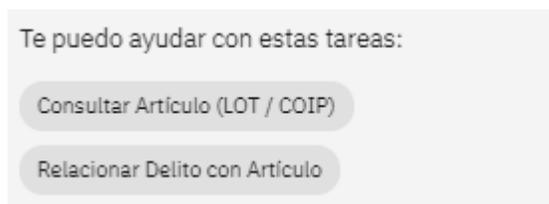
En otras cosas

Diálogo de ayuda que es iniciado cuando el asistente no ha entendido la respuesta insertada por el usuario, el mismo puede ser iniciado en cualquier flujo de la conversación.

Standby After Acerca

Diálogo generado posterior al diálogo “Acerca de Nosotros” el cual brinda la posibilidad de iniciar los diálogos “Consulta Artículo” y “Busca Delito”.

Figura 60*Diálogo Standby After Acerca*

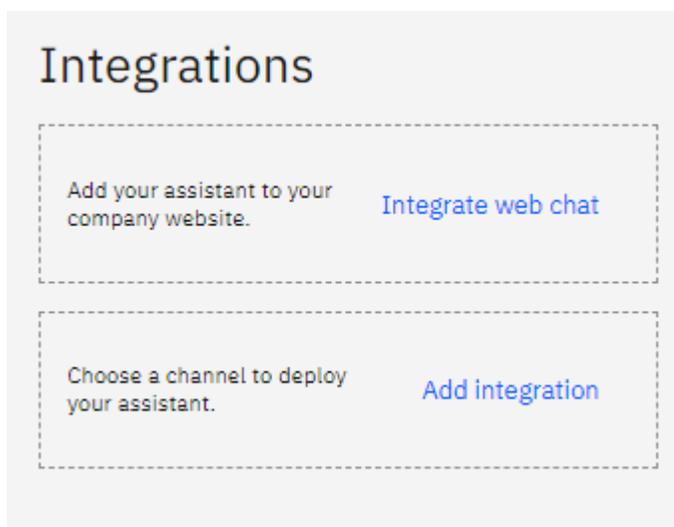


Consumir el servicio Watson Assistant

A través del uso del servicio de Watson Assistant es posible crear una integración del asistente previamente creado con todos sus componentes de manera sencilla a través de la interfaz gráfica de IBM Cloud.

Figura 61

Integración del servicio de Watson Assistant

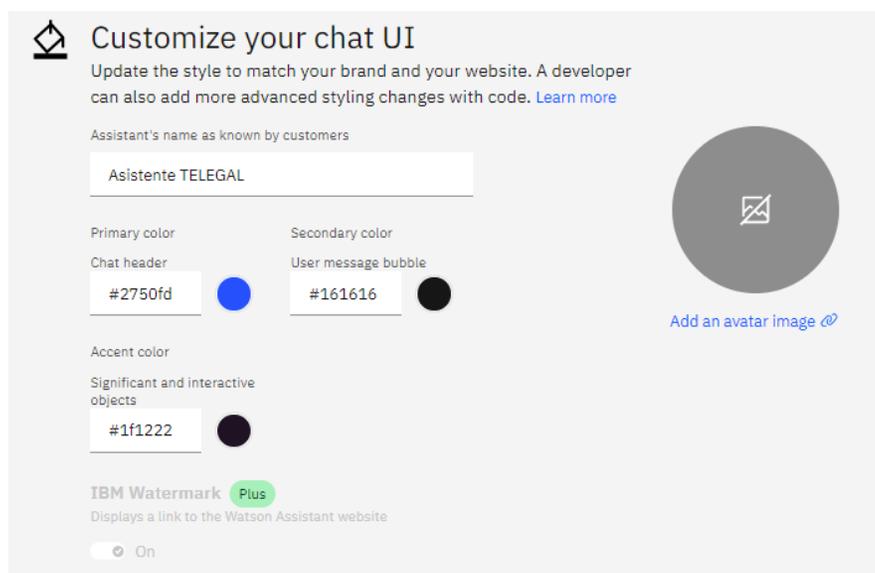


Se ha seleccionado la integración por web chat ya que a través de la misma es posible crear un widget en un cliente web de cualquier dispositivo, de esta forma es posible interactuar con el asistente. Un script del lenguaje de programación JavaScript es generado y el mismo crea una instancia única del asistente para cada cliente que solicite ayuda del asistente.

A través de la interfaz de IBM Cloud es posible personalizar el nombre y los colores primarios, secundarios y de acento del asistente, los cuales se verán reflejados en el cliente web, así mismo existe una funcionalidad para mantener la seguridad del web chat a través de encriptación de data sensible, autenticación y autorización, funcionalidad que no ha sido usada, ya que el manejo de autenticación se dará a través del frontend, pudiendo así solamente usar el asistente los usuario registrados a la plataforma.

Figura 62

Personalización del asistente



Customize your chat UI
Update the style to match your brand and your website. A developer can also add more advanced styling changes with code. [Learn more](#)

Assistant's name as known by customers
Asistente TELEGAL

Primary color
Chat header
#2750fd

Secondary color
User message bubble
#161616

Accent color
Significant and interactive objects
#1f1222

IBM Watermark **Plus**
Displays a link to the Watson Assistant website
 On

Add an avatar image 

Este es el paso final del backend del proyecto ya que a través del widget será posible usar el asistente y todos los módulos previamente creados, desde el frontend del proyecto como se detalla posteriormente.

Asistente de Voz

El asistente de voz implementado integra todos los servicios y APIs previamente explicados e incorpora adicionalmente los servicios de Speech to Text (para capturar la voz del usuario) y Text-to-Speech (para reproducir la respuesta) que ofrece IBM Watson para extender la funcionalidad del asistente de texto y ofrecer una interacción totalmente basada en la voz en una aplicación web.

La implementación del asistente de voz se realizó mediante el uso del lenguaje de programación Python empleando el Framework Flask y la librería de IBM Watson para utilizar los servicios de Speech-to-Text, Text-to-Speech y Watson Assistant. Adicionalmente para la página web se utilizó la librería de JQuery, HTML5 y CSS.

IBM Watson Text to Speech

IBM Watson Text to Speech es un servicio REST de IBM Cloud que permite convertir texto a voz en diversos idiomas y voces con un sonido natural. Para la presente arquitectura este servicio se encargará de convertir la respuesta o los mensajes que generó el asistente de Watson a un audio de voz. El audio generado se envía a la aplicación web y se reproduce de

forma automática. Adicionalmente a la reproducción de audio, en la interfaz de usuario se muestra la respuesta del asistente como texto.

Para utilizar este servicio se debe crear o instanciar el mismo en IBM Cloud. Una vez creado el servicio REST, este se puede consumir utilizando el método POST de HTTP y las credenciales del servicio REST (API KEY, URL). El servicio de Texto to Speech utilizado es del plan Lite o gratuito de IBM Cloud, que permite la traducción de hasta un total de 10 000 caracteres mensualmente. Para el uso del mismo se deben enviar los siguientes parámetros en la solicitud al servicio:

- **text:** representa el texto que se desea convertir a voz, por ejemplo: “Hola Bienvenidos”
- **accept:** define el formato de audio que se retornará, por ejemplo: audio/wav
- **voice:** define el idioma y tipo de voz para generar la respuesta, por ejemplo: “es-ES_LauraVoice” indica que se obtendrá la respuesta en español de España de la voz del tipo LauraVoice.

Para la presente arquitectura, en Flask se definió la ruta “/api/text-to-speech” que procesará las peticiones del tipo “POST” para este servicio y retornará el audio de voz.

IBM Watson Speech to Text

El servicio IBM Watson Speech to Text proporciona una REST API para la traducción de voz en diversos idiomas y formatos de audio a texto plano. En la presente arquitectura este servicio tiene la función de traducir la entrada del usuario que será a través de la voz a texto

para ser interpretado por el Asistente de Watson. Además, el texto transcrito por el servicio se despliega como texto en la interfaz de usuario.

Al igual que el servicio de Text to Speech para utilizar el servicio de Speech to Text en primer lugar se debe crear o instanciar el mismo en IBM Cloud y obtener las credenciales del servicio. Este servicio REST se puede consumir de forma síncrona y asíncrona con solicitudes HTTP. Adicional al consumo mediante solicitudes HTTP se puede consumir el servicio mediante WebSockets, que ofrecen la funcionalidad de establecer un canal de comunicación de baja latencia y full dúplex, donde se envía las solicitudes y audio al servicio y se retornan los resultados a través de una sola conexión de forma asíncrona.

El servicio de Speech to Text utilizado es del plan Lite de IBM Cloud, que permite la traducción de hasta un total de 500 minutos mensualmente de forma gratuita. Para la presente arquitectura se utilizó WebSockets para enviar y recibir la respuesta de las peticiones al servicio REST de Speech to Text, por lo cual se debe hacer uso del método “recognize()” y enviar los siguientes parámetros en la solicitud:

- **audio**: representa el audio que se desea convertir a texto
- **content_type**: define el formato de audio que se envía, por ejemplo: “audio/wav”
- **model**: define el modelo que se utilizará para reconocer el audio, por ejemplo: 'es-MX_NarrowbandModel', utiliza como frecuencia de muestreo 8 kHz.

En Flask se define la ruta “/api/speech-to-text” que procesará las peticiones del tipo “POST” para este servicio y retornará la transcripción del audio como texto y se envía como entrada al Asistente de Watson. En caso de que el servicio no pueda transcribir el audio enviado, se retornará un mensaje indicando que no se pudo realizar la transcripción.

Asistente de Watson

Los servicios de Text to Speech y Speech to Text por sí solos no pueden interactuar con el usuario, por tal razón es necesario de un componente que se encargue de interactuar con el usuario para obtener información y procesarla. En la presente arquitectura el componente tiene esta función es el Asistente de Watson, quién se comunica con el usuario utilizando la inteligencia artificial para brindar una respuesta rápida al usuario.

El Asistente de Voz implementado hace uso del asistente de texto, los servicios y APIs explicados previamente. Sin embargo, debido a la limitación de la capa gratuita se abrevio las respuestas del Asistente de Watson para no agotar los servicios del plan lite. En Flask se define la ruta `"/api/conversation"` que procesará las peticiones del tipo `"POST"` y `"GET"` para este servicio. Esta ruta funciona como una caja negra a la cual se envía los siguientes parámetros:

- **Context:** es un identificador para distinguir cada conversación y en que diálogo se encuentran las mismas
- **workspace_id:** es el id del Asistente de Watson
- **Input:** es la entrada del usuario en texto plano

Se obtiene como respuesta los siguientes parámetros:

- **responseText:** la respuesta que entrega el asistente de Watson a la entrada del usuario.
- **context:** identificador para distinguir cada conversación y en que diálogo se encuentran las mismas

El Asistente de Watson por defecto solo interpreta solicitudes enviadas en formato de texto, es decir, no puede interpretar directamente una entrada de audio o responder mediante uno; para implementar esa funcionalidad se definen ciertas funciones asíncronas en JavaScript que se ejecutan en el lado del cliente e interconectan a los servicios mediante las rutas creadas para cada servicio en Flask y se encargan además de renderizar el contenido en la página web y de reproducir los archivos de audio.

Base de Datos de la Aplicación Web y Móvil

La base de datos a la que se refiere esta sección es la DB Aplicación, explicada en la arquitectura del sistema, la cual posee cuatro tablas, donde se engloba todo el comportamiento de la aplicación web y móvil.

Se ha definido la tabla denominada "Usuarios", su objetivo es poder registrar y validar los diferentes usuarios que pueden ser creados, posee 9 columnas las cuales son:

Figura 63

Detalle de tabla Usuarios

Usuarios
IdUsuario: VARCHAR(20) NOT NULL [PK]
Estatus: VARCHAR(1) NOT NULL
Clave: VARBINARY(15) NOT NULL
Apellidos: VARCHAR(40) NOT NULL
Nombres: VARCHAR(40) NOT NULL
Email: VARCHAR(200) NOT NULL
FehcaIng: DATE NOT NULL
UsuarioIng: VARCHAR(20) NOT NULL
TerminalIng: VARCHAR(20) NOT NULL

- **IdUsuario** columna del tipo VARCHAR de longitud 20, definida como clave principal de la tabla y encargada de identificar de manera única a los usuarios de la aplicación.
- **Estatus** columna del tipo VARCHAR de longitud 1, donde se valida si el usuario se encuentra activo o no.
- **Clave** columna del tipo VARBINARY de longitud 15, encargada de almacenar en una serie binaria la clave introducida por el usuario, a través de este campo y de la identificación del usuario se puede realizar el login a las aplicaciones.
- **Apellidos** columna del tipo VARCHAR de longitud 40, donde se almacenan los apellidos del usuario.
- **Nombres** columna del tipo VARCHAR de longitud 40, donde se almacenan los nombres del usuario.
- **Email** columna del tipo VARCHAR de longitud 200, donde se almacena el correo electrónico del usuario.
- **Fecha Ingreso** columna del tipo DATE, encargada de brindar información sobre la fecha en la que fue ingresado el registro.

- **Usuario Ingreso** columna del tipo VARCHAR de longitud 20, encargada de brindar información sobre el usuario que ingresó el registro.
- **Terminal Ingreso** columna del tipo VARCHAR de longitud 20, encargada de brindar información sobre el lugar en que fue ingresado el registro, pudiendo ser WEB o MOVIL dependiendo de la aplicación que se esté usando.

Es importante señalar que las últimas tres columnas Fecha Ingreso, Usuario Ingreso y Terminal Ingreso son usadas como servicio de no repudio, así mismo el atributo NOT NULL presente en cada columna, significa que la misma es de carácter obligatorio, es decir, todas las columnas deben ser insertadas con información válida.

Se ha definido la tabla denominada "Estadísticas", su objetivo es poder registrar la información de pruebas realizadas usando la matriz de confusión, las mismas serán detalladas en el capítulo 4, posee 7 columnas las cuales son:

Figura 64

Detalle de tabla Estadísticas

Estadísticas
Id: INTEGER NOT NULL [PK]
Modelo: VARCHAR(6) NOT NULL
Clase: VARCHAR(10) NOT NULL
Exactitud: NUMERIC(6, 4) NOT NULL
Precision: NUMERIC(6, 4) NOT NULL
Exhaustividad: NUMERIC(6, 4) NOT NULL
F1: NUMERIC(6, 4) NOT NULL

- **Id** columna del tipo INTEGER, a la cual se le definió el atributo de identidad de 1 y 1, lo que significa que genera valores secuenciales automáticos iniciados en el valor 1, e incrementa este valor cada vez que un registro es ingresado a la tabla, esta columna ha sido definida como clave principal de la tabla, su función es identificar de manera única cada uno de los registros.
- **Modelo** columna del tipo VARCHAR de longitud 6, donde se diferencia los resultados para cada uno de los modelos sus valores son: AWS, AZURE, WATSON.
- **Clase** columna del tipo VARCHAR de longitud 10, donde se diferencia cada uno de los artículos de los cuales se ha obtenido las variables estadísticas sus valores son: LOT_117, LOT_118, LOT_119, LOT_120, COIP_188, COIP_190, COIP_191, COIP_192, COIP_193, COIP_194, COIP_195, COIP_229, COIP_230, COIP_232 y COIP_234.
- **Exactitud** columna del tipo NUMERIC de longitud 6 y dígitos decimales 4, encargada de almacenar los valores de exactitud de las estadísticas de cada clase.
- **Precisión** columna del tipo NUMERIC de longitud 6 y dígitos decimales 4, encargada de almacenar los valores de precisión de las estadísticas de cada clase.
- **Exhaustividad** columna del tipo NUMERIC de longitud 6 y dígitos decimales 4, encargada de almacenar los valores de exhaustividad de las estadísticas de cada clase.
- **F1** columna del tipo NUMERIC de longitud 6 y dígitos decimales 4, encargada de almacenar los valores de F1 Score de las estadísticas de cada clase.

Se ha definido la tabla denominada "Encuesta", su objetivo es poder registrar la información de las encuestas de experiencia de usuario disponibles en las aplicaciones, posee 18 columnas las cuales son:

Figura 65

Detalle de tabla Encuesta

Encuesta	
Id:	INTEGER NOT NULL [PK]
Usuario:	VARCHAR(20) NOT NULL
Pregunta1:	INTEGER NOT NULL
Pregunta2:	INTEGER NOT NULL
Pregunta3:	INTEGER NOT NULL
Pregunta4:	INTEGER NOT NULL
Pregunta5:	INTEGER NOT NULL
Pregunta8:	INTEGER NOT NULL
Pregunta6:	INTEGER NOT NULL
Pregunta7:	INTEGER NOT NULL
Pregunta8:	INTEGER NOT NULL
Pregunta9:	INTEGER NOT NULL
Pregunta10:	INTEGER NOT NULL
Pregunta11:	INTEGER NOT NULL
Pregunta12:	INTEGER NOT NULL
Pregunta13:	INTEGER NOT NULL
FehcaIng:	DATE NOT NULL
UsuarioIng:	VARCHAR(20) NOT NULL
TerminalIng:	VARCHAR(20) NOT NULL

- **Id** columna del tipo INTEGER, a la cual se le definió el atributo de identidad de 1 y 1, lo que significa que genera valores secuenciales automáticos iniciados en el valor 1, e incrementa este valor cada vez que un registro es ingresado a la tabla, esta columna ha sido definida como clave principal de la tabla, su función es identificar de manera única cada uno de los registros.

- **Usuario** columna del tipo VARCHAR de longitud 20, encargada de almacenar la información del usuario que ha llenado la encuesta.
- **Pregunta 1 - 13** columna del tipo INTEGER, la cual representa el valor seleccionado entre 1 y 5 en la encuesta para cada una de las preguntas planteadas, las cuales se detallan en la sección pruebas cualitativas en el capítulo 4.

Las tres últimas columnas son usadas como servicio de no repudio.

Se ha definido la tabla denominada "Pruebas Tiempos", su objetivo es poder registrar la información de las pruebas de rendimiento de cada uno de los modelos, posee 3 columnas las cuales son:

Figura 66

Detalle de tabla Pruebas Tiempos

PruebasTiempos	
Tiempo:	INTEGER NOT NULL
Horario:	VARCHAR(9) NOT NULL
Modelo:	VARCHAR(6) NOT NULL

- **Tiempo** columna del tipo INTEGER donde se almacena el valor del tiempo que tardó cada uno de los modelos en procesar una petición en milisegundos.
- **Estatus** columna del tipo VARCHAR de longitud 9, encargada de almacenar el valor del horario en el que se plantearon las pruebas, sus valores son: Madrugada, Mediodía y Tarde.

