



Instituto Politécnico Nacional

ESCUELA SUPERIOR DE CÓMPUTO

Trabajo Terminal I

2019-A087

Ingeniería en Sistemas Computacionales

BURROBOT

PRESENTAN:

Carlos David Hernández Martínez
José Ricardo López García

DIRECTORES:

M. en C. Ivan Giovanny Mosso García M. en C. Pabel Carrillo Mendoza

Palabras clave:

Chatbot, Inteligencia Artificial, Machine Learning, Procesamiento de
Lenguaje Natural, Mensajería Instantánea, Messenger Bot, Bases de datos,
AWS, Docker, Facebook, Graph API.



Ciudad de México, 11 de Noviembre del 2019

Índice general

Índice de figuras	III
Índice de tablas	V
1. Introducción	1
1.1. Resumen	1
1.2. Presentación	1
1.3. Justificación	2
1.4. Planteamiento del problema	2
1.5. Objetivos	3
1.5.1. Objetivo general	3
1.5.2. Objetivos específicos	4
1.6. Estado del arte	5
1.6.1. Investigación	5
1.7. Descripción del documento	9
2. Marco teórico	10
2.1. Ciencias de la computación	10
2.1.1. Inteligencia artificial	10
2.1.2. Procesamiento del lenguaje natural	10
2.1.3. Aprendizaje Automático	12
2.2. Entendimiento del lenguaje natural	12
2.3. Generador de lenguaje natural	12
2.4. Servicios de la nube	13
2.4.1. AWS	13
2.4.2. Micro servicios	14
2.4.3. Contenedor	14
2.4.4. Tipos de servicios	15
3. Análisis del prototipo	16
3.1. Reglas de negocio	16
3.2. Requerimientos funcionales	17
3.3. Requerimientos no funcionales	18

3.4.	Análisis de factibilidad económica	20
3.4.1.	Análisis de punto de función	20
3.4.2.	Análisis de riesgos	29
3.4.3.	Planes de contingencia	30
4.	Diseño del prototipo	33
4.1.	Metodología	33
4.2.	Tecnologías	34
4.3.	Modelado	35
4.3.1.	Diagramas de casos de uso	35
4.3.2.	Diagramas de procesos	39
4.3.3.	Diagramas Entidad - Relación	42
4.3.4.	Diagramas de clases	43
4.3.5.	Diagramas de secuencia	44
4.3.6.	Diagramas de infraestructura	47
4.3.7.	Construcción del modelo	48
4.3.8.	Resultados esperados	52
5.	Resultados	53
5.1.	Módulo de extracción (<i>Scraper</i>)	53
5.2.	Módulo de procesamiento de lenguaje natural	55
6.	Anexos	68
6.1.	Anexo I. Análisis de campo	68
6.1.1.	Encuesta	68
6.1.2.	Resultados	70
6.2.	Anexo II. Entrevista	76
6.3.	Anexo III. Métricas del ajuste de la completitud técnica	76

Índice de figuras

1.1. Captura de pantalla correspondiente al grupo de Facebook IPN Zacatenco	3
1.2. Captura de pantalla correspondiente al grupo de Facebook ESCOM IPN MX	3
4.1. Diagrama de casos de uso para el modulo de extracción denominado Scraper.	36
4.2. Diagrama de casos de uso para el modulo de pre-procesamiento de texto.	37
4.3. Diagrama de casos de uso para el modulo de <i>topic analysis</i>	38
4.4. Diagrama de casos de uso para el modulo de graficación.	38
4.5. Diagrama de procesos del módulo de extracción.	40
4.6. Diagrama de procesos del módulo de procesamiento de lenguaje natural.	41
4.7. Diagrama entidad-relación de la base de datos relacional.	42
4.8. Diagrama de clases del módulo <i>Scraper</i> o de extracción.	43
4.9. Diagrama de clases del módulo de procesamiento de lenguaje natural.	44
4.10. Secuencia por la que viaja una pregunta dentro de las tecnologías de AWS.	46
4.11. Infraestructura del agente conversacional	47
4.12. Infraestructura de los modulo de extracción y NLP respectivamente	48
4.13. Arquitectura <i>Seg2Seg</i>	49
4.14. Arquitectura de la LSTM	50
4.15. Arquitectura del mecanismo de atención	51
4.16. Resultados esperados usando la técnica de <i>Seg2Seg con mecanismos de atención</i>	52
5.1. Datos crudos obtenidos de <i>Graph API</i> almacenados en base de datos <i>NoSQL</i>	54
5.2. Datos crudos obtenidos de <i>Graph API</i> almacenados en el sistema local de archivos por conversación.	54
5.3. Datos crudos obtenidos de <i>Graph API</i> almacenados en el sistema local de archivos por mensaje.	54
5.4. Datos crudos obtenidos de <i>Graph API</i> almacenados en el sistema local de archivos visto desde un editor de texto.	55
5.5. Datos en diferentes niveles de normalización almacenados en la base de datos relacional.	56
5.6. Tabla de conversaciones.	56
5.7. Resultado LDA con documentos con <i>stopwords</i> y sin lemas (a).	57
5.8. Resultado LDA con documentos con <i>stopwords</i> y sin lemas (b).	57
5.9. Resultado LDA con documentos con <i>stopwords</i> y con lemas (a).	58

5.10. Resultado LDA con documentos con <i>stopwords</i> y con lemas (b).	58
5.11. Resultado LDA con documentos sin <i>stopwords</i> y sin lemas (a).	59
5.12. Resultado LDA con documentos sin <i>stopwords</i> y sin lemas (b).	60
5.13. Resultado LDA con documentos sin <i>stopwords</i> y sin lemas (c).	60
5.14. Resultado LDA con documentos sin <i>stopwords</i> y sin lemas (d).	61
5.15. Resultado LDA con documentos sin <i>stopwords</i> y sin lemas (e).	61
5.16. Resultado LDA con documentos sin <i>stopwords</i> y sin lemas (f).	62
5.17. Resultado LDA con documentos sin <i>stopwords</i> y con lemas (a).	63
5.18. Resultado LDA con documentos sin <i>stopwords</i> y con lemas (b).	64
5.19. Resultado LDA con documentos sin <i>stopwords</i> y con lemas (c).	64
5.20. Resultado LDA con documentos sin <i>stopwords</i> y con lemas (d).	65
5.21. Resultado LDA con documentos sin <i>stopwords</i> y con lemas (e).	65
5.22. Resultado LDA con documentos sin <i>stopwords</i> y con lemas (f).	66
5.23. Resultado LDA con documentos sin <i>stopwords</i> y con lemas (g).	66
5.24. Resultado LDA con documentos sin <i>stopwords</i> y con lemas (h).	67
6.1. Gráfico de barras: Pregunta 1	70
6.2. Gráfico de barras: Pregunta 2	71
6.3. Gráfico de barras: Pregunta 3	71
6.4. Gráfico de barras: Pregunta 4	72
6.5. Gráfico de barras: Pregunta 5	72
6.6. Gráfico de barras: Pregunta 6	73
6.7. Gráfico de barras: Pregunta 7	73
6.8. Gráfico de barras: Pregunta 8	74
6.9. Gráfico de barras: Pregunta 9	74
6.10. Gráfico de barras: Pregunta 10	75
6.11. Mapa de correlación de variables ente las preguntas de la encuesta	75