- **Modelo** columna del tipo VARCHAR de longitud 6, la cual almacena el valor del modelo del cual se realizó la prueba, sus valores son: AWS, AZURE, WATSON.

En el capítulo 4 se detalla cómo fue usada esta tabla para cada una de las pruebas realizadas.

Servicio WCF

Para efectos de uso de la base de datos previamente creada, ya que se trata de una base de datos la cual necesitará realizar inserciones y consultas de información, se creó un servicio web del tipo Windows Communication Foundation (WCF) el cual es un framework para construir aplicaciones y arquitecturas orientadas hacia los servicios, donde se busca el intercambio de información desde y hacia servicios alojados en servidores o en aplicaciones. Esta herramienta permite desarrollar, manejar y consumir servicios web de manera rápida y sencilla.

Es posible destacar las siguientes características de estos servicios:

- **Orientado a Servicios:** permitiendo así que el acoplamiento flexible en lugar de usar los servicios codificados en cada aplicación, de esta forma el servicio ha sido integrado tanto en la aplicación web, como en la aplicación móvil, manteniendo su misma codificación sin necesidad de hacer distinciones adicionales para el manejo en diferentes plataformas.
- **Interoperabilidad:** este tipo de servicios ofrecen una gran integración e interoperabilidad con diversas plataformas de Microsoft entre las que destacan ASP.NET y COM+.

- **Metadatos:** este tipo de servicios usan los formatos estandarizados tales como WSDL, esquemas XML y WS-Policy para generar y configurar de manera automática clientes que puedan hacer uso del mismo.
- **Seguridad:** es posible encriptar y proteger las comunicaciones desde y hacia estos servicios usando estándares como Secure Socket Layer (SSL).

A través de la plantilla proporcionada por Visual Studio para el tipo de aplicaciones "WCF Service Application" un proyecto es creado con un ejemplo de su uso. Básicamente se crean tres archivos los cuales son:

- **IService:** interfaz del servicio, donde se realizan las declaraciones definiendo las diversas operaciones del servicio denominados "Operation Contract", es declarado el nombre y los parámetros que tendrá la operación anteponiendo la etiqueta [OperationContract], se usarán datos primitivos en su mayoría a excepción del tipo de datos DataTable provisto por la dependencia System.Data
- **Service.svc:** es el archivo principal del servicio, donde se encuentra el código de cada una de las operaciones definidas en la interfaz, por lo tanto, este archivo hereda la interfaz previamente creada.
- **Web.config:** detalles de la configuración del endpoint o cadenas de conexión hacia la base de datos.

Se definieron 6 operaciones las cuales se conectan hacia la base de datos DB Aplicación a través de una cadena de conexión, un objeto SqlConnection (encargado de establecer la conexión con el servidor de base de datos) y un SqlCommand (encargado de ejecutar comandos del tipo SQL), estas operaciones se detallan a continuación:

- **Login:** encargada de realizar la validación de los datos de los usuarios, ingresados en el login.
- **GrabaUsuario:** encargada de almacenar los datos de registro insertados por los nuevos usuarios de la aplicación.
- **GrabaEncuesta:** encargada de almacenar los datos de las encuestas realizadas por los usuarios a través de la interfaz.
- **ConsultaEstadísticas:** encargada de consultar los datos de las estadísticas obtenidas para cada uno de los modelos.
- **ConsultaPruebasTiempos:** encargada de consultar los datos de las pruebas realizadas para cada modelo en cada horario establecido.
- **ConsultaEncuestas:** encargada de consultar los datos de las encuestas realizadas por los usuarios.

Cada operación tiene asociado un procedimiento almacenado que contiene un SQL query, el cual es ejecutado a través del SqlCommand, con el envío de parámetros de ser necesarios, así es posible obtener la respuesta a esta operación desde la aplicación web, móvil y cualquier cliente en general.

Servidor

Ya que han sido creados varios servicios del tipo Web API y WCF es necesario alojarlos de manera que se encuentren accesibles a través del internet, estos servicios en conjunto con las herramientas y servicios provistos por cada una de las plataformas, permiten al sistema funcionar de manera conjunta.

El servidor usado para alojar los diversos servicios y la aplicación web que será detallada en la siguiente subsección, es Internet Information Services (IIS), el cual es un servidor web flexible, ejecutado en un sistema Windows "Microsoft Windows Server 2019 Essentials" este tipo de servidor aceptar peticiones de clientes remotos y ofrece la respuesta adecuada en el formato configurado para las mismas. Soporta los siguientes protocolos, Hypertext Transfer Protocol (HTTP) en su versión 1 y 2, Hypertext Transfer Protocol Secure (HTTPS), File Transfer Protocol (FTP) y su versión segura FTPS, Simple Network Management Protocol (SMTP) y Network News Transport Protocol (NNTP). Adicionalmente el servidor cuenta con una memoria RAM física de 4 GB y una memoria RAM virtual de 8 GB. Su procesador es Intel Core 2 Quad Q8200 de 4 núcleos y 4 subprocesos.

Este servidor funciona con ASP.NET framework y permite alojar Active Server Page (ASP) el cual es un motor de secuencia de comandos del lado del servidor que produce páginas web interactivas con la cual ha sido realizada la aplicación web y adicionalmente permite alojar servicios que usan el framework Web API y WCF, en general puede ser alojada cualquier tipo de proyecto realizado con las herramientas de Microsoft en este servidor.

Frontend

En la sección anterior se señaló que el servidor usado para el proyecto ha sido un IIS y que a través del uso del framework ASP.NET es posible la creación de páginas web interactivas, así como también es posible alojar diversos servicios web, los cuales interactúan entre sí ofreciendo la posibilidad de interconectar bases de datos y lógica específica, de acuerdo a las necesidades de las aplicaciones.

Para el manejo del frontend de este proyecto se plantean los siguientes escenarios:

- **Interfaz web:** como se ha mencionado a través de la plataforma .NET es posible usar diferentes framework entre los cuales se encuentra ASP el cual permite la creación de páginas web dinámicas, adicionando lógica a través de los lenguajes de programación C#, F# o Visual Basic. En este caso se plantea para la aplicación web, el uso de Visual Basic y adicionalmente la suite de controles Telerik UI para ASP.NET AJAX, “el cual permite obtener alrededor de 120 componentes de formularios Web Forms ASP.NET versátiles y optimizados para el rendimiento, de manera que permite crear proyectos profesionales de alta calidad...” (Telerik, 2021)
- **Interfaz móvil:** para la aplicación móvil se usa la plataforma de desarrollo de .NET, Xamarin, la cual permite la creación de aplicaciones Android y iOS con .NET y C#. Donde es posible tener una única base compartida de código, de manera que se reúse un alto porcentaje del mismo entre plataformas a través del uso de Xamarin.Forms una de las característica más importantes de Xamarin, es posible integrar páginas, diseños y controles para crear y diseñar aplicaciones móviles desde una única API que es altamente extensible, en esta plataforma se usa el patrón de diseño Model-View-ViewModel (MVVM) donde la interfaz de usuario se denomina vista (view), la estructura de los datos, modelo (model) y la lógica del negocio o aplicación es el modelo de vista (view model), conjuntamente con el set de controles de la plataforma Telerik UI para Xamarin. Es importante señalar que para este proyecto se creará solamente la aplicación de la plataforma Android.

Interfaz web

Debido a que la característica principal de este proyecto es el asistente de texto y de voz, donde se incorporan la mayoría de los servicios web creados, así como también base de datos y los modelos de machine learning de cada una de las plataformas. Para consumir estos asistentes es necesario el uso de un cliente web, en otras palabras, una página HTML que al menos integre JavaScript.

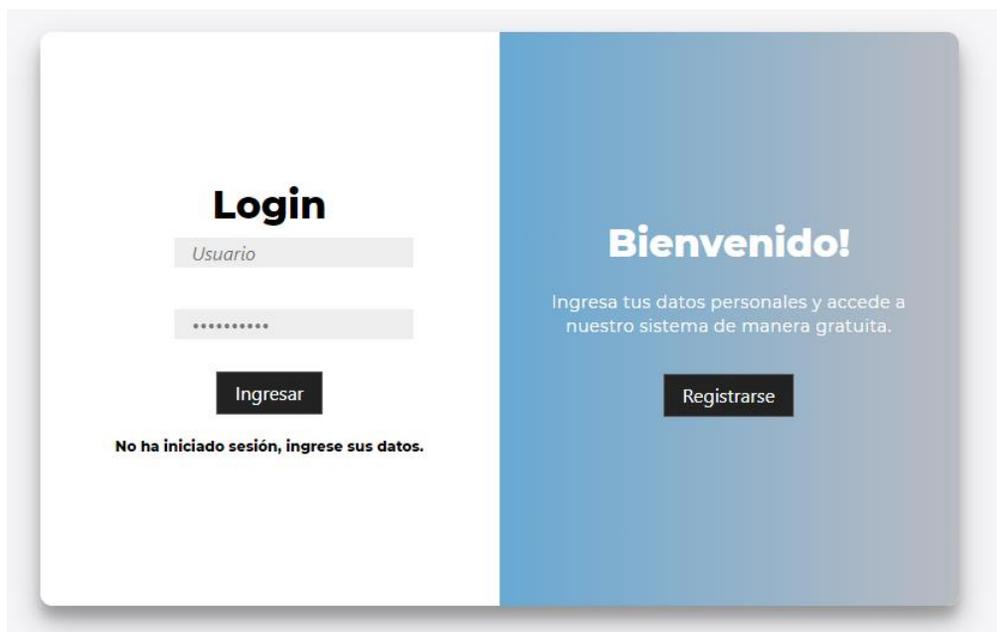
Se ha dividido la interfaz en dos secciones las cuales son: Registro - Login y página Principal.

Registro y Login

Para hacer uso de la aplicación web a través de la pantalla principal, es posible realizar una de las siguientes opciones:

Figura 67

Pantalla de inicio de la aplicación web



- **Registro:** una vez realizado un click en el botón registrarse, se inicia una nueva pantalla con el proceso de registro el cual permite ingresar datos y generar un usuario único que pueda acceder a la página principal de la aplicación.

Figura 68

Pantalla de registro de la aplicación web



The image shows a registration form on a blue background. The form consists of six input fields stacked vertically, each with a label to its left: 'Nombre', 'Apellido', 'Usuario', 'Email', 'Contraseña', and 'Confirmar Contraseña'. Below the last two fields, there is a text label 'No registrado.' and a button labeled 'Registrarse'.

- **Login:** una vez registrado, se podrá hacer uso de la aplicación en su totalidad, accediendo a la página principal de la misma, brindando Usuario y Contraseña válidos.

Página principal

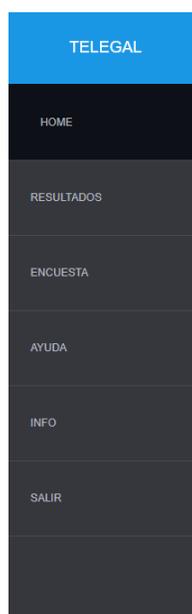
Una vez realizado el login, es posible acceder hacia la pantalla principal de la aplicación web, donde se usan varios recursos importantes de la plataforma entre los que destacan:

- **Valores de Sesión:** a través del uso del objeto `HttpSessionState`, el cual se refiere a la petición HTTP actual, es posible almacenar el valor del usuario, el cual será usado a través de la aplicación.
- **Master Page:** básicamente permite crear una apariencia consistente para todas las páginas que se deseen en la aplicación web. Es usada como una plantilla con diseño, estilos y funcionalidades compartidas en este caso esto se manejará el menú lateral de manera compartida a través de la página maestra y adicionalmente se incluye al menos

un control ContentPlaceholder, el cual define una región específica de la pantalla para que se muestre un contenido específico. Este contenido cambia de acuerdo al ítem del menú lateral que sea seleccionado.

Figura 69

Menú lateral establecido en la Master Page



En la página maestra se validará la presencia del valor de sesión de usuario, de no existir este valor, se presentará un error imposibilitando el uso de la aplicación hasta realizar un login de manera satisfactoria.

Figura 70

Página de error de la aplicación web

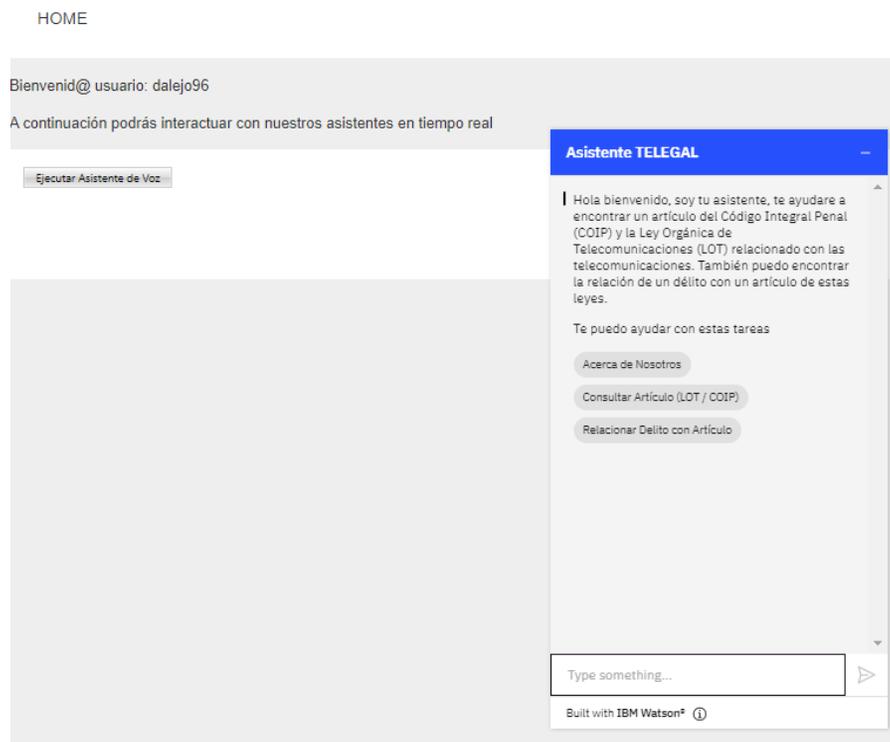


La página principal de la aplicación ofrece la posibilidad de navegar por cinco pantallas de contenido, las cuales son:

- **Home:** a través de esta página es posible acceder a los asistentes de texto y voz, el asistente de texto es desplegado por defecto con su diálogo de bienvenida y un botón es usado para abrir una nueva página web con el asistente de texto.

Figura 71

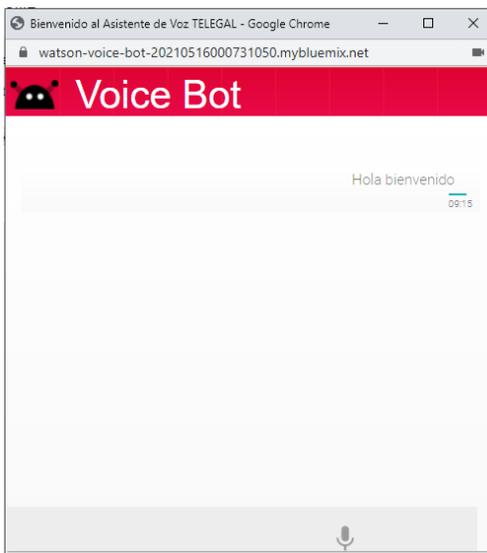
Asistente de texto en la página de Inicio



Una vez realizado click en el botón “Ejecutar Asistente de Voz”, se despliega la siguiente página.

Figura 72

Asistente de Voz en página auxiliar

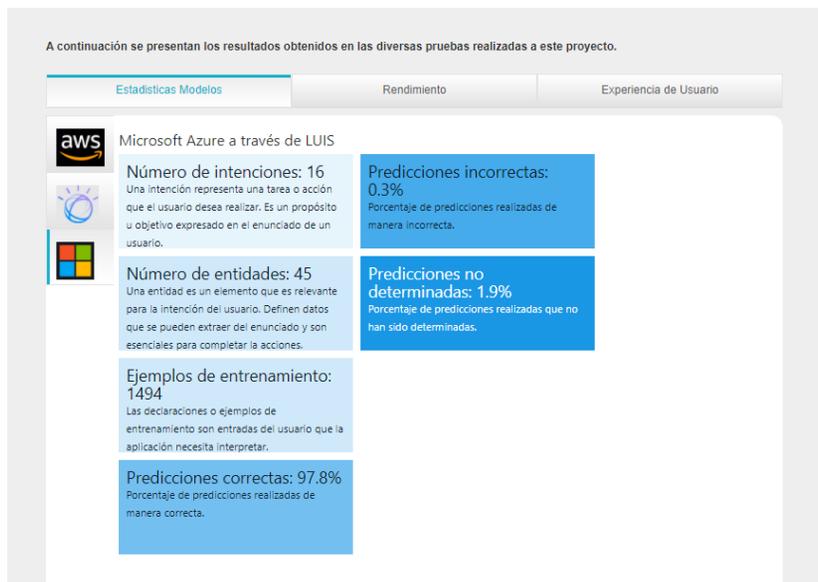


- **Resultados:** a través de esta página el usuario de la aplicación puede conocer y visualizar diversa información obtenida mediante las pruebas realizadas. Esta página está dividida en tres tabs que son:
 - **Estadísticas de Modelos:** en este tab se puede visualizar los resultados obtenidos en cada plataforma, luego del entrenamiento del modelo, se muestra información relevante como diversas métricas, número de documentos usados en el proceso de entrenamiento, entidades, intenciones y etiquetas, entre otros. Esta información no ha sido procesada, es meramente informativa y fue extraída de cada plataforma.

Figura 73

Contenido del tab Estadísticas Modelos de la página de Resultados

RESULTADOS



Mediante la barra lateral con los logos de Amazon Web Services, IBM Watson y Microsoft Azure es posible navegar hacia el detalle de cada una de las plataformas.

- **Rendimiento:** en este tab se puede visualizar los resultados de cada uno de los modelos, mediante el uso de una gráfica de dispersión es posible analizar el tiempo de respuesta de los modelos para tres horarios definidos (Madrugada, Mediodía, Tarde), una imagen con la tabulación de las pruebas en una matriz de confusión y las métricas de cada uno de los artículos basado en esta matriz, el detalle de estos cálculos será abordando a profundidad en el capítulo 4.

Figura 74

Contenido del tab Rendimiento de la página de Resultados

RESULTADOS

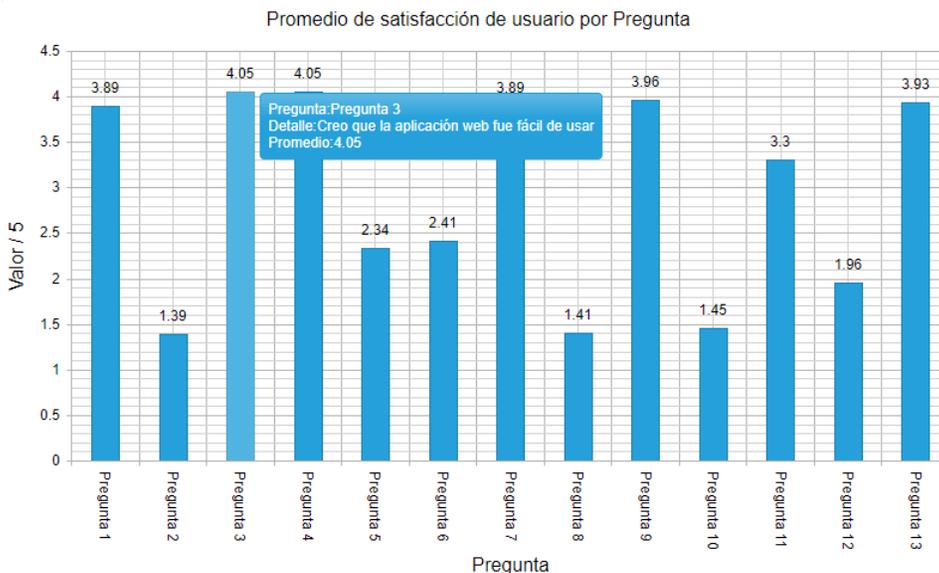


De la misma forma que en el tab previo, es posible navegar hacia el detalle de cada una de las plataformas mediante el uso del menú lateral.

- **Experiencia de Usuario:** se han realizado un gráfico de barras, el cual permite conocer los valores promedio de cada pregunta, de esta forma se obtiene una evaluación general de la experiencia de los usuarios con la aplicación web.

Figura 75

Contenido del tab Experiencia de Usuario de la página de Resultados



- Encuesta:** a través de esta página es posible registrar los datos de una encuesta de experiencia de usuario (UX) la cual fue realizada basada en el sistema de escalas de usabilidad (SUS por sus siglas en inglés) con una puntuación entre 1 y 5, la cual permite medir la satisfacción general que ha tenido el usuario al usar la aplicación, esta encuesta en su totalidad se detalla en la sección pruebas cualitativas en el capítulo 4.

Figura 76

Contenido de la página Encuesta

ENCUESTA

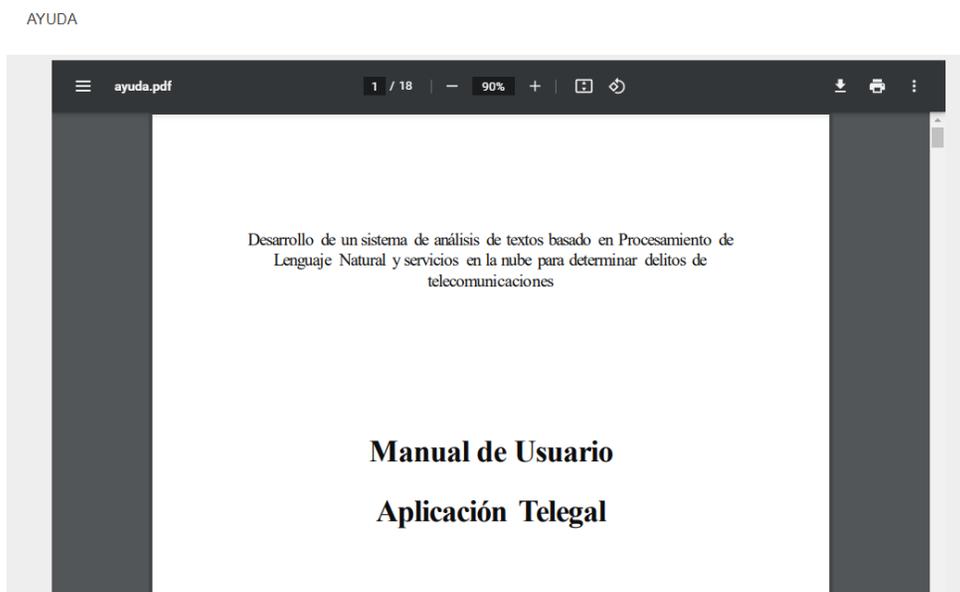
A continuación se presenta una pequeña encuesta de la experiencia de usar el sistema, responda de manera honesta entre 1 a 5, donde 1 significa Total desacuerdo y 5 significa Total acuerdo, una vez concluido realizar click en aceptar.

Creo que usaría esta aplicación frecuentemente	★☆☆☆☆
Encuentro esta aplicación innecesariamente complejo	★☆☆☆☆
Creo que la aplicación web fue fácil de usar	★☆☆☆☆
Creo que el asistente de texto fue fácil de usar	★☆☆☆☆
Creo que el asistente de voz fue fácil de usar	★☆☆☆☆
Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación	★☆☆☆☆
Las funciones de esta aplicación están bien integradas	★☆☆☆☆
Creo que la aplicación es muy inconsistente	★☆☆☆☆
Imagino que la mayoría de la gente aprendería a usar esta aplicación en forma muy rápida	★☆☆☆☆
Encuentro que la aplicación es muy difícil de usar	★☆☆☆☆
Me siento confiado al usar esta aplicación	★☆☆☆☆

- **Ayuda:** dentro de esta página se encuentra un manual de usuario, donde se detallan los procesos de registro, login, así como el manejo de cada uno de los asistentes y la navegación general que puede existir dentro de la aplicación.

Figura 77

Contenido de la página Ayuda con el Manual de Usuario



- **Info:** dentro de esta página se encuentra información referente a los integrantes del grupo, el tutor, título y objetivo general y específicos, simplemente es una página informativa acerca del proyecto realizado.

Figura 78

Contenido de la página Info

Título del trabajo

Desarrollo de un sistema de análisis de textos basado en Procesamiento de Lenguaje Natural y Servicios en la Nube para determinar delitos de telecomunicaciones.

Objetivo General

- Diseñar e implementar un sistema de análisis de procesos judiciales, basado en Procesamiento de Lenguaje Natural (PLN) y Servicios en la Nube para determinar delitos en sistemas de telecomunicaciones y su relación con la normativa jurídica del Ecuador.

Objetivos Específicos

- Crear una base de datos con los procesos judiciales relacionados a delitos informáticos y de telecomunicaciones basados en los registros de la función judicial del Ecuador.
- Analizar las plataformas proveedoras de servicios en la nube con inteligencia artificial y procesamiento de lenguaje natural.
- Entrenar al sistema con los datos obtenidos de procesos judiciales relacionados a delitos informáticos y de telecomunicaciones.
- Diseñar una arquitectura basada en servicios para la detección de delitos informáticos y de telecomunicaciones.
- Realizar pruebas de rendimiento, funcionamiento y usabilidad de la plataforma.

Adicionalmente existe la opción salir, donde se retorna hacia la pantalla de login y registro, adicionalmente se borran las cookies de sesión. Reiniciando el flujo de la aplicación.

Interfaz móvil

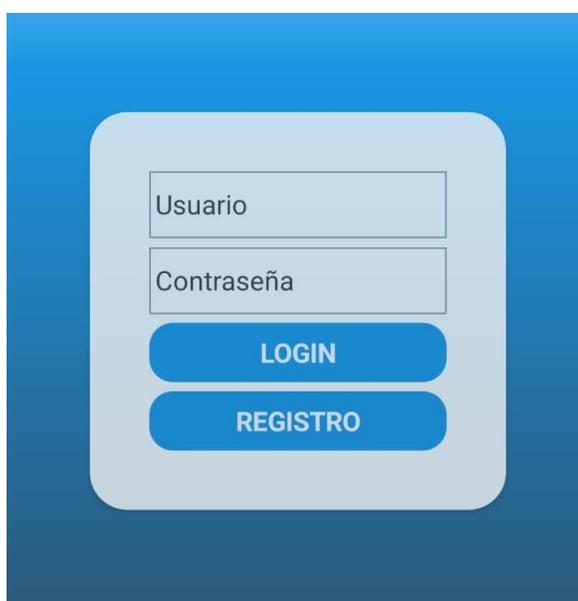
De la misma forma que en la subsección anterior, para consumir estos asistentes es necesario el uso de un cliente web, en otras palabras, una página HTML que al menos integre JavaScript, razón por la cual la aplicación móvil implementada será híbrida, es decir que algunos componentes y partes de la misma se generarán como procesos web y se implementará un control dentro de la misma para poder visualizar estas páginas, además, se tendrán componentes y bloques desarrollados de manera nativa que permitan hacer uso de las mismas funcionalidades planteadas en la interfaz web.

Registro - Login

Para hacer uso de la aplicación móvil, se tiene una página inicial similar a la página de la aplicación web, en la cual es posible realizar una de las siguientes opciones:

Figura 79

Pantalla de inicio de la aplicación móvil



- **Registro:** una vez realizado un click en el botón registrarse, se inicia una nueva pantalla con el proceso de registro el cual una vez ingresados los datos correctos, genera un usuario único el cual puede acceder a la aplicación.

Figura 80

Pantalla de registro de la aplicación móvil

Nombres

Apellidos

Usuario

Email

Contraseña

Confirmar Contraseña

REGISTRARSE

- **Login:** una vez registrado, se podrá hacer uso de la aplicación en su totalidad, accediendo a la página principal de la misma, los valores requeridos son el usuario único y la contraseña.

Actividad Principal

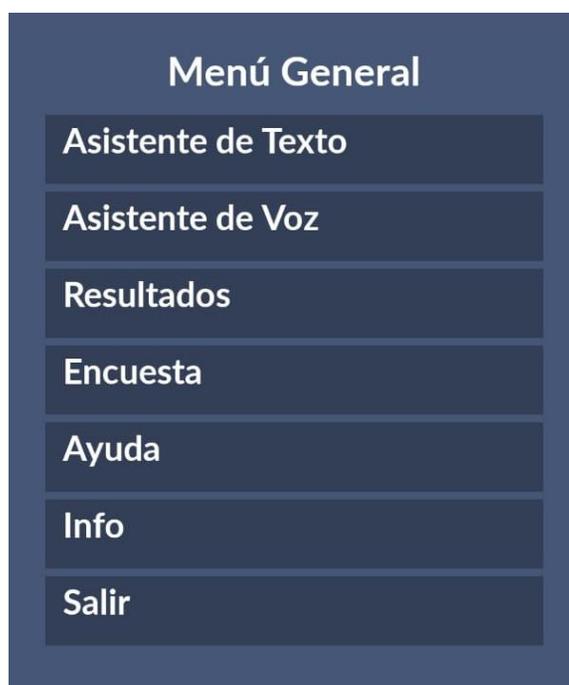
Una vez realizado el login, es posible acceder hacia la pantalla principal de la aplicación móvil, donde se usan:

- **Master Detail Page:** es una página responsable de dos páginas a la vez, siendo la master page la que contiene los ítems y la detail page, donde se muestran los detalles de estos ítems seleccionables de la master page. Funciona de manera similar a la master page explicada en la subsección anterior, la misma se encarga de manejar el menú de la

aplicación y a través de la página de detalle, es posible visualizar cada una de las páginas.

Figura 81

Menú lateral establecido en la Master Detail Page de la aplicación móvil



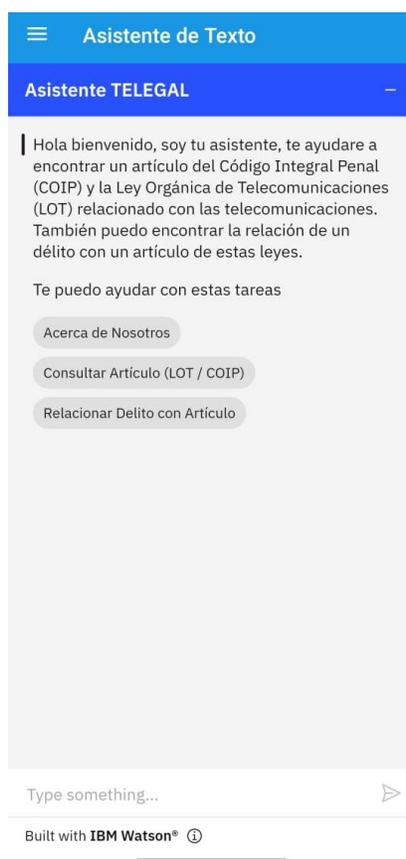
Los datos del usuario que inició sesión serán transmitidos por de la página maestra hacia las páginas de detalle a través de la navegación propia de Xamarin Forms.

Es posible navegar por seis pantallas de detalle, las cuales son:

- **Asistente de Texto:** en esta página se encuentra el asistente de texto aislado, el cual siempre será ingresado como página por defecto una vez efectuado el login satisfactorio.

Figura 82

Asistente de texto en aplicación móvil



- **Asistente de Voz:** en esta página se encuentra el asistente de voz aislado.

Figura 83

Asistente de Voz en aplicación móvil



- **Resultados:** esta página ofrece la misma información que la aplicación web, a través del uso de tres tabs, la única diferencia es que cada tab es una página scrolleable:
 - **Modelos:** se visualizan los resultados obtenidos en cada plataforma.

Figura 84

Contenido del tab Modelos de la página de Resultados de la aplicación móvil



Mediante la barra superior es posible navegar hacia los siguientes tabs.

- **Rendimiento:** se visualizan los resultados de cada uno de los modelos, mediante el uso de gráficas y tablas.

Figura 85

Contenido del tab Rendimiento de la página de Resultados de la aplicación móvil



- **UX:** se han realizado un gráfico de barras, el cual permite conocer los valores promedio de cada pregunta, obteniendo una evaluación general de la experiencia de los usuarios con la aplicación móvil.

Figura 86

Contenido del tab UX de la página de Resultados de la aplicación móvil



- **Encuesta:** a través de esta página es posible registrar los datos de una encuesta de experiencia de usuario (UX) la cual permite medir la satisfacción general que ha tenido el usuario al usar la aplicación móvil.

Figura 87

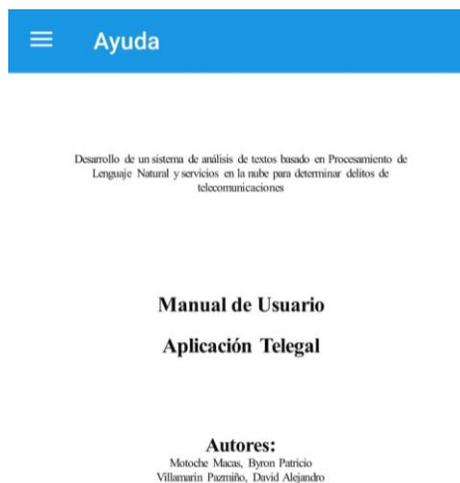
Contenido de la página Encuesta en la aplicación móvil



- **Ayuda:** dentro de esta página se encuentra el manual de usuario donde se detallan varios procesos de la aplicación móvil.

Figura 88

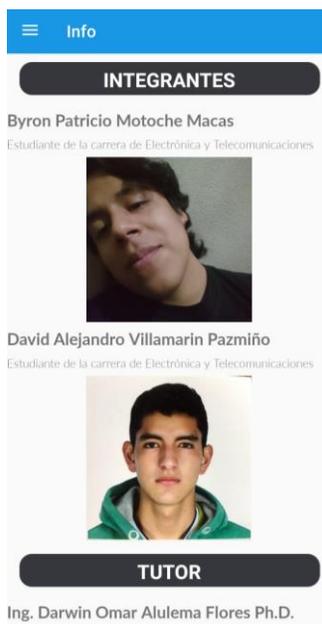
Contenido de la página Ayuda en la aplicación móvil



- **Info:** esta página ofrece información referente al proyecto.

Figura 89

Contenido de la página Info en la aplicación móvil



Adicionalmente existe la opción salir, donde se retorna hacia la pantalla de login y registro, reiniciando el flujo de la aplicación.

Capítulo IV

Resultados

Metodología

Métricas de los Modelos

Con el fin de evaluar los modelos entrenados de cada plataforma, se utilizó la matriz de confusión, la cual es una herramienta utilizada en el machine learning con el objetivo principal de visualizar el rendimiento de un modelo de aprendizaje. En la matriz de confusión las columnas indican el número de predicciones de cada clase, en nuestro caso de cada artículo, a

su vez las filas de la matriz de confusión indican los valores reales de cada clase. Los resultados obtenidos se pueden clasificar en las siguientes categorías:

- **Verdaderos positivos:** el modelo predijo positivo y la etiqueta fue realmente positiva.
- **Verdaderos negativos:** el modelo predijo lo negativo y la etiqueta fue realmente negativa.
- **Falsos positivos:** el modelo predijo positivo y la etiqueta fue en realidad negativa.
- **Falsos negativos:** el modelo predijo lo negativo y la etiqueta fue en realidad positiva.