Índice de tablas

3.1. Reglas de negocio.	16
3.2. Requerimientos funcionales.	18
3.3. Requerimientos no funcionales.	19
3.4. Identificación de funciones y asignación de tipo.	20
3.5. Valores estándar de la <i>International Function Point Users Group</i>	21
3.6. Puntos de función sin ajustar.	21
3.7. Tabla de factores de ajuste	22
3.8. Distribución del esfuerzo	23
3.9. Costo humano	24
3.10. Costos materiales	24
3.11. Servicios de la capa gratuita que ocuparemos	25
3.12. AWS API gateway - precios.	25
3.13. AWS Lambda - precios.	26
3.14. AWS SQS tipo de encolador- precios.	26
3.15. AWS S3 almacenamiento - precios.	26
3.16. AWS S3 solicitudes - precios.	26
3.17. AWS tarifas de transferencia de datos entrante- precios.	27
3.18. AWS tarifas de transferencia de datos saliente- precios.	27
3.19. AWS SageMaker tarifas por hora de tipos de instancias contempladas en la capa gratuita	27
3.20. AWS SageMaker tarifas por GB de datos procesados	27
3.21. Gastos	27
3.22. Identificadores de riesgo.	29
3.23. Análisis de riesgo.	29
3.24. Plan de contingencia correspondiente al riesgo 01.	30
3.25. Plan de contingencia correspondiente al riesgo 02.	30
3.26. Plan de contingencia correspondiente al riesgo 03.	31
3.27. Plan de contingencia correspondiente al riesgo 04.	31
3.28. Plan de contingencia correspondiente al riesgo 05.	31
3.29. Plan de contingencia correspondiente al riesgo 06.	32
3.30. Plan de contingencia correspondiente al riesgo 07.	32
3.31. Plan de contingencia correspondiente al riesgo 09.	32

Capítulo 1

Introducción

1.1. Resumen

Los servicios de mensajería han resultado muy útiles para acercar a los clientes de productos o servicios con las empresas, sin embargo, el personal requerido para responder en tiempo real sobrepasa muchas veces las capacidades de la empresa o la institución. La Inteligencia Artificial ha permitido avanzar en la automatización de la comunicación vía mensajería instantánea. Se desarrollará un prototipo de agente conversacional, popularmente conocido como chatbot, que interactúa con los usuarios utilizando técnicas de Inteligencia Artificial y Procesamiento de Lenguaje Natural integrado a la red de mensajería Messenger. La finalidad es apoyar a los alumnos y aspirantes del Instituto Politécnico Nacional orientándolos en las necesidades más frecuentes en el ámbito de la gestión educativa.

1.2. Presentación

Debido a la gran presencia que tienen las redes sociales en nuestro día a día, una gran parte de la población del mundo las han adoptado como un medio tradicional de comunicación, ya que nos ofrecen una forma rápida para poder mantener contacto con familiares, amigos y conocidos.

Los medios que se suelen utilizar para llevar la conversación suelen ser aplicaciones de mensajería instantánea como Facebook Messenger y WhatsApp, los cuales están posicionados como las principales aplicaciones de mensajería debido a su gran y creciente popularidad. Además, en los últimos años estos medios se han potenciado debido a la posibilidad de comunicar directamente a las personas con instituciones y así establecer una atención personalizada.

Los avances significativos en los campos de la inteligencia artificial, el aprendizaje automático y el procesamiento del lenguaje natural, han provocado importantes cambios de como percibíamos las redes sociales, temas como segmentación del mercado, sistemas de recomendación, análisis de imágenes, agentes conversacionales, entre otros, son hoy en día métodos por los cuales las empresas hacen llegar a las personas sus productos.

Uno de los sistemas más usados por los usuarios de Facebook, puesto que han sido de gran ayuda por sus respuestas rápidas, eficientes y concisas, son los agentes conversacionales también conocidos como Chatbots.

Un chatbot es un servicio o herramienta con el que puedes comunicarte mediante mensajes de texto. Utiliza las bases de la Inteligencia Artificial y el Procesamiento de Lenguaje Natural haciéndolo capaz de comprender el contexto de lo el usuario está queriendo decir y responde con un mensaje coherente, relevante y directo relacionado con la tarea o petición que le estás solicitando.