Figura 90

Matriz de confusión 2x2

		VALOR PREDICHO	
		VERDADEROS POSITIVOS	FALSOS NEGATIVOS
VALOR REAL	VERDADEROS POSITIVOS	VERDADEROS POSITIVOS	FALSOS NEGATIVOS
	FALSOS POSITIVOS	FALSOS POSITIVOS	VERDADEROS NEGATIVOS

Con los valores obtenidos en la matriz de confusión se puede calcular fácilmente las métricas más comunes con las que se evalúan a los sistemas de NLP y son las siguientes:

- **Exactitud:** permite conocer la proximidad de un valor medido a un valor conocido, dicho de otra forma, es la relación entre el total de las predicciones correctas y el total de observaciones, se calcula como:

$$\text{Exactitud} = \frac{VP + VN}{VP + FP + VN + FN}$$

- **Precisión:** indica el número de predicciones marcadas como positivas por el modelo que son realmente positivas, en otras palabras, es la relación entre las predicciones positivas correctas (verdaderos positivos) y el total de observaciones positivas predichas, se calcula mediante:

$$\text{Precisión} = \frac{VP}{VP + FP}$$

- **Exhaustividad:** esta métrica indica el número de verdaderos positivos que nuestro modelo identifica correctamente como tales, es decir, es la relación entre las predicciones positivas correctas (verdaderos positivos) y el total de observaciones en la clase actual, se determina mediante:

$$\text{Recall} = \frac{VP}{VP + FN}$$

- **F1 score:** es una métrica que relaciona dos métricas complementarias (precisión y recall) en una sola métrica, se determina como:

$$\text{F1 Score} = \frac{2 * (\text{Precisión} * \text{Recall})}{\text{Precisión} + \text{Recall}}$$

Para evaluar los modelos de cada plataforma se utiliza una matriz de confusión de dimensión 16x16. En la cual se tiene un total de 16 clases (15 artículos y la clase "None"). Para determinar las métricas de evaluación para cada artículo se debe reducir la matriz de 16x16 a una matriz de 2x2 como la indicada anteriormente. Para lo cual se debe de tener en cuenta que la posición i x i son los verdaderos positivos, los falsos positivos es la suma de todos los valores

en la columna i a excepción del valor $i \times i$, los falsos negativos a su vez es la suma de todos los valores en la fila i a excepto el valor $i \times i$ y los verdaderos negativos se encuentran conformados por la suma de todos los valores a excepción en la fila i y columna i , por ejemplo en la figura que se muestra a continuación se observa cómo se reduce la matriz de confusión de 16x16 para el artículo 193 y se obtiene los valores para la matriz de 2x2.

Figura 91

Reducción de matriz de confusión 16x16 a 2x2

		ARTÍCULOS PREDICHOS POR EL MODELO															NONE																	
		ARTÍCULOS LOT					ARTÍCULOS COIP																											
		117	118	119	120	188	190	191	192	193	194	195	229	230	232	234																		
ARTÍCULOS ACTUALES	ARTÍCULOS LOT	117	VERDADEROS NEGATIVOS					FALSOS POSITIVOS										VERDADEROS NEGATIVOS																
		118																																
		119																																
		120																																
	ARTÍCULOS COIP	188	VERDADEROS NEGATIVOS					FALSOS POSITIVOS										VERDADEROS NEGATIVOS																
		190																																
		191																																
		192																																
		193																	FALSOS NEGATIVOS															FALSOS NEGATIVOS
		194																	VERDADEROS NEGATIVOS					FALSOS POSITIVOS										VERDADEROS NEGATIVOS
		195																																
		229																																
		230																																
		232																																
		234																																
		NONE																																

Para obtener las métricas de exactitud, precisión, exhaustividad y el F1 score de cada modelo en las plataformas de IBM Watson, Microsoft Azure y Amazon Web Services se utilizó un total de 330 ejemplos, 20 ejemplos por cada artículo y 30 ejemplos por la clase "None" y se obtuvo las matrices de confusión que se indican en la sección de resultados y posteriormente se calculó las métricas utilizando las fórmulas correspondientes.

Tiempo de respuesta

El tiempo de respuesta se refiere al intervalo de tiempo que tarda una aplicación o un servicio web en responder una solicitud enviada. El tiempo de respuesta puede variar dependiendo de diversos factores como el ancho de banda de la red, el volumen de usuarios, el número de solicitudes enviadas de forma simultánea, el tamaño de la solicitud e incluso el horario en el que se realiza la petición al servidor.

El tiempo de respuesta es un factor importante a tener en cuenta para brindar una buena experiencia al usuario. El valor límite aceptable de este parámetro depende de la aplicación, por ejemplo, para aplicaciones que requieren una respuesta en tiempo real como Streaming, videollamadas, etc el valor máximo aceptable es de 0.15 segundos. Por otro lado, para aplicaciones como navegación por internet se considera aceptables valores de hasta 1 segundo. Tiempos de respuesta mayores a 1 segundo son problemáticos, ya que el usuario podría abandonar el sitio web o la aplicación por completo.

Para el presente trabajo de investigación es de gran importancia medir el tiempo de respuesta de cada modelo de las plataformas en la nube utilizadas, IBM Watson, Microsoft Azure y Amazon Web Services. Para lo cual se ha tenido en cuenta las siguientes consideraciones:

- La longitud de la consulta se ha limitado a 150 caracteres (valor aproximado de las consultas por cada artículo entrenado en los diferentes modelos).
- Tres horarios de pruebas: 0-1 am, 10 am a 12pm y 6pm a 8pm.
- Una red con un ancho de banda aproximado de 90 Mbps.
- La medición se realizará utilizando el Software Postman.
- 80 consultas por modelo para cada horario establecido.

Pruebas de Carga y Esfuerzo

Las pruebas de carga y esfuerzo se encuentran estrechamente relacionadas, pero se diferencian principalmente en su objetivo. Las pruebas de carga tienen como objetivo principal validar el funcionamiento de un sistema web o una aplicación en un entorno controlado con un determinado número de solicitudes enviadas de forma simultánea. El resultado de las pruebas de carga garantiza que una aplicación o sistema web funcione sin mucha degradación bajo determinado nivel de tráfico.

Por otro lado, las pruebas de esfuerzo tienen como finalidad llevar al sistema web o aplicación al punto de falla o hasta que se agoten los recursos con una gran cantidad de peticiones de forma simultánea y observar cómo reacciona el sistema.

Existen diversas herramientas que permiten realizar pruebas de rendimiento, los cuales se pueden clasificar en dos grupos. El primero simula las transacciones a nivel de protocolo y consume pocos recursos de memoria, denominado HTTP-Level o Protocol-based, ejemplos de estas herramientas son JMeter y Postman. El otro grupo, simula las transacciones a nivel de navegador o mediante GUIs y consume mayor cantidad de recursos, pero ofrece mayor versatilidad en las pruebas, se destaca la herramienta LoadNinja.

Para la presente investigación es de suma importancia realizar pruebas de carga y pruebas de esfuerzo enfocadas al asistente de texto y a los servicios de procesamiento de lenguaje natural entrenados en las plataformas de Amazon Web Services, Microsoft Azure e IBM Watson con la finalidad de evaluar el rendimiento de la aplicación web a un determinado número de transacciones y también durante picos de solicitudes.

Se ha utilizado la herramienta de Postman para realizar las pruebas a nivel de protocolo de las API creadas para los servicios de NLP y del Asistente de Watson. Para lo cual se ha tenido en consideración un retardo entre cada petición de 100 ms (con la finalidad de simular el comportamiento de varios usuarios utilizando la aplicación de forma simultánea). Además, para que cada petición sea considerada exitosa se ha establecido el siguiente criterio: Respuesta del protocolo HTTP: 200 y un tiempo de respuesta menor a 500 ms, lo cual se especifica en Postman mediante el siguiente código:

```
pm.test("Código de estado 200 y Tiempo de Respuesta menor a 500 ms", function () {
    pm.response.to.have.status(200);
    pm.expect(pm.response.responseTime).to.be.below(500)
});
```

Para el escenario de pruebas de carga se ha establecido el número máximo de solicitudes simultáneas para los servicios de NLP en 15 y para el Asistente de Watson en 60 debido a que es el servicio que interactúa directamente con el usuario, adquiriendo como métricas el número de respuestas exitosas y fallidas y el tiempo medio de respuesta.

Por otro lado, para las pruebas de estrés se ha ido incrementando el número de peticiones simultáneas de forma progresiva hasta llegar a un total de 100 peticiones y

obteniendo como métricas el número de respuestas exitosas y fallidas, la tasa de error y el tiempo medio de respuesta.

Pruebas de Usabilidad

Las pruebas de Usabilidad tienen como objetivo medir la experiencia del usuario al interactuar con una aplicación o algún dispositivo. Uno de los sistemas más utilizados para este propósito es el SUS (del inglés, System Usability State), que consiste en una encuesta de 10 preguntas con cinco niveles de respuesta desde totalmente en desacuerdo hasta totalmente de acuerdo (Escala de Likert). Este sistema consta de 10 preguntas las cuales se encuentran detalladas en el Anexo 1

A través de las páginas de “Encuesta” tanto en la aplicación web como en la aplicación móvil, se ha recopilado esta información mediante el uso de encuestas, las cuales han sido almacenadas en la tabla Encuesta de la base de datos DB Aplicación, para el momento de los cálculos que se detallan, alrededor de 60 usuarios han respondido la encuesta, partiendo de estos datos se ha obtenido el valor medio de cada pregunta y se ha redondeado a enteros teniendo en cuenta 2 decimales para este proceso.

Pruebas Cuantitativas

Métricas de los Modelos

En la presente sección se detallan las matrices de confusión, las métricas de evaluación de sistemas de NLP y una comparativa de los resultados obtenidos de los modelos entrenados en las plataformas de AWS, Microsoft Azure e IBM.

Figura 92

Matriz de confusión del Modelo en AWS

		ARTÍCULOS PREDICHOS POR EL MODELO															NONE
		ARTÍCULOS LOT					ARTÍCULOS COIP										
		117	118	119	120	188	190	191	192	193	194	195	229	230	232	234	
ARTÍCULOS LOT	117	19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	118	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	119	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0
	120	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
ARTÍCULOS ACTUALES ARTÍCULOS COIP	188	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0
	190	0	0	0	0	0	19	0	0	0	0	1	0	0	0	0	0
	191	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0
	192	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0
	193	0	0	0	0	0	0	3	0	15	1	0	0	0	1	0	0
	194	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
	195	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0
	229	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0
	230	0	0	0	0	0	0	0	0	0	2	0	1	16	1	0	0
	232	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
234	0	0	0	0	1	0	0	0	0	0	0	0	0	0	19	0	
NONE	1	1	1	1	1	3	3	4	0	1	2	1	2	8	1	0	

Tabla 3*Métricas de cada artículo para el Modelo de AWS*

Artículo	Reales	Predichos	Exactitud	Precisión	Exahustividad	F1 Score
117	20	20	99.39%	0.95	0.95	0.95
118	20	21	99.70%	0.95	1	0.98
119	20	21	99.70%	0.95	1	0.98
120	20	22	99.39%	0.91	1	0.95
188	20	21	99.70%	0.95	1	0.98
190	20	22	98.79%	0.86	0.95	0.9
191	20	26	98.18%	0.77	1	0.87
192	20	24	98.79%	0.83	1	0.91
193	20	15	98.48%	1	0.75	0.86
194	20	24	98.79%	0.83	1	0.91
195	20	23	99.09%	0.87	1	0.93
229	20	22	99.39%	0.91	1	0.95
230	20	18	98.18%	0.89	0.8	0.84
232	20	31	96.67%	0.65	1	0.78
234	20	20	99.39%	0.95	0.95	0.95
None	30	0	90.91%	0	0	0
TOTAL	330	330	98.41%	0.8294	0.9000	0.8588

En base a la matriz de confusión y las métricas obtenidas para el modelo no supervisado en la plataforma de AWS se puede observar que existen predicciones erróneas (falsos negativos) para los artículos 117, 190, 193, 230, 234 y para la etiqueta None, lo que conlleva a tener valores de exhaustividad menores a 1 (caso ideal).

Al analizar los resultados del artículo 193 se evidencia que se presentan cinco falsos negativos que conllevan a tener una exhaustividad del 0.75 para esta categoría. Tres de los cinco falsos negativos están relacionados al artículo 191, estos dos artículos están relacionados con el reemplazo y modificación de información de identificación de celulares. Por otro lado, los cuatro falsos negativos implican una exhaustividad del 0.8 para el artículo 232. Dos de los cuatro falsos negativos corresponden al artículo 194 (comercialización ilícita de teléfonos), esto debido a que uno de los literales del artículo 230 está relacionado a la comercialización de información en dispositivos móviles.

Además, debido a que el modelo de AWS es del tipo no supervisado (no se realizan anotaciones de entidades que faciliten la clasificación del modelo) se ha obtenido varios falsos positivos para las categorías relacionadas a los artículos de textos que no tiene correspondencia con delitos informáticos y de las telecomunicaciones, lo que conlleva a tener valores de precisión menores a 1 en varias categorías de los artículos. Un ejemplo de esta problemática es el caso de la categoría relacionada con el artículo 232, en la cual se tienen ocho falsos positivos. Uno de estos falsos positivos es el texto “La ciencia es el gran antídoto contra el veneno del entusiasmo y la superstición” clasificado por el modelo de AWS como artículo 232 con un puntaje del 0.996.

Figura 93

Matriz de confusión del Modelo en Azure

		ARTÍCULOS PREDICHOS POR EL MODELO															NONE			
		ARTÍCULOS LOT					ARTÍCULOS COIP													
		117	118	119	120	188	190	191	192	193	194	195	229	230	232	234				
ARTÍCULOS ACTUALES	ARTÍCULOS LOT	117	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
		118	0	18	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
		119	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		120	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	ARTÍCULOS COIP	188	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	1
		190	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	1
		191	0	0	0	0	0	0	14	2	0	0	0	0	0	0	0	0	0	4
		192	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	1
		193	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	1	0
		194	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	1
		195	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0
		229	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
		230	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0
		232	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	1
234	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0		
NONE	2	1	1	1	1	1	1	0	1	0	0	0	0	1	10	5	5	5		

Tabla 4

Métricas de cada artículo para el Modelo de Microsoft Azure

Artículo	Reales	Predichos	Exactitud	Precisión	Exhaustividad	F1 Score
117	20	22	99.39%	0.91	1	0.95
118	20	19	99.09%	0.95	0.9	0.92
119	20	20	99.39%	0.95	0.95	0.95
120	20	20	99.39%	0.95	0.95	0.95
188	20	20	99.39%	0.95	0.95	0.95
190	20	20	99.39%	0.95	0.95	0.95

191	20	15	97.88%	0.93	0.7	0.8
192	20	21	99.09%	0.9	0.95	0.93
193	20	20	99.39%	0.95	0.95	0.95
194	20	19	99.70%	1	0.95	0.97
195	20	20	100%	1	1	1
229	20	20	100%	1	1	1
230	20	21	99.70%	0.95	1	0.98
232	20	30	96.36%	0.63	0.95	0.76
234	20	26	98.18%	0.77	1	0.87
None	30	17	88.79%	0.29	0.17	0.21
TOTAL	330	330	98.45%	0.8800	0.8981	0.8838

En base a las métricas presentadas y a la matriz de confusión del modelo supervisado en Microsoft Azure se puede destacar que se tiene falsos negativos por los artículos 118, 119, 120, 188, 190, 191, 192, 194 y 232, lo que implica tener valores de exhaustividad menores a 1. Estos falsos negativos en su mayoría han sido relacionados por el modelo con la Intención “none”.

El comportamiento antes mencionado puede ser debido a que el texto consultado a pesar de estar relacionado con el artículo no cumple la estructura de datos con la que se entrenó a los modelos o porque la conjugación del verbo no fue considerada durante el entrenamiento. Un ejemplo del caso descrito es el texto “Modificación de IMEI de dispositivos móviles” relacionado al artículo 191 reconocido por el modelo con la intención “none” debido a que carece de sujeto y complemento de lugar, como se entrenó al modelo.

Por otro lado, se puede observar que se tiene un alto número de falsos positivos por los artículos 232 y 234, acarreando precisiones del 0.63 y 0.77, respectivamente. Sin embargo, es importante mencionar que estos falsos positivos tienen puntajes demasiado bajos, incluso menores a 0.005 y podrían ser descartados utilizando una lógica de rechazo en base al puntaje retornado por el modelo.

Figura 94

Matriz de confusión del modelo en IBM Watson

		ARTÍCULOS PREDICHOS POR EL MODELO															NONE		
		ARTÍCULOS LOT					ARTÍCULOS COIP												
		117	118	119	120	188	190	191	192	193	194	195	229	230	232	234			
ARTÍCULOS ACTUALES	ARTÍCULOS LOT	117	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
		118	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
		119	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		120	0	2	0	17	0	0	0	0	0	0	0	0	0	0	0	0	1
		188	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	1
	ARTÍCULOS COIP	190	1	0	0	0	0	16	0	0	0	1	0	0	0	1	0	1	
		191	0	1	0	0	0	0	9	0	0	0	0	0	2	0	0	8	
		192	1	0	0	0	0	0	0	13	0	0	0	0	1	2	0	3	
		193	4	1	0	0	0	0	0	0	5	0	0	0	0	2	0	8	
		194	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	1	
		195	1	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	
		229	0	0	0	0	0	0	0	11	0	0	0	9	0	0	0	0	
		230	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	
		232	0	0	0	0	1	0	0	0	0	0	0	0	0	16	0	3	
		234	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	5	
		NONE	2	0	1	1	1	0	0	1	1	0	0	0	1	3	1	18	

Tabla 5

Métricas de cada artículo para el Modelo de IBM Watson

Artículo	Reales	Predichos	Exactitud	Precisión	Exhaustividad	F1 Score
117	20	27	96.67%	0.67	0.9	0.77
118	20	21	97.88%	0.81	0.85	0.83
119	20	20	99.39%	0.95	0.95	0.95
120	20	18	98.79%	0.94	0.85	0.89
188	20	21	99.09%	0.9	0.95	0.93
190	20	16	98.79%	1	0.8	0.89
191	20	9	96.67%	1	0.45	0.62
192	20	25	94.24%	0.52	0.65	0.58
193	20	6	95.15%	0.83	0.25	0.38
194	20	20	99.39%	0.95	0.95	0.95
195	20	19	99.70%	1	0.95	0.97
229	20	9	96.67%	1	0.45	0.62
230	20	24	98.79%	0.83	1	0.91
232	20	24	96.36%	0.67	0.8	0.73
234	20	16	98.18%	0.94	0.75	0.83
None	30	55	85.15%	0.33	0.6	0.42
TOTAL	330	330	96.93%	0.8338	0.7594	0.7669

Conforme a la matriz de confusión y a las métricas presentadas del modelo supervisado en IBM Watson se puede destacar que se tiene falsos negativos por todos los artículos a excepción del artículo 230. Por este motivo, solo se tiene un valor de exhaustividad igual a 1

correspondiente al artículo 230. Los falsos negativos presentes en su mayoría han sido relacionados por el modelo con la Intención “none”. Sin embargo, se presenta un caso particular correspondiente al artículo 229, en el que se ha tenido 11 falsos negativos etiquetados por el modelo con el artículo 192.

Al analizar los falsos negativos etiquetados por el modelo con la intención “none” se evidencia que son producto de las siguientes causas: el texto consultado a pesar de estar relacionado con el artículo no cumple la estructura de datos con la que se entrenó a los modelos o alguna de las palabras clave o verbos presenta tilde o carece de la misma. Un ejemplo del caso debido a la presencia/ausencia de tilde es el texto “Fabricio Aguilar y Byron Torres sustituyeron las etiquetas IMEI de terminales *móviles* en la ciudad de Paquisha.” relacionado al artículo 193 reconocido por el modelo con la intención “none” debido a la presencia de la tilde en la palabra móviles, al realizar la misma consulta sin la presencia de la tilde se obtiene la relación correspondiente al artículo 193 con una certeza del 0.9393.

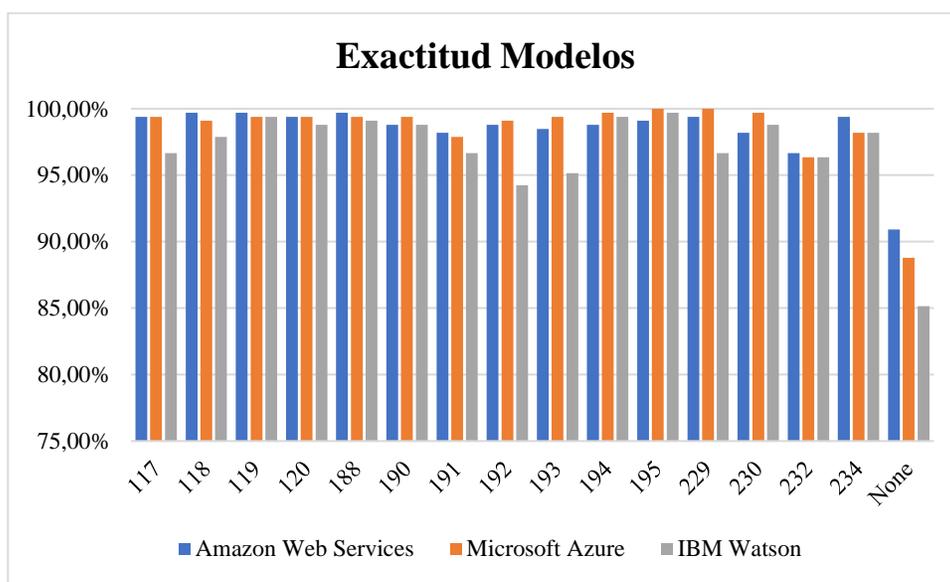
Respecto al caso particular, los 11 falsos negativos correspondientes al artículo 229 etiquetados por el modelo de IBM Watson con la relación del artículo 192 son debido a que durante el proceso de anotación de las relaciones para los textos de ejemplo del artículo 229 se etiquetaron algunos de forma errónea con la relación del artículo 192.

Teniendo en cuenta los falsos positivos, a excepción de los once del artículo 192 producto del error durante las anotaciones, se puede observar que se tiene de uno a tres falsos positivos de la etiqueta “none” predichos por el modelo con algunos de los artículos, lo que implica que la precisión disminuya para los mismos. Estos casos son para las consultas en las que se tienen adverbios de modo. Por ejemplo, al consultar el texto “El señor Lucas Correa accedió

de forma legal al sistema telemático de la Contraloría del Estado” el modelo lo relaciona con el artículo 234 con una certeza del 0.6742. Sin embargo, el delito es acceder **de forma ilegal**. Estos resultados podrían ser descartados utilizando una lógica de rechazo en base al puntaje retornado por el modelo.

Figura 95

Comparativa de la exactitud de los Modelos

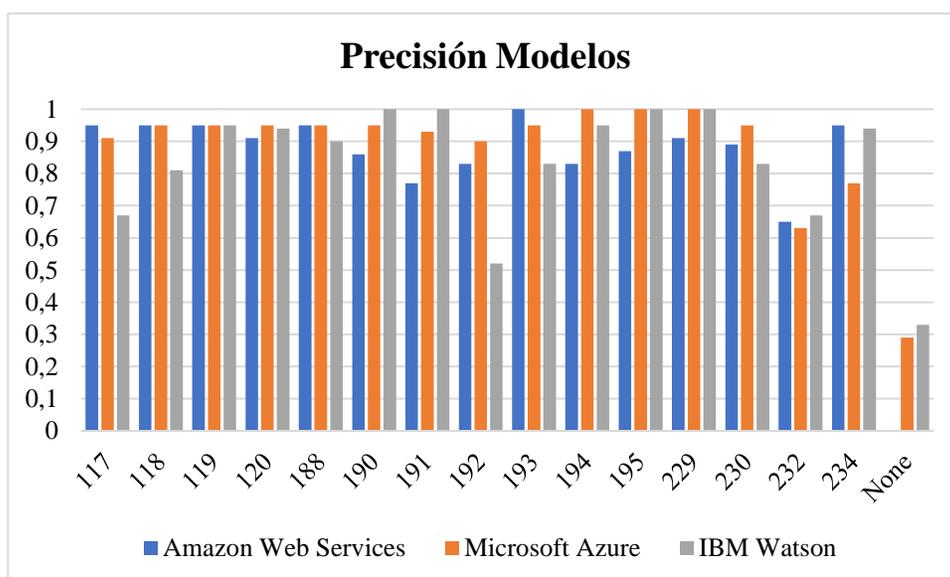


El modelo entrenado con la mayor exactitud es el modelo de Microsoft Azure con una exactitud promedio del 98.45%. En segundo lugar, por una pequeña diferencia en el orden de los decimales se encuentra el modelo de AWS con 98.41% y al final IBM Watson con 96.93%. Es importante notar que al reducir la matriz de confusión a una matriz de 2x2 el número de verdaderos positivos, falsos positivos y falsos negativos es mucho menor al número de verdaderos negativos. Por esta razón, la exactitud no varía en gran medida cuando se reduce el número de verdaderos positivos como en el caso de la etiqueta “none” del modelo AWS que a

pesar de tener cero verdaderos positivos se obtiene una exactitud mayor al 90%, lo cual podría ser interpretado de forma incorrecta. Por lo tanto, se debe analizar los valores de precisión, exhaustividad y el F1 Score para comparar el rendimiento de los modelos.

Figura 96

Comparativa de la precisión de los Modelos

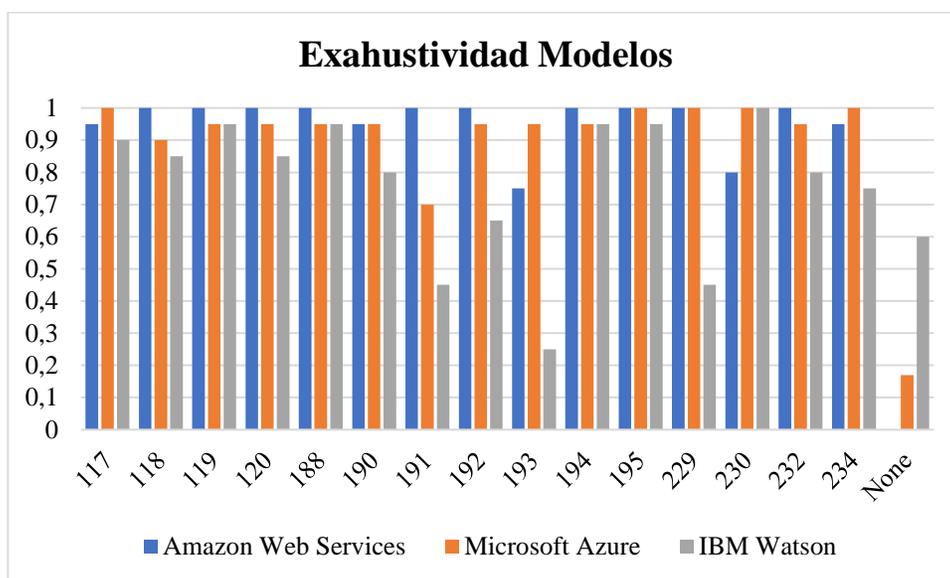


El modelo con la métrica más alta de precisión es el de Microsoft Azure con un promedio de 0.88. En segundo lugar, se encuentra el modelo de IBM Watson con una precisión del 0.8338 y le sigue muy de cerca el modelo de AWS con 0.8294. Estos valores se pueden interpretar de la siguiente forma: de 100 textos que han sido predichos por los modelos como delitos relacionados al artículo 230, para el modelo de Microsoft Azure 88 si están relacionados al artículo 230 y los otros 12 son producto de falsos positivos. De forma similar, para los

modelos de AWS e IBM Watson 82 y 83 de los textos predichos si corresponden al artículo 230 y 18 y 17 son resultado de falsos positivos.

Figura 97

Comparativa de la exhaustividad de los Modelos

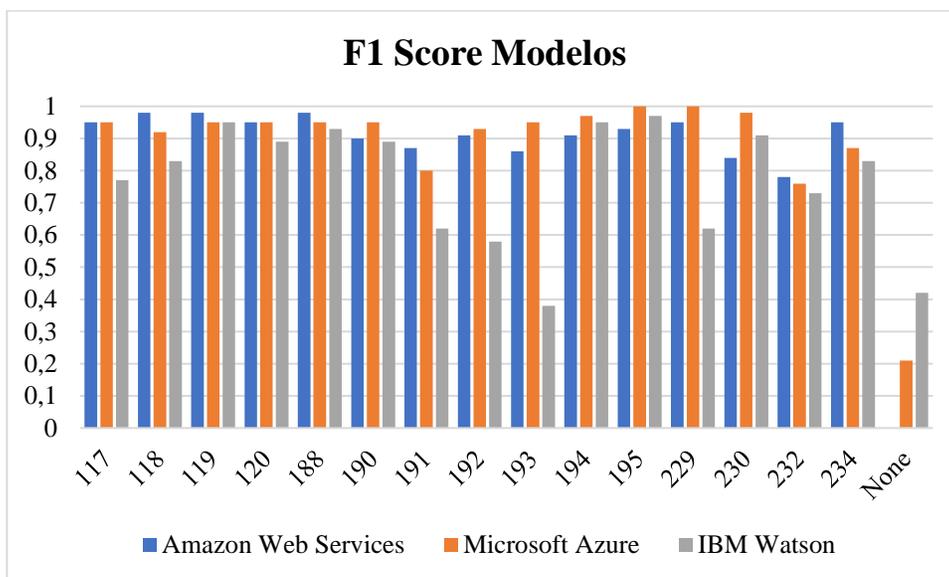


Al observar los resultados de la exhaustividad de los modelos entrenados es evidente de forma gráfica que los modelos de AWS y Microsoft Azure presentan mejores resultados en comparación al modelo de IBM Watson; numéricamente el que mejor puntuación de exhaustividad presenta es el modelo de AWS con un promedio de 0.9, le sigue el modelo entrenado en Microsoft Azure con un valor de 0.8981 e IBM Watson con un valor de 0.7594. Ejemplificando, de 100 textos con delitos informáticos AWS identificará 90 textos de forma correcta con que artículos están relacionados, Microsoft Azure por su parte identificará de

forma correcta 89 textos e IBM Watson 75, aquellos valores no identificados de forma correcta resultan en falsos negativos.

Figura 98

Comparativa del F1 Score de los Modelos



Respecto al F1 Score, como se indicó anteriormente permite combinar las medidas de precisión y exhaustividad en una sola medida y facilitar la comparación del rendimiento entre diferentes modelos. Se evidencia gráficamente que los modelos entrenados en AWS y Microsoft Azure tienen un mejor puntaje en la mayoría de artículos con respecto al modelo de IBM Watson. Sin embargo, el modelo en IBM Watson presenta un puntaje de F1 mayor que los otros modelos al tratarse de textos no relacionados o aquellos clasificados con la etiqueta None, dicho de otra forma, no genera tantos falsos positivos con textos que no se encuentran relacionados con delitos informáticos como lo hacen los modelos de AWS y Microsoft Azure. Al obtener el promedio de los puntajes F1 de cada modelo se obtiene que el modelo entrenado en la

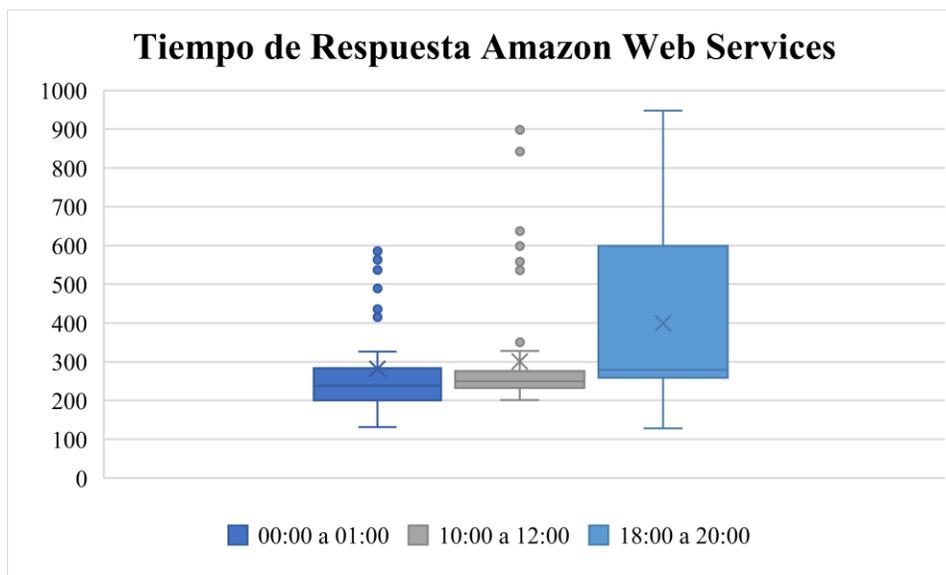
plataforma de Microsoft Azure presenta el mayor puntaje F1 con un promedio de 0.8838, seguido por el modelo en AWS con un valor de 0.8588 y al final se encuentra el modelo de IBM Watson con un valor de 0.7669.

Tiempo de Respuesta

A continuación, se presentan los resultados referentes a los tiempos de respuesta obtenidos en los horarios de pruebas planteados para los servicios de AWS Comprehend, Microsoft Azure LUIS e IBM Watson NLU mediante diagramas de caja y bigotes para analizar su distribución.

Figura 99

Comparativa tiempo de respuesta del modelo en AWS en horarios de prueba



Se puede observar de la figura que para el servicio de Comprehend de AWS los resultados obtenidos si varían dependiendo del horario en el que se ejecuta la solicitud al servicio.

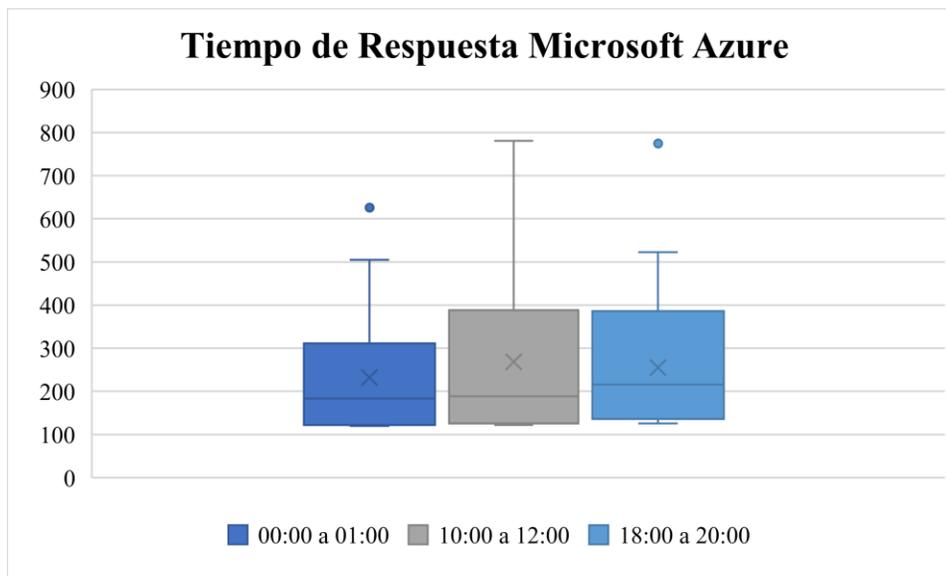
Para el primer horario, de 00:00 a 01:00, el valor mínimo del tiempo de respuesta es de 131 ms, el valor del primer quintil es de 200 ms, la mediana tiene un valor de 237 ms, el valor del tercer cuartil es de 283 ms, el valor máximo es de 326 ms, el valor promedio es de 281.95 ms y se presentan valores atípicos a partir de los 400 ms hasta los 585 ms.

En el segundo horario, de 10:00 a 12:00, los valores obtenidos se encuentran menos dispersos en comparación con los otros horarios. Sin embargo, se presentan varios casos extremos que están muy distantes de la mediana. El valor mínimo obtenido es de 200 ms, mientras que el valor máximo es de 342 ms. El 50% de las mediciones obtenidas se encuentran entre el Q_1 de 232 ms y el Q_3 de 276 ms. Es importante destacar que debido a los altos valores atípicos obtenidos el valor promedio de 300.87 ms se encuentra fuera del rango Intercuartil.

Durante el tercer horario, de 18:00 a 20:00, se evidencia que los resultados se encuentran más dispersos respecto a los otros escenarios. A pesar de que durante este horario se ha obtenido el valor máximo más alto en comparación a los otros escenarios, con un valor de 948 ms, también se ha presentado el menor valor mínimo en las mediciones con 128 ms. Además, se observa que el rango intercuartil, de Q_1 259 ms a Q_3 597 ms, que abarca el 50% de las mediciones es más mucho más amplio que en los otros horarios y el tiempo medio de respuesta es de 399.46 ms.

Figura 100

Comparativa tiempo de respuesta del modelo en Microsoft Azure en horarios de prueba



De acuerdo a la figura presentada se evidencia que los resultados obtenidos del tiempo de respuesta para el servicio de Microsoft Azure LUIS son semejantes durante los tres horarios. Por tanto, se puede inferir que el tiempo de respuesta al consumir este servicio es independiente del horario en el que se realiza la consulta.