Lo que los hace tan relevantes hoy en día es que: permite la interacción con usuarios por medio de sistemas de mensajería instantánea. Los chatbots modernos no dependen únicamente de los mensajes de texto. También pueden enviar imágenes, enlaces, vídeos, audios, etc. Esto les permite ser utilizados para diversos propósitos, tales como compras, servicio al cliente, noticias, juegos o campañas de mercadotecnia.

1.3. Justificación

Cada día se realiza una gran cantidad de preguntas relacionadas con temas administrativos por parte de los alumnos del Instituto Politécnico Nacional, los cuales buscan soluciones rápidas y precisas. Actualmente el instituto asigna recursos humanos y técnicos, destinados a dar soporte y solución a las dudas de los alumnos, algunas formas que se tienen para atenderlos son: presencial, asistencia telefónica, redes sociales, correo electrónico y páginas web. Sin embargo, el uso de recursos humanos para atender esta problemática demanda una gran cantidad de tiempo para dar solución a cada una de las consultas, por lo que no es posible dar una respuesta en tiempo y forma a muchas de las peticiones por parte de los usuarios.

La propuesta de solución consiste en la creación de un prototipo de agente conversacional que lleva por nombre Burrobot. Este sistema será capaz de atender usuarios con técnicas de inteligencia artificial y procesamiento del lenguaje natural a través de la plataforma de Facebook, para brindarles respuestas automáticas a sus consultas más recurrentes de manera personalizada y con un tiempo de respuesta inmediato y garantizando una alta disponibilidad.

1.4. Planteamiento del problema

Muchos de los procesos para realizar trámites en el ámbito de la gestión educativa son desconocidos por parte de los alumnos del Instituto Politécnico Nacional, naturalmente ellos recurren a sus propios recursos para poder obtener la información respectiva, sin embargo, al no haber un medio de consulta enfocado exclusivamente a la resolución de estos problemas donde toda esta información se encuentre centralizada, provoca que la información obtenida por los alumnos no sea completa, oportuna, relevante, accesible, entendible, clara o precisa. Esto conlleva en una serie de problemas que van desde pequeños contratiempos hasta no poder realizar el trámite. Las

figuras 1.1 y 1.2 son ejemplos de la problemática ya mencionada, donde no hay respuesta alguna por parte de los administradores a las peticiones de los usuarios.



Figura 1.1: Captura de pantalla correspondiente al grupo de Facebook **IPN Zacatenco**.



Figura 1.2: Captura de pantalla correspondiente al grupo de Facebook **ESCOM IPN MX**.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar un prototipo de agente conversacional, el cual se encargará de interactuar con los usuarios de una página de Facebook (la cual lleva por nombre **ESCOM IPN MX**) por medio de su sistema de mensajería instantánea "Messenger", orientándolos en las necesidades más frecuentes en el ámbito de la gestión educativa haciendo uso de técnicas de Inteligencia Artificial y Procesamiento de Lenguaje Natural.

1.5.2. Objetivos específicos

- Identificar las necesidades de administración escolar que puedan ser resueltas con la aplicación.
- Procesar recursos lingüísticos para el entrenamiento del chatbot.
- Construir el modelo de lenguaje.
- Integración con el servicio de mensajería Messenger.
- Garantizar la entrega de mensajes.
- El *bot* tendrá como usuarios finales a los aspirantes, alumnos y ex-alumnos de la Escuela Superior de Cómputo **ESCOM**.