Durante el primer horario se encuentra el menor de los valores mínimos y el menor de los valores máximos con tiempos de respuesta de 120 y 505 ms, respectivamente. Si bien las mediciones obtenidas durante los tres horarios son semejantes, el rango intercuartil del primer horario es más angosto respecto a los otros horarios. Se tiene un valor medio de respuesta de 232.39 ms y un valor atípico de 626 ms.

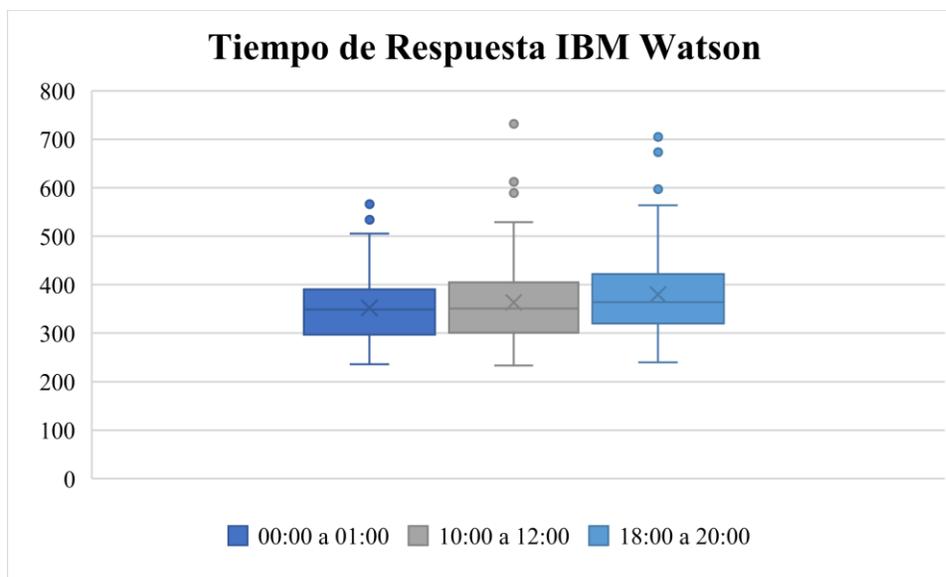
En el horario de 10:00 a 12:00, el bigote superior es más ancho en comparación a los otros horarios debido a que los valores del Q_3 y el rango intercuartil son mayores durante este horario. Además, no se presentan valores atípicos debido a que se tiene un bigote superior más

ancho y el valor máximo de 781 ms. El tiempo de respuesta medio durante este horario es de 268.28 ms.

Para el tercer horario de 18:00 a 20:00, el rango intercuartil es semejante al del segundo horario. Sin embargo, el primer 25% de los resultados (desde el valor mínimo a Q_1) se encuentran más dispersos respecto a los otros dos horarios y por el contrario el último 25% de los resultados (desde Q_3 al valor máximo) se encuentran menos dispersos, provocando un bigote superior más angosto. Durante el horario de 18:00 a 20:00 se obtiene un tiempo medio de respuesta de 254.72 ms y un valor atípico de 774 ms.

Figura 101

Comparativa tiempo de respuesta del modelo en IBM Watson en horarios de prueba



En base a resultados presentados en los diagramas de caja y bigotes por los tres horarios de pruebas establecidos se verifica que el tiempo de respuesta para el servicio de IBM Watson es independiente del horario en el que se realiza la consulta.

En el horario de 00:00 a 01:00, se tiene valores mínimos y máximos con tiempos de respuesta de 236 y 505 ms, respectivamente. El valor promedio del tiempo de respuesta de 352.31 ms se encuentra muy cercano a la mediana con valor de 346 ms. Como valores atípicos se tiene los valores de 534 y 566 ms. El rango intercuartil es semejante a los otros dos horarios y tiene un valor de 91 ms.

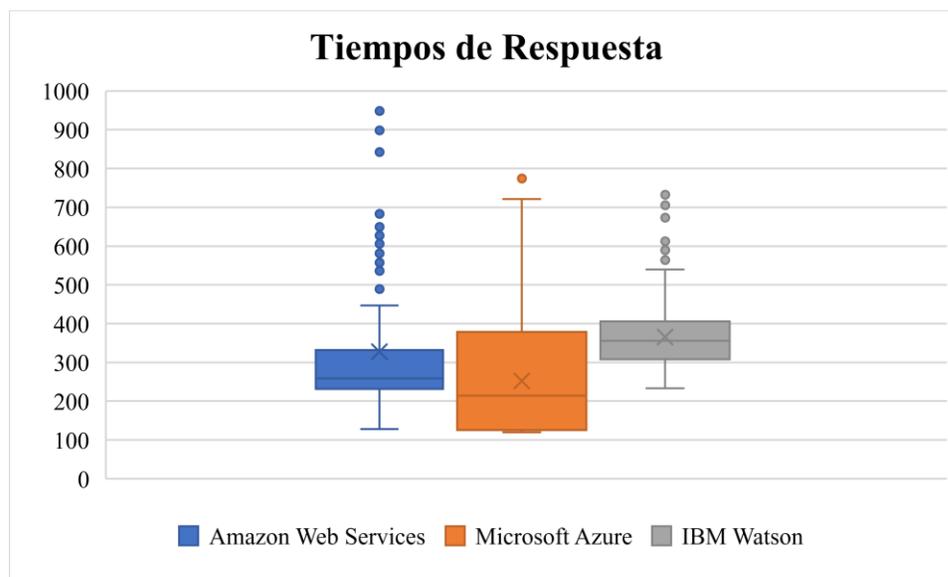
Durante el segundo horario, la media de 363.45 ms se encuentra un poco más distante de la mediana a 350 ms, con respecto al primer horario. Para este horario se tiene un total de tres valores extremos que se encuentran fuera del bigote superior y son 589, 612 y 732 ms.

En el horario de 18:00 a 20:00, los resultados del 75% en adelante se encuentran más dispersos respecto a los otros dos horarios debido a que el bigote superior es ligeramente más largo. El tiempo medio de respuesta es de 363.45 ms y el valor mínimo de respuesta es de 240ms. El rango intercuartil tiene un valor similar a los otros dos horarios y es de 102 ms. Adicionalmente, se presentan tres valores extremos superiores al valor máximo de 564 ms.

A continuación, se expone una comparativa de los tiempos de respuesta de los servicios de AWS Comprehend, Microsoft Azure LUIS e IBM Watson NLU tomando en consideración los tres horarios planteados en conjunto a través de diagramas de caja y bigotes para analizar su dispersión y comportamiento.

Figura 102

Comparativa tiempos de respuesta de los tres modelos



Al analizar los valores mínimos y máximos del tiempo de respuesta se observa que el menor valor mínimo obtenido corresponde al servicio de Microsoft Azure LUIS con 120 ms, le sigue el servicio de AWS Comprehend con 128 ms y al final se encuentra el servicio de IBM Watson NLU con 233 ms. El servicio con el menor valor máximo de tiempo de respuesta es AWS Comprehend con 447 ms, seguido de IBM Watson NLU con 540 ms y Microsoft Azure LUIS con 721 ms.

De forma similar, al analizar el valor medio del tiempo de respuesta de cada servicio se evidencia que el servicio con el menor tiempo medio de respuesta es Microsoft Azure LUIS con 251.8 ms, seguido de AWS Comprehend con 327.43 ms y al final se encuentra IBM Watson NLU con 365.15 ms.

Por otro lado, teniendo en cuenta las longitudes de los bigotes inferiores y superiores se observa que las mediciones obtenidas para Microsoft Azure LUIS se encuentran mucho más dispersas en el bigote superior en comparación a los servicios de IBM Watson NLU y AWS Comprehend; lo contrario ocurre con los bigotes inferiores en donde los tiempos de respuesta se encuentran más dispersos en IBM Watson NLU y AWS Comprehend en comparación a Microsoft Azure LUIS donde el 25% de las mediciones se encuentran concentradas entre los valores de 120 ms y 126 ms.

De igual forma, al valorar los rangos intercuartiles donde se encuentra el 50% de los tiempos de respuesta se revela que para Microsoft Azure LUIS estos se encuentran más dispersos que para los otros servicios de NLP, pudiendo variar desde $Q_1 = 126$ ms hasta $Q_3 = 379$ ms.

Otro aspecto importante a tener en consideración son los valores atípicos, donde se evidencia que en AWS Comprehend se tiene varios valores fuera del bigote superior llegando hasta 948 ms. De manera similar ocurre en IBM Watson NLU y Microsoft Azure LUIS, donde se presentan valores extremos de hasta 732 ms y 774 ms, respectivamente.

Pruebas de Carga y Esfuerzo

En la siguiente tabla se presentan los resultados de las pruebas de carga con las métricas establecidas previamente para los servicios de IBM Watson Assistant, AWS Comprehend, Microsoft Azure LUIS e IBM Watson NLU.

Tabla 6

Resultados de las pruebas de carga de los servicios de NLP y el Asistente de Watson

Servicio	Peticiones	Respuestas	Tiempo de Respuesta
	Simultáneas	Exitosas/No Exitosas	Promedio [ms]
IBM Watson Assistant	60	60/0	221.17
AWS Comprehend	15	15/0	178
Microsoft Azure LUIS	15	15/0	170.73
IBM Watson NLU	15	15/0	440.87

En base a los resultados obtenidos de las pruebas de carga se verifica que la aplicación Telegal puede soportar de forma simultánea 15 peticiones hacia los servicios de NLP (AWS Comprehend, Microsoft Azure LUIS e IBM Watson NLU) y hasta 60 transacciones simultáneas con el Asistente de Watson, sin degradaciones en los tiempos de respuesta. Por tanto, los resultados de las pruebas de carga para los servicios evaluados son satisfactorios en su totalidad. Cabe destacar que el servicio de NLP con menor tiempo de respuesta es el servicio de Microsoft Azure LUIS debido a la poca complejidad del consumo del servicio en comparación con AWS Comprehend e IBM Watson NLU.

A continuación, se detallan los resultados de las pruebas de esfuerzo para los servicios de IBM Watson Assistant, AWS Comprehend, Microsoft Azure LUIS e IBM Watson NLU.

Tabla 7

Resultados de las pruebas de esfuerzo de los servicios de NLP y el Asistente de Watson

Respuestas Exitosas / No Exitosas								
Peticiones Simultáneas	IBM Watson	AWS	Microsoft			IBM Watson		
	Assistant	Comprehend	Azure	LUIS	NLU			
15	15	0	15	0	15	0	15	0
20	20	0	20	0	20	0	20	0
25	25	0	25	0	25	0	25	0
30	30	0	30	0	30	0	30	0
40	40	0	40	0	40	0	40	0
50	50	0	50	0	50	0	50	0
60	60	0	58	2	60	0	59	1
70	70	0	60	10	70	0	58	12
80	80	0	63	17	80	0	73	7
90	90	0	65	25	86	4	80	10
100	100	0	71	29	98	2	93	7

De los resultados obtenidos en las pruebas de esfuerzo se puede observar que los servicios de AWS Comprehend e IBM Watson NLU son los primeros servicios en presentar una respuesta no exitosa a partir de 60 peticiones simultáneas. El servicio de Microsoft Azure LUIS

presenta respuestas no exitosas a partir de 90 peticiones simultáneas. Por otro lado, en el servicio de IBM Watson Assistant no se evidencia respuestas no exitosas incluso cuando se tiene hasta 100 peticiones de forma simultánea.

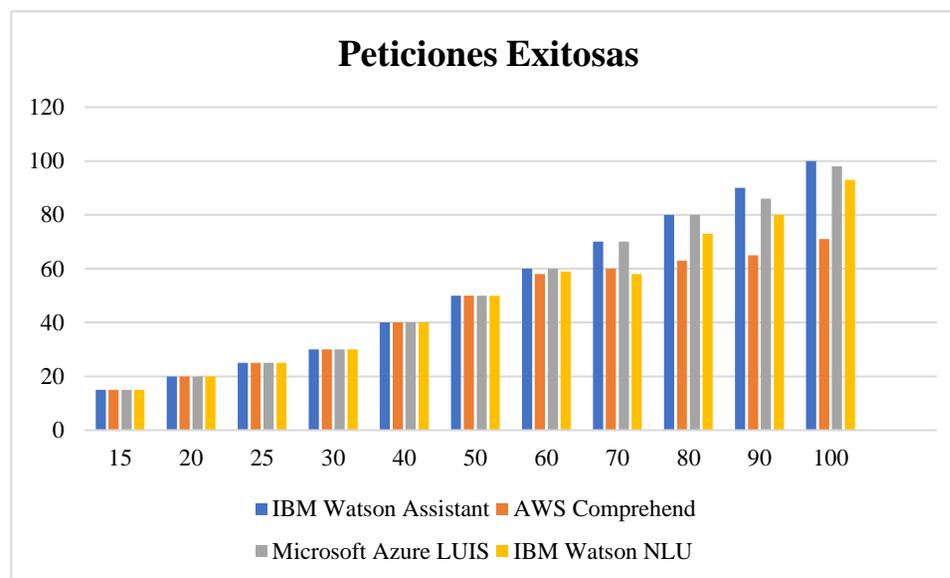
Para el servicio de AWS Comprehend se obtienen tasas de error desde el 3.33% (con 60 peticiones simultáneas) hasta el 29% (con 100 peticiones simultáneas). En el servicio de IBM Watson NLU las tasas de error para 60 y 100 peticiones simultáneas son del 1.67% y 7%, respectivamente. Mientras que la tasa de error para 100 peticiones simultáneas utilizando el servicio de Microsoft Azure es del 2%.

Es importante mencionar sobre las respuestas no exitosas que a pesar de que se obtuvo una respuesta a nivel del protocolo HTTP del tipo 200, los resultados se consideran no exitosos debido a que el tiempo de respuesta es mayor a 500 ms.

A continuación, se presentan diagramas de barras con las peticiones exitosas y no exitosas obtenidas de las pruebas de esfuerzo de los servicios evaluados.

Figura 103

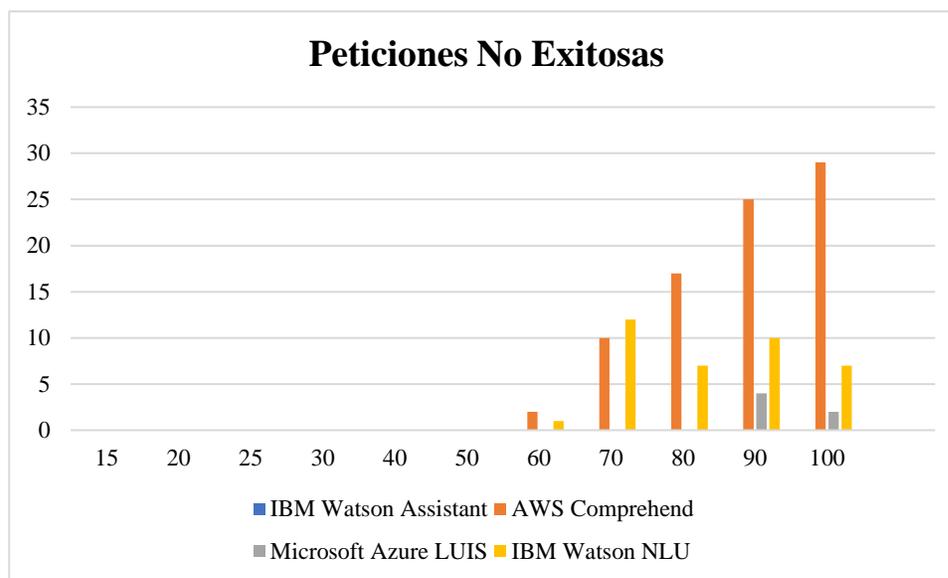
Peticiones simultáneas exitosas de los servicios de NLP y el Asistente de Watson



En base al diagrama se evidencia que los servicios no presentan ninguna degradación cuando se realizan hasta 50 peticiones de forma simultánea. Sin embargo, al sobrepasar este umbral se empieza a tener una degradación notoria en los servicios de AWS Comprehend e IBM Watson NLU. Cabe destacar que el servicio de IBM Watson Assistant no presenta ninguna degradación cuando se ejecutan hasta 100 transacciones simultáneas. Por otro lado, al servicio de Microsoft Azure LUIS se puede realizar hasta un total de 80 peticiones de forma simultánea sin presentar degradación.

Figura 104

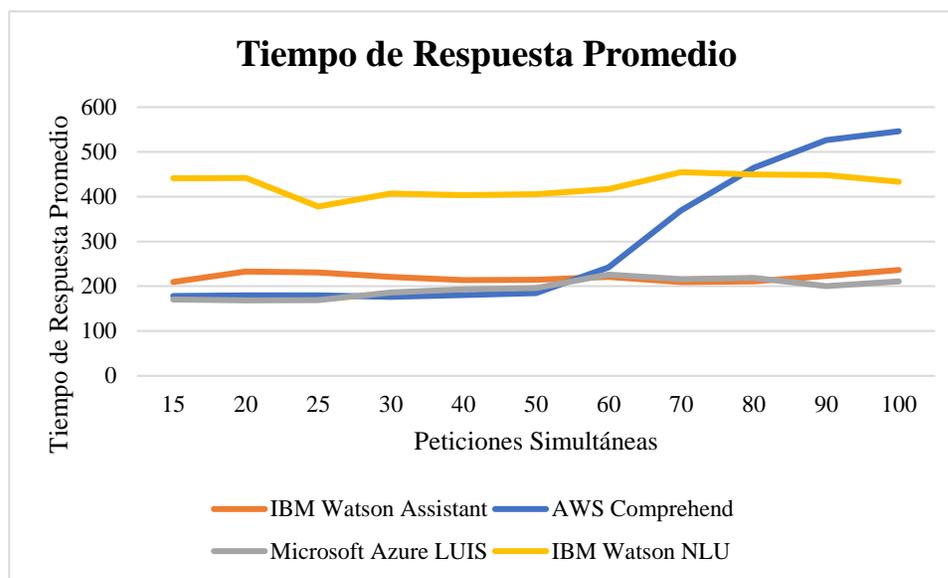
Peticiones no exitosas de los servicios de NLP y el Asistente de Watson



En base a la figura se observa que al incrementar el número de peticiones simultáneas al servicio de AWS Comprehend por encima de las 60 transacciones, el número de peticiones no exitosas aumenta con una tendencia lineal. Por otro lado, la tendencia de aumento de las peticiones no exitosas de los servicios de Microsoft Azure LUIS e IBM Watson NLU no es lineal.

Figura 105

Tiempo de respuesta promedio durante las pruebas de esfuerzo



De acuerdo a la figura se valida que existe un incremento notorio en el tiempo de respuesta promedio del servicio de AWS Comprehend al realizar más de 50 peticiones de forma simultánea, pasando de una media aproximada de 200 ms en 50 solicitudes a un valor de 546 ms en 100 solicitudes. Por otro lado, el tiempo de respuesta promedio de los servicios de IBM Watson Assistant y Microsoft Azure LUIS mantiene una tendencia menor a los 250 ms. El servicio de IBM Watson NLU por su parte presenta tiempos de respuesta promedios superiores a los 400ms y menores a 500ms.

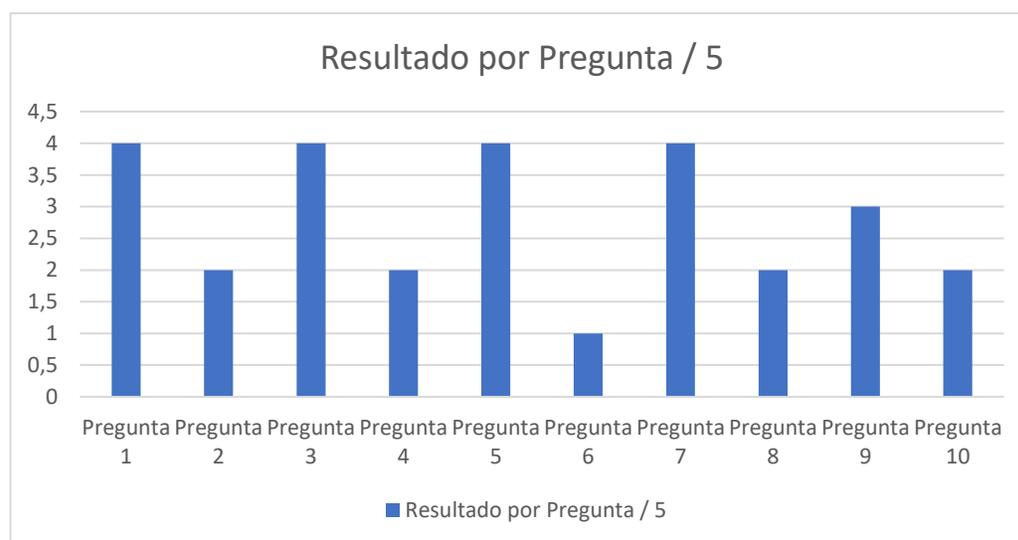
Es importante señalar que en base a los resultados antes mencionados el servicio de Microsoft Azure LUIS es el servicio de NLP con menor degradación en el tiempo de respuesta y menor número de peticiones no exitosas al incrementar la cantidad de transacciones simultáneas.

Pruebas Cualitativas

A continuación, se presenta una gráfica de barras con los resultados obtenidos en las encuestas para cada pregunta.

Figura 106

Resultado Encuestas SUS



Para el cálculo serán necesarios dos valores, los cuales se sumarán y al resultante se multiplicará por 2.5, el primer valor se obtiene de las preguntas impares, se sumarán los puntos y al resultado se substraen 5. El segundo valor es obtenido de las preguntas pares, restando su suma de 25.

$$\text{Valor 1} = \text{Preguntas impares} = 4 + 4 + 4 + 4 + 3 = 19$$

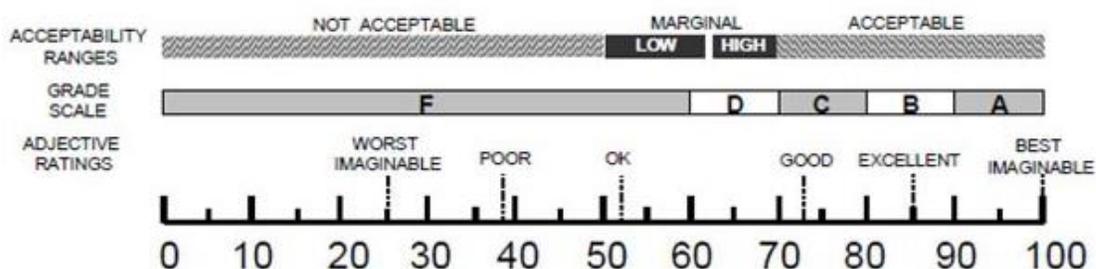
$$\text{Valor 2} = \text{Preguntas pares} = 25 - (2 + 2 + 1 + 2 + 2) = 16$$

$$\text{El puntaje SUS} = (\text{Valor 1} + \text{Valor 2}) (2.5) = (19+16) (2.5) = 75 \text{ SUS}$$

Es importante señalar que si bien el valor teórico total posible es de 100, el valor SUS no representa un porcentaje o un percentil, existe un valor promedio de 68 SUS, de forma sencilla al obtener puntuaciones inferiores o superiores al valor promedio, brinda información necesario sobre la usabilidad general actual del sistema (Smyk, 2020).

Figura 107

Representación de los resultados de un SUS



Nota. Se presentan los diferentes rangos de aceptabilidad, escalas y adjetivos de calificación de los resultados SUS, (Montiel Anguiano, 2017).

En base a la figura previa, se puede determinar que la aplicación Telegal se encuentra en un rango aceptable en cuanto a valor SUS se refiere, sin embargo, existen puntos de mejora referente a experiencia de usuario, para lograr una puntuación excelente o mejor posible (entre 85y 100). La escala de calificación coloca a la aplicación en una C con un "Bueno" como adjetivo de clasificación. Además, es posible medir la efectividad que han tenido los usuarios

completando tareas y logrando los objetivos de consulta, eficiencia en el sentido del uso de diferentes recursos usados para el logro de los objetivos y satisfacción que permite conocer la comodidad que experimentaron los usuarios alcanzando sus objetivos.

Cabe señalar que pese a que el valor SUS es un valor referencial que indica la experiencia de usuario percibida en general por el usuario, no brinda información adicional y complementaria del mismo, es decir no es sencillo determinar cuáles son los puntos a tener en cuenta, simplemente indica un valor promedio, así mismo, los resultados están expuestos a la subjetividad y percepción de cada uno de los usuarios que ha respondido de la encuesta.

Adicionalmente a las 10 preguntas usadas para el cálculo del valor del SUS previamente realizado, en la encuesta fueron también planteadas tres preguntas:

1. Creo que el asistente de texto fue fácil de usar.
2. Creo que el asistente de voz fue fácil de usar.
3. La predicción realizada por el asistente fue útil.

Estas preguntas son relevantes para el proyecto ya que permiten conocer cuáles son las impresiones generales de los usuarios de la aplicación tanto móvil como web referente a los asistentes de texto, voz y a la predicción como tal, siguiendo la misma lógica y el mismo número de encuestas respondidas que en el caso anterior, se pudo determinar el valor de las mismas y es:

1. Creo que el asistente de texto fue fácil de usar: 4 puntos
2. Creo que el asistente de voz fue fácil de usar: 2 puntos.
3. La predicción realizada por el asistente fue útil: 4 puntos.

De esta forma resulta sencillo determinar que, para los usuarios el asistente de texto resultó mucho más fácil de usar frente al asistente de voz, y de manera complementaria la predicción realizada por cualquiera de estos asistentes resultó de utilidad para la mayoría de usuarios.

Capítulo V

Conclusiones y Trabajos Futuros

Conclusiones

Se diseñó un sistema de análisis de textos el cual hace uso de servicios en la nube de tres plataformas (Amazon Web Services, Microsoft Azure, IBM Watson) relacionados con procesamiento de lenguaje natural, el sistema está enfocado en determinar delitos de las telecomunicaciones en el Ecuador teniendo en cuenta 11 artículos del Código Integral Penal y 4 artículos de la Ley Orgánica de Telecomunicaciones. Para determinar el rendimiento y usabilidad general del sistema se realizaron pruebas tanto cuantitativas como cualitativas.

La arquitectura del sistema implementada está orientada a servicios, donde se integran los modelos de cada plataforma, las bases de datos de artículos y la aplicación web y móvil, se evaluó diversos parámetros de los modelos entre los cuales se encuentran las métricas de exactitud, precisión, exhaustividad y F1 score. Debido a que el número de verdaderos positivos, falsos positivos y falsos negativos es mucho menor al número de verdaderos negativos la exactitud arroja resultados no fiables para comparar los modelos entrenados. Por lo tanto, se debe analizar las métricas de precisión, exhaustividad y el F1 Score para evaluar el desempeño de los modelos.

En base a la métrica de precisión que evalúa los verdaderos positivos respecto a los falsos positivos el modelo entrenado en Microsoft Azure presenta los mejores resultados con un valor de 0.88. En segundo lugar, se encuentra el modelo de IBM Watson con una precisión del 0.8338 y al final esta modelo de AWS con 0.8294. Se concluye que el modelo de Microsoft Azure es el que menos falsos positivos genera.

Por otro lado, al analizar la exhaustividad que evalúa la relación entre los verdaderos positivos y falsos negativos. Se evidencia que el modelo con mejor puntuación respecto a esta métrica es el modelo de AWS con un valor de 0.90. Seguido por el modelo entrenado en Microsoft Azure con una exhaustividad de 0.8981 y en tercer lugar el modelo de IBM Watson con un valor de 0.7594. Por lo tanto, el modelo de AWS presenta una menor cantidad de falsos negativos.

Finalmente, al analizar el F1 Score que permite evaluar diferentes modelos de machine learning con una sola métrica se obtiene que Microsoft Azure es el modelo con mayor puntaje, con un valor de 0.8838. A continuación, se encuentra el modelo de AWS con un valor de 0.8588 y finalmente se posiciona el modelo de IBM Watson con un valor de 0.7669. En consecuencia, el modelo de Microsoft Azure presenta la mejor relación entre precisión y exhaustividad en comparación a los modelos de IBM Watson y AWS.

Se analizó también los tiempos de respuestas de cada uno de los servicios en tres horarios diferentes (madrugada, mediodía y tarde). De los resultados obtenidos se evidencia que existe un tiempo de respuesta promedio de 399.46 ms en las peticiones realizadas en el horario de 18:00 a 20:00 para el caso de AWS Comprehend y se tiene un mayor rango de variación respecto a otros escenarios. Por otro lado, los servicios en IBM Watson NLU y

Microsoft Azure LUIS no presentan variación respecto a los tiempos de respuesta en los diferentes horarios planteados.

Al evaluar el tiempo medio de respuesta de los tres servicios de forma general, se concluye que el servicio con el menor tiempo medio de respuesta corresponde a Microsoft Azure LUIS con 251.8 ms. A continuación, se encuentra el servicio de AWS Comprehend con 327.43 ms y al final, con el mayor tiempo de respuesta, se encuentra IBM Watson NLU con 365.15 ms.

Además, se realizaron pruebas de carga y esfuerzo para los servicios de procesamiento de lenguaje natural de cada plataforma, así como para el servicio de IBM Watson Assistant debido a que es el servicio que interactúa directamente con el usuario. En base a los resultados obtenidos, se comprueba que la aplicación Telegal puede soportar de forma simultánea hasta 15 peticiones hacia los servicios de NLP (AWS Comprehend, Microsoft Azure LUIS e IBM Watson NLU) y hasta 60 transacciones simultáneas con el Asistente de Watson, sin degradaciones en los tiempos de respuesta. Por consiguiente, los resultados de las pruebas de carga para los servicios evaluados son satisfactorios en su totalidad.

Respecto a las pruebas de esfuerzo, se evidencia que los servicios no presentan ninguna degradación cuando se realizan hasta 50 peticiones de forma simultánea. Sin embargo, al sobrepasar este umbral la degradación es notoria en los servicios de AWS Comprehend e IBM Watson NLU. Cabe destacar que el servicio de IBM Watson Assistant no presenta ninguna degradación cuando se ejecutan hasta 100 transacciones simultáneas. Por otro lado, al servicio de Microsoft Azure LUIS se puede realizar hasta un total de 80 peticiones de forma simultánea sin presentar degradación.

Teniendo en consideración las métricas más comunes con las que se evalúan a los sistemas de NLP, el tiempo de respuesta promedio, la cantidad de transacciones simultáneas soportadas se evidencia que la plataforma de Microsoft Azure presenta mejores resultados en comparación a las plataformas de IBM y Amazon Web Services.

Para evaluar la usabilidad del sistema han sido realizadas encuestas siguiendo el Sistema de Escalas de Usabilidad (SUS), una vez calculado el valor SUS se pudo determinar que la aplicación Telegal se encuentra dentro del rango aceptable con una puntuación de 75 siendo calificada como “Buena”, lo cual significa que de forma general ha sido satisfactoria para la mayoría de usuarios, de igual forma fueron realizadas preguntas complementarias en la encuesta que permiten determinar que el asistente de texto es más sencillo de usar que el asistente de voz y que las predicciones realizadas por los modelos han resultado de utilidad.

Se crearon modelos en cada una de las plataformas siendo dos del tipo supervisados (Microsoft Azure e IBM Watson) y uno del tipo no supervisado (Amazon Web Services), cada modelo fue entrenado con el mismo corpus de información donde ha sido integrado registros de procesos judiciales disponibles en la página web de la función judicial, el detalle de los artículos del COIP y la LOT y diversas noticias obtenidas de medios de información digital del país.

A través del uso de un webhook dentro del servicio Watson Assistant es posible integrar los modelos de machine learning, el Web API de la base de datos de artículos, así como también los servicios de Speech to Text y Text to Speech.

Dentro de la aplicación Telegal han sido usadas diversos frameworks y herramientas como: ASP.NET, Web API, Xamarin, Windows Communication Foundation (WCF), Django y

HTML5, así como también algunos lenguajes de programación entre los cuales están: C#, Visual Basic, Python, Node JS y JavaScript.

Una arquitectura orientada a servicios permite usar diversas tecnologías y lenguajes de programación en cada uno de sus módulos, las cuales tienen un comportamiento independiente y aislado, a través del uso de un formato de intercambio de datos como JSON, es posible concatenar y comunicar cada módulo entre sí, usando servicios REST y Web APIs.

Trabajos Futuros

Ya que el modelo creado en la plataforma Amazon Web Services es del tipo no supervisado y se comporta como un clasificador, se plantea crear un modelo del tipo supervisado con la herramienta Amazon SageMaker Studio, además de usar una mayor cantidad de datos e información para el entrenamiento de todos los modelos, finalmente comparar las métricas de cada uno de los modelos del presente trabajo con las métricas obtenidas de un modelo entrenado con mayor cantidad de información.

Se propone adicionalmente al Sistema de Escalas de Usabilidad (SUS), implementar cuestionarios como el Software Usability Measurement Inventory (SUMI) o Measuring the Usability of Multi-Media Systems (MUMMS) ya que estos son más específicos y con sus resultados será más sencillo determinar de forma puntual, donde aplicar mejoras al software, así como también conocer herramientas y desarrollos que los usuarios quisieran para el sistema.

Al fin de continuar con la línea de investigación se propone usar servicios de procesamiento de lenguaje natural de otras plataformas cloud como son Google Cloud Platform

y Huawei Cloud, de modo que se comparen las métricas de rendimiento y determinar cuál es la plataforma que hace al sistema más eficiente.

Integrar servicios de traducción de voz a texto y de texto a voz de las plataformas de Amazon Web Services, Google Cloud Platform y Microsoft Azure y comparar las métricas del tiempo de respuesta. Desplegar toda la aplicación como nativa de la nube para que pueda ser escalable y aceptar mayor demanda de usuarios.

Hacer uso de servicios de inteligencia artificial como el Asistente Virtual, procesamiento de lenguaje natural, traducción de Voz a Texto y de Texto a Voz a otras áreas como la educación, salud y diversos ámbitos laborales para facilitar y agilizar diversos procesos.