1.6. Estado del arte

1.6.1. Investigación

Sistema	Descripción	Características	Estado actual	Referencia
B. R. Ranoliya, N. Raghuwanshi and S. Singh, Chatbot for university related FAQs,”2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017, pp. 1525-1530.	Chatbot basado en el conjunto de datos de preguntas frecuentes utilizando el lenguaje de marcado de inteligencia artificial y el análisis semántico latente (LSA)	<ul style="list-style-type: none"> ■ Resuelve preguntas generales usando un lenguaje de marcado ■ El modelo de NLP se basa en la obtención de tópicos obtenidos por el LSA 	Proyecto de investigación en desarrollo - No comerciable	Chatbot for university related FAQs
M. A. Limón, Construcción de un prototipo de programa personalizado de tipo chatbot en ambiente java”, Tesis, UPIICSA, 2016-07-04	Prototipo de sistema de chatbot capaz de realizar consultas a oraciones para obtener datos que ayuden en la toma de decisiones	<ul style="list-style-type: none"> ■ Aplicación de escritorio ■ Análisis de tópicos ■ Análisis de contexto 	Trabajo de terminal - No comerciable	Construcción de un prototipo de programa personalizado de tipo chatbot en ambiente java

Continuación de la tabla: 3.23

Sistema	Descripción	Características	Estado actual	Referencia
P. Muangkam-muen, N. Intiruk and K. R. Saikaew, "Automated Thai-FAQ Chatbot using RNN-LSTM," 2018 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2018, pp. 1-4.	Se desarrollo un Chatbot de preguntas frecuentes (FAQ) que responde automáticamente a los clientes mediante el uso de una red neuronal recurrente (RNN) en forma de memoria de corto plazo (LSTM) para la clasificación de texto. Los resultados experimentales han demostrado que chatbot podría reconocer el 86.36 % de las preguntas y responder con una precisión del 93.2 %	<ul style="list-style-type: none"> ■ Integración en pagina web ■ Redes neuronales recurrentes para procesar las solicitudes de los clientes ■ Cortos periodos de respuesta 	Proyecto desarrollado para uso interno de la organización - No comerciable	Automated Thai-FAQ Chatbot using RNN-LSTM
H. Agus Santoso et al., "Dinus Intelligent Assistance (DINA) Chatbot for University Admission Services," 2018 International Seminar on Application for Technology of Information and Communication, Semarang, 2018, pp. 417-423.	Chatbot que actúa como un agente conversacional que da asesoría a los estudiantes candidatos. DINA utiliza las bases de conocimiento provistas por la institución como parte fundamental a la hora de entrenar sus modelos predictivos. El patrón extraído puede usarse para proporcionar respuestas al usuario. La principal fuente de conocimiento es el libro de visitas de <i>Universitas Dian Nuswantoro</i> (UDINUS) que contiene preguntas y respuestas sobre los servicios de admisión	<ul style="list-style-type: none"> ■ Entrenamiento rápido ■ Se basa a partir de una lista de FAQs proporcionadas por la universidad 	Proyecto de investigación en desarrollo - No comerciable	Dinus Intelligent Assistance (DINA) Chatbot for University Admission Services

Continuación de la tabla: 3.23

Sistema	Descripción	Características	Estado actual	Referencia
BBVA Smart Assistant Nov 2017.[Blog Digital]. Disponible en : http://www.bbva.com	<p>Es un sistema integrado a su producto BBVA móvil. Ofrece a sus clientes una mejor experiencia al usuario al resolver sus dudas sobre gestión financiera, uso de sus aplicaciones y operaciones bancarias. Debido a su conectividad entre las diversas aplicaciones de la empresa, facilita toda la información financiera alojada en los sistemas de banca móvil mediante una conversación más humana, en un lenguaje cotidiano y consultas por medio de comandos de voz</p>	<ul style="list-style-type: none"> ■ Herramienta integrada a una aplicación móvil ■ Resuelve preguntas sencillas ■ Integración con redes sociales ■ Comandos de voz ■ Enfocado a la ayuda de gestión en las finanzas de sus clientes 	<p>Sistema de uso interno - No comerciable</p>	<p>BBVA</p>