Referencias

- Amazon Web Services, Inc. (2020). *What is AWS*. Recuperado el 10 de Junio de 2020, de <https://aws.amazon.com/es/what-is-aws/>
- Ashley, K. D. (2000). Designing Electronic Casebooks that Talk Back THE CATO PROGRAM. *Jurimetrics*, 40(3), 275-319.
- AWS. (28 de Septiembre de 2018). *Características de Amazon Comprehend*. Obtenido de <https://aws.amazon.com/es/comprehend/features/>
- Bailey, J., Budgen, D., Turner, M., Kitchenham, B., Brereton, P., & Linkman, S. (2007). Evidence relating to Object-Oriented software design: A survey. *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)* (págs. 482-484). Madrid: IEEE. doi:10.1109/esem.2007.58
- Berkowitz, R. (2005). *The gift of science: Leibniz and the modern legal tradition*. Lóndres: Harvard university press.
- Binildas, C. A., Caselli, V., & Barai, M. (2008). *Service Oriented Architecture with Java*. Birmingham: Packt Publishing Ltd.
- Brewster, B., Ingle, T., & Rankin, G. (2014). Crawling Open-Source Data for Indicators of Human Trafficking. *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing* (págs. 714-719). Lóndres: IEEE. doi:10.1109/UCC.2014.116
- Buber, E., Diri, B., & Sahingoz, O. K. (2017). NLP Based Phishing Attack Detection from URLs. *ISDA 2017: Intelligent Systems Design and Applications* (págs. 608-618). Cham: Springer. doi:10.1007/978-3-319-76348-4_59
- BuenTrip. (17 de 04 de 2019). *Radar Tech Startup 4.0*. Recuperado el 03 de 04 de 2021, de <https://www.buentriphub.com/blog/2019/4/17/radar-tech-startup-40?rq=tech%20radar%204.0>
- Bues, M., & Matthaehi, E. (2017). LegalTech on the Rise: Technology Changes Legal Work Behaviours, But Does Not Replace Its Profession. *Management for Professionals*, 89-109. doi:10.1007/978-3-319-45868-7_7
- Cambria, E., & White, B. (2014). Jumping NLP Curves: A Review of Natural Language. *IEEE Computational Intelligence Magazine*, 9(2), 48-57. doi:10.1109/MCI.2014.2307227
- Carrell, D. (2011). A Strategy for Deploying Secure Cloud-Based Natural Language Processing Systems for Applied Research Involving Clinical Text. *2011 44th Hawaii International Conference on System Sciences* (págs. 1-11). Kauai: IEEE. doi:10.1109/HICSS.2011.32

- Centro Europeo de Postgrado. (08 de Febrero de 2019). *¿QUÉ ES SOA?* Obtenido de <https://www.ceupe.com/blog/que-es-soa.html>
- Chai, J., & Li, A. (2019). Deep Learning in Natural Language Processing: A State-of-the-Art Survey. *2019 International Conference on Machine Learning and Cybernetics (ICMLC)* (págs. 1-6). Kobe: IEEE. doi:10.1109/ICMLC48188.2019.8949185
- Chandramouli, R. (2019). Security Strategies for Microservices-based Application Systems. *NIST Special Publication 800-204* (págs. 1-41). Gaithersburg: National Institute of Standards and Technology. doi:10.6028/NIST.SP.800-204
- Chandrasekaran, K., & Ananth, A. (2016). Cloud Services and Service Providers. En S. Murugesan, & I. Bojanova, *Encyclopedia of Cloud Computing* (págs. 17-28). Chichester: John Wiley & Sons, Ltd.
- Chishti, S. (2020). *The LegalTech Book*. West Sussex: John Wiley & Sons Ltd.
- Corrales, M., Fenwick, M., & Haapio, H. (2019). Digital Technologies, Legal Design and the Future of the Legal Profession. *Perspectives in Law, Business and Innovation*, 1-7.
- De Marco, L., Ferrucci, F., Kechadi, M. T., Napoli, G., & Salza, P. (2016). Towards Automatic Service Level Agreements Information Extraction. *Proceedings of the 6th International Conference on Cloud Computing and Services Science* (págs. 59-66). Roma: Science and Technology Publications. doi:10.5220/0005873100590066
- Doglio, F. (2015). *Pro REST API Development with Node.js*. Canelones: APRESS.
- Dudhabaware, R., & Madankar, M. (2014). Review on natural language processing tasks for text documents. *2014 IEEE International Conference on Computational Intelligence and Computing Research* (págs. 1-5). Coimbatore: IEEE. doi:10.1109/ICCIC.2014.7238427
- El Kamali, M., Angelini, L., Caon, M., Khaled, O., Mugellini, E., Dulack, N., . . . Andreoni, G. (2020). NESTORE: Mobile Chatbot and Tangible Vocal Assistant to Support Older Adults' Wellbeing. *Proceedings of the 2nd Conference on Conversational User Interfaces (CUI '20)* (págs. 1-3). New York: Association for Computing Machinery. doi:10.1145/3405755.3406167
- Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., . . . Newling, T. (2004). *Patterns: Service Oriented Architecture and Web Services*. New York: IBM Corporation.
- Escobar, F. (2015). Leibniz, la Ciencia y el Código Civil. *IUS ET VERITAS*(50), 4-10.
- Farshchian, B. A., & Dahl, Y. (2015). The role of ICT in addressing the challenges of age-related falls: a research agenda based on a systematic mapping of the literature. *Personal and Ubiquitous Computing*, 19, 649–666. doi:10.1007/s00779-015-0852-1

- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Irvine: University of California. Obtenido de https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
- Gartner. (1 de Septiembre de 2020). *Magic Quadrant for Cloud Infrastructure and Platform Services*. Obtenido de <https://www.gartner.com/doc/reprints?id=1-242R58F3&ct=200902&st=sb>
- Giorgi, R., Fameli, E., & Nannucci, R. (1992). A Legal Expert System Prototype Integrated With Databases. *Expert Systems With Applications*, 4(4), 397-407.
- Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. *Journal of Artificial Intelligence Research*, 57, 345-420. Obtenido de <https://arxiv.org/pdf/1510.00726.pdf>
- Google Cloud. (20 de Julio de 2016). *Natural Language*. Obtenido de <https://cloud.google.com/natural-language>
- Google Cloud. (21 de Marzo de 2018). *Why google cloud*. Obtenido de <https://cloud.google.com/why-google-cloud/>
- Gour, R. (23 de Enero de 2019). *9 Major Characteristics of Cloud Computing*. Obtenido de DZone: <https://dzone.com/articles/9-major-characteristics-of-cloud-computing>
- Guo, W., Huang, L., & Pan, J. (2020). A Review of the Development and Application of Natural Language Processing. *International Conference on Genetic and Evolutionary Computing* (págs. 437-443). Singapore: Springer. doi:10.1007/978-981-15-3308-2_47
- Gupta, J. (6 de Marzo de 2019). *Cloud comparison tool – AWS Vs Google Vs IBM Vs Microsoft Vs Alibaba Cloud*. Obtenido de Wire 19: <https://wire19.com/cloud-comparison-tool-aws-vs-google-vs-ibm-vs-microsoft-vs-alibabacloud/>
- Gupta, S., Jagannath, K., Aggarwal, N., Sridar, R., Wilde, S., & Chen, Y. (2019). Artificially Intelligent (AI) Tutors in the Classroom: A Need Assessment Study of Designing Chatbots to Support Student Learning. *23rd Pacific Asia Conference on Information Systems*, (págs. 1-8). Xi'an. Obtenido de <https://aisel.aisnet.org/pacis2019/213/>
- Hacking, I. (1995). *El surgimiento de la probabilidad: un estudio filosófico de las ideas tempranas acerca de la probabilidad, la inducción y la inferencia estadística*. Barcelona: Gedisa.
- Ho, S. M., Oliveira, D., & Rathi, R. (2019). Consciousness of Cyber Defense: Boundary Objects for Expansive Learning Through Creation of Contradictions. *HCI 2019: HCI in Business, Government and Organizations. Information Systems and Analytics* (págs. 338-353). Cham: Springer. doi:10.1007/978-3-030-22338-0_28
- IAT. (2020). *LA INTELIGENCIA ARTIFICIAL Y SU APLICACIÓN AL DERECHO*. Recuperado el 29 de 03 de 2021, de <https://iat.es/tecnologias/inteligencia-artificial/derecho/>

- IBM Cloud Education. (23 de Octubre de 2019). *Microservices*. Obtenido de IBM: <https://www.ibm.com/cloud/learn/microservices>
- IBM Cloud Education. (2 de Julio de 2020). *What is Natural Language Processing?* Obtenido de <https://www.ibm.com/cloud/learn/natural-language-processing>
- IBM Cloud Education. (6 de 4 de 2021). *REST APIs*. Recuperado el 10 de 4 de 2021, de https://www.ibm.com/cloud/learn/rest-apis?mhsrc=ibmsearch_a&mhq=rest
- IBM Cloud Team. (2 de Septiembre de 2020). *SOA vs. Microservices: What's the Difference?* Obtenido de IBM: <https://www.ibm.com/cloud/learn/soa>
- International Data Corporation. (18 de Agosto de 2020). *Worldwide Semiannual Public Cloud Services Tracker*. Obtenido de <https://www.idc.com/getdoc.jsp?containerId=prUS46780320>
- Jayathilaka, K., Weerasinghe, A., & Wijesekara, W. (2016). Making sense of large volumes of unstructured email responses. *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)* (págs. 35-40). Negombo: IEEE. doi:10.1109/ICTER.2016.7829896
- Jones, T. (22 de Julio de 2020). *8 key characteristics of cloud computing*. Obtenido de TechTarget: <https://searchcloudcomputing.techtarget.com/feature/7-key-characteristics-of-cloud-computing>
- Karmel, A., Chandramouli, R., & Iorga, M. (2016). NIST Definition of Microservices, Application Containers and System Virtual Machines. *NIST Special Publication 800-180* (págs. 1-5). Gaithersburg: National Institute of Standards and Technology .
- Keserwani, P. K., & Samaddar, S. G. (2017). An SLA Design with Digital Forensic Capabilities. *2017 Ninth International Conference on Advanced Computing (ICoAC)* (págs. 109-113). Chennai: IEEE. doi:10.1109/ICoAC.2017.8441460
- Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Keele: Keele University and University of Durham. Obtenido de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.471&rep=rep1&type=pdf>
- Labaj, M., Rástočný, K., & Chudá, D. (2019). Towards Automatic Comparison of Cloud Service Security Certifications. *SOFSEM 2019: SOFSEM 2019: Theory and Practice of Computer Science* (págs. 298-309). Cham: Springer. doi:10.1007/978-3-030-10801-4_24
- Lau, R. (07 de 09 de 2020). *Update: LegalMation's New Approach to Legal Analytics*. Recuperado el 02 de 04 de 2021, de LegalMation: <https://www.legalmation.com/blog/update-legalmations-new-approach-to-legal-analytics/>

- Legaltechies. (14 de 07 de 2020). *El estado de la Legaltech en Ecuador*. Recuperado el 04 de 04 de 2021, de Legaltechies: <https://legaltechies.es/2020/06/25/el-estado-de-la-legaltech-en-ecuador/>
- Lesmo, L., Mazzei, A., Palmirani, M., & Radicioni, D. (2013). TULSI: an NLP system for extracting legal modificatory provisions. *Artificial Intelligence and Law*, 21, 139–172. doi:10.1007/s10506-012-9127-6
- Llangarí Salazar , A. M. (2016). *Repositorio de la Universidad de las Fuerzas*. Obtenido de Tema de tesis de grado: Análisis de los delitos informáticos y de telecomunicaciones en el Ecuador bajo las nuevas normas jurídicas.: <http://repositorio.espe.edu.ec/bitstream/21000/11654/1/T-ESPE-053079.pdf>
- Loxam, J. (25 de 10 de 2018). Applying AI and ‘new maths’ to solve complex real-world challenges. *Information Age*, págs. 10-11. Obtenido de <https://www.information-age.com/applying-ai-123475909/>
- Luo, Y., Cheng, L., Hu, H., Peng, G., & Yao, D. (2020). Context-rich Privacy Leakage Analysis through Inferring Apps in Smart Home IoT. *IEEE Internet of Things Journal*, 1-15. doi:10.1109/JIOT.2020.3019812
- Maciá Fernández, G. (11 de Marzo de 2008). El fraude en roaming: estrategias de ataque y de defensa. *Taller de la Infraestructura Regional Suramericana*, págs. 1-5.
- Mahmoud, Q. H. (Abril de 2005). *Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI)*. Obtenido de Oracle: <https://www.oracle.com/technical-resources/articles/javase/soa.html>
- Mathews, S. M. (2019). Explainable Artificial Intelligence Applications in NLP, Biomedical, and Malware Classification: A Literature Review. *CompCom 2019: Intelligent Computing* (págs. 1269-1292). Cham: Springer. doi:10.1007/978-3-030-22868-2_90
- Mekni, M., Baani, Z., & Sulieman, D. (2020). A Smart Virtual Assistant for Students. *Proceedings of the 3rd International Conference on Applications of Intelligent Systems (APPIS 2020)* (págs. 1-6). New York: Association for Computing Machinery. doi:10.1145/3378184.3378199
- Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. *NIST Special Publication 800–145* (págs. 1-7). Gaithersburg: National Institute of Standards and Technology. Obtenido de <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- Meza Ayala, M. (2008). *Fraudes en Telecomunicaciones*. Quito: Publiasesores.
- Microsoft. (05 de Abril de 2014). *What is Azure?* Obtenido de <https://azure.microsoft.com/en-us/overview/what-is-azure/>
- Microsoft Azure. (23 de Diciembre de 2016). *Language Understanding*.

- Ministerio de Justicia, Derechos Humanos y Cultos. (10 de Febrero de 2014). *Código Orgánico Integral Penal*. Obtenido de <http://www.funcionjudicial.gob.ec/index.php/es/normativa/codigo-organico-integral-penal.html>
- Ministerio de Telecomunicaciones y de la Sociedad de la Información. (18 de Febrero de 2015). *Ley Orgánica de Telecomunicaciones*. Obtenido de <https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2016/05/Ley-Org%C3%A1nica-de-Telecomunicaciones.pdf>
- Mittal, S., Gupta, A., Joshi, K. P., Pearce, C., & Joshi, A. (2017). A Question and Answering System for Management of Cloud Service Level Agreements. *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)* (págs. 684-687). Honolulu: IEEE. doi:10.1109/CLOUD.2017.92
- Murcia, J., Moreno, S., Díaz, D., & Gómez, F. (2019). C3-Sex: a Chatbot to Chase Cyber Pervert. *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCoM/CyberSciTech)* (págs. 50-57). Fukuoka: IEEE. doi:10.1109/DASC/PiCom/CBDCoM/CyberSciTech.2019.00024
- Murugesan, S., & Bojanova, I. (2016). Cloud Computing: An Overview. En S. Murugesan, & I. Bojanova, *Encyclopedia of Cloud Computing* (págs. 3-15). Chichester: John Wiley & Sons, Ltd.
- Nayak, S. P., & Pasumarthi, S. (2019). Automatic Detection and Analysis of DPP Entities in Legal Contract Documents. *2019 First International Conference on Digital Data Processing (DDP)* (págs. 70-75). Lóndres: IEEE. doi:10.1109/DDP.2019.00023
- Nazir, F., Butt, W., Anwar, M., & Khan Khattak, M. (2017). The Applications of Natural Language Processing (NLP) for Software Requirement Engineering - A Systematic Literature Review. *ICISA 2017: Information Science and Applications 2017* (págs. 485-493). Singapore: Springer. doi:10.1007/978-981-10-4154-9_56
- Padro, L., & Turmo, J. (2015). TextServer: Cloud-Based Multilingual Natural Language Processing. *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* (págs. 1636-1639). Atlantic City: IEEE. doi:10.1109/ICDMW.2015.102
- Patil, A., Marimuthu, K., Nagaraja Rao, A., & Niranchana, R. (2017). Comparative study of cloud platforms to develop a Chatbot. *International Journal of Engineering & Technology*, 6(3), 57-61.
- Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. *EASE'08: Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering* (págs. 68-77). Swindon: BCS

Learning & Development Ltd. Obtenido de
<https://dl.acm.org/doi/10.5555/2227115.2227123>

- Primicias. (19 de Septiembre de 2019). *Los cuatro delitos informáticos más recurrentes en Ecuador*. Obtenido de <https://www.primicias.ec/noticias/tecnologia/estos-delitos-informaticos-mas-recurrentes-ecuador/>
- Priyadarshini, S. B., Bagjadab, A. B., & Mishra, B. K. (2020). A Brief Overview of Natural Language Processing and Artificial Intelligence. En B. K. Mishr, & R. Kumar, *Natural Language Processing in Artificial Intelligence* (págs. 211-224). Burlington: Apple Academic Press Inc.
- Rath, M., Agarwal, S., & Shyamasundar, R. K. (2017). Semi Supervised NLP Based Classification of Malware Documents. *ICISS 2017: Information Systems Security* (págs. 334-344). Cham: Springer. doi:10.1007/978-3-319-72598-7_21
- Ray, A., & Mathew, R. (2019). Review of Cloud-Based Natural Language Processing Services and Tools for Chatbots. *Lecture Notes on Data Engineering and Communications Technologies* (págs. 156-162). Cham: Springer. doi:10.1007/978-3-030-24643-3_18
- Safanov, V. O. (2016). *Trustworthy Cloud Computing*. New Jersey: John Wiley & Sons, Inc.
- Simson, C. (6 de Agosto de 2019). *How ROSS AI Turns Legal Research On Its Head*. Obtenido de <https://blog.rossintelligence.com/post/how-ross-ai-turns-legal-research-on-its-head>
- Singhal, A., Winograd, T., & Scarfone, K. (2007). Guide to Secure Web Services. *NIST Special Publication 800-95* (págs. 1-128). Gaithersburg: National Institute of Standards and Technology.
- Sinoara, R., Antunes, J., & Rezende, S. (2017). Text mining and semantics: a systematic mapping study. *Journal of the Brazilian Computer Society*, 23(9), 1-20. doi:10.1186/s13173-017-0058-7
- Sulea O, M., Dinu L, P., & A, P. (2015). Using NLP Specific Tools for Non-NLP Specific Tasks. A Web Security Application. *ICONIP 2015: Neural Information Processing* (págs. 631-638). Cham: Springer. doi:978-3-319-26561-2_74
- Superintendencia de Telecomunicaciones. (2011). Delitos en Telecomunicaciones. 3-6, 12,16.
- Tablan, V., Bontcheva, K., Roberts, I., Cunningham, H., & Dimitrov, M. (2013). AnnoMarket: An Open Cloud Platform for NLP. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (págs. 19-24). Sofia: Association for Computational Linguistics. Obtenido de <https://www.aclweb.org/anthology/P13-4004.pdf>
- Vaquero, L., Rodero, L., Cáceres, J., & Lindner, M. (2009). A Break in the Clouds: Towards a Cloud Definition. *ACM SIGCOMM Computer Communication Review*, 39(1), 50-55. doi:10.1145/1496091.1496100

- Vennam, S. (18 de Agosto de 2020). *Cloud Computing*. Obtenido de IBM: <https://www.ibm.com/cloud/learn/cloud-computing>
- Yalla, P., & Sharma, N. (2015). Integrating Natural Language Processing and Software Engineering. *International Journal of Software Engineering and Its Applications*, 9(11), 127-136. Obtenido de <https://pdfs.semanticscholar.org/8fa7/a235347c2a9ba5e91cb054bd8167bb417491.pdf>
- Yang, H., Huang, L., Luo, C., & Yu, Q. (2020). Research on Intelligent Security Protection of Privacy Data in Government Cyberspace. *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)* (págs. 284-288). Chengdu: IEEE. doi:10.1109/ICCCBDA49378.2020.9095637
- Yogish, D., Manjunath, T., & Hegadi, R. (2019). Review on Natural Language Processing Trends and Techniques Using NLTK. *Recent Trends in Image Processing and Pattern Recognition (RTIP2R)* (págs. 589-606). Singapore: Springer. doi:10.1007/978-981-13-9187-3_53
- Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent Trends in Deep Learning. *IEEE Computational Intelligence Magazine*, 13(3), 55-75. doi:10.1109/MCI.2018.2840738
- Zahour, O., Benlahmar, E., Eddaoui, A., Ouchra, H., & Hourrane, O. (2020). Towards a Chatbot for educational and vocational guidance in Morocco: Chatbot E-Orientation. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(2), 2479-2487. doi:10.30534/ijatcse/2020/237922020

Anexos