Continuación de la tabla: 3.23

Sistema	Descripción	Características	Estado actual	Referencia
C. D. Hernandez y J. R. López, "Burrobot", ISC. Trabajo Terminal 2019-A087, Escuela Superior de Cómputo, CDMX, México, 2019.	Agente conversacional capaz de responder a preguntas frecuentes relacionadas a la gestión educativa por medio de la red social <i>Facebook Messenger</i> integrado a la página interna de la Escuela Superior de Cómputo ESCOM IPN MX , brindado respuestas automatizadas dando solución en cortos periodos de tiempo con un servicio continuo y de bajo costo operacional	<ul style="list-style-type: none"> ■ Gran escalabilidad ■ Tolerancia a fallos ■ Bajo costo operacional ■ Concurrencia de solicitudes ■ Alta disponibilidad ■ Prontitud de respuestas ■ Robustez 	Prototipo para uso de la comunidad sin costo para el público pero con un costo estimado de \$240,556	ESCOM IPN MX

1.7. Descripción del documento

A lo largo de este trabajo terminal explicaremos a detalle cada uno de los capítulos que conforman este documento. A continuación daremos un resumen de cada uno de estos.

Capítulo 1 Introducción: Resumen, planteamiento, objetivos, estado del arte y justificación del proyecto.

Capítulo 2 Marco conceptual: Conceptos técnicos de los cuales se hará referencia en los siguientes capítulos, esto con la finalidad de tener un mayor entendimiento del tema.

Capítulo 3 Análisis del sistema: Estudio de factibilidad, análisis de requerimientos, reglas del negocio, análisis de riesgos y finalmente los factores tomados en cuenta para la elección de las herramientas de software que emplearemos para el desarrollo del prototipo.

Capítulo 4 Diseño: Diagramas de procesos, secuencia, casos de uso, clases, entidad-relación así como el diseño de la infraestructura con el que el servicio será desplegado.

Capítulo 2

Marco teórico

Historia : En 1950, Alan Turing publica un artículo de Investigación llamado: Computing Machinery and Intelligence [1]; en la investigación, se cuestiona si las computadoras podrían pensar, aprender e interactuar con el usuario, además define la inteligencia de las máquinas como: “computadora humana capaz de procesar instrucciones y tener conciencia”. Gracias a la investigación de Turing, se inician las bases científicas para realizar las primeras investigaciones sobre inteligencia artificial y la comunicación entre humano computadora de una forma natural. Turing predijo que al final del siglo, el uso de las palabras y las opiniones educadas podrían alterarse tanto que se iba lograr hablar con las computadoras.

Objetivo del capítulo : Dado que este trabajo terminal emplea conceptos de diversas áreas de la Ingeniería en Sistemas Computacionales, resulta fundamental dar una serie de definiciones técnicas que permitan un mejor entendimiento del documento.

2.1. Ciencias de la computación

2.1.1. Inteligencia artificial

Es una rama de las ciencias de la computación que se encarga de realizar tareas complejas emulando ciertas funciones cognitivas propias de los seres humanos

2.1.2. Procesamiento del lenguaje natural

El Procesamiento de Lenguaje Natural, o también conocido como computación lingüística, le permite a la computadora interpretar el lenguaje humano basado en el razonamiento, aprendizaje y entendimiento. El PLN fue transformado por investigadores para poder construir un modelo exitoso en la traducción humano-computadora con lenguajes empíricos de datos. Para entender el lenguaje humano se han desarrollado tres técnicas:

Machine Translation: Es la forma como la computadora traduce lenguajes humanos y logra descomponerlos en una estructura semántica entendida por una computadora. Según Hirschberg, esta tecnología ha avanzado gracias al *Deep Learning*, que consiste en entrenar un modelo con diferentes representaciones para optimizar un objeto final, en este caso, las traducciones [3]. Además, permite el uso de tecnologías actuales como: Google Translate y Skype Translator.

Speech Recognition: Es el proceso de convertir una señal del diálogo en una secuencia de palabras por medio de un algoritmo implementado por un programa de computadora [2]. La tecnología de speech recognition ha hecho posible a la computadora responder por comandos de voz y entender el lenguaje natural como lo hacen los asistentes virtuales de los teléfonos y los parlantes como Alexa. Existen tres formas del speech recognition: palabras insoladas, palabras conectadas, y diálogo espontáneo [2].

Speech Synthesis: Es la forma en que la computadora pasa de texto a diálogo, y el software debe comprender la entonación, pronunciamiento y duración de este.

En el mundo del procesamiento de lenguaje natural, existen técnicas y conceptos estudiados por computólogos y lingüistas expertos en esta área con la finalidad de poder lograr que las máquinas entiendan con mayor rapidez y facilidad el lenguaje humano, algunos de estos conceptos son:

Acceso a la información: Es la capacidad que permite dar acceso a información relevante a un usuario a medida que nuevos elementos estén disponibles.

Adquisición de conocimiento: Es la capacidad que permite adquirir conocimiento útil de un texto, generalmente se obtiene con la revisión a detalle del texto y haber obtenido patrones tras sintetizar múltiples documentos.

Stemming: La radicación o stemming es el proceso que consiste en descomponer una palabra en sus elementos constituyentes o afijos, es decir: prefijos, infijos y sufijos.

Lema: Es la forma canónica de una palabra, que por convenio se acepta como representante de todas las formas flexionadas de una misma palabra.

Stopwords: Es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc.

Tokenización: La tokenización es el proceso de dividir textos en unidades mínimas significativas.

Latent Dirichlet Allocation: La Asignación Latente de Dirichlet es un algoritmo de aprendizaje no supervisado que intenta describir un conjunto de observaciones como una mezcla de distintas categorías.

Chatbot: Un chatbot es una tecnología capaz de simular una conversación humana a través de una interfaz conversacional.

2.1.3. Aprendizaje Automático

Es una rama de las ciencias de la computación la cual se encarga de automatizar la construcción de modelos con base en el análisis de los datos a través del uso de algoritmos que iterativamente aprenden de los datos, su objetivo es proveer a la inteligencia artificial la capacidad de aprender automáticamente, algunas definiciones importantes dentro de esta área son:

Neurona artificial: Unidad básica de procesamiento cuyo funcionamiento se basa en el comportamiento de una neurona biológica

Red neuronal: Un algoritmo del aprendizaje automático que consiste en un conjunto de neuronas artificiales interconectadas, dichas conexiones albergan valores que representan la relevancia de partes específicas de la información de entrada.

Aprendizaje supervisado: Es una rama del aprendizaje automático donde el modelo aprende con base en ejemplos etiquetados e intentara encontrar una relación con los valores de salida. Tras haber entrenado de manera iterativa el modelo con una serie de conjuntos de valores de entrada y salida hasta alcanzar una métrica de error aceptable, el modelo sera capaz de predecir predecir y/o clasificar un valor desconocido con respecto al conjunto de entrenamiento.

Aprendizaje no supervisado: Es una rama del aprendizaje automático en la el modelo aprende con base de observaciones de las entradas, carece de una variable objetivo por lo que el modelo intentara agrupar los elementos.

Aprendizaje profundo: Es la implementación de redes neuronales en los algoritmos de aprendizaje automático, lo cual implica que el aprendizaje se haga de forma jerarquizada, a medida que la información pase por las distintas capas se obtendrá una mayor abstracción de la información entrante, la cantidad de capas que contengan estas redes neuronales pueden influir en la capacidad del algoritmo en entender el problema.

2.2. Entendimiento del lenguaje natural

Es el responsable de transformar los enunciados de los mensajes obtenidos por parte del usuario a un formato comprensible para el agente conversacional, una de las principales tareas de este módulo es el detectar la intención del mensaje así como obtener nueva información adicional con forme a las restricciones del dominio cerrado

2.3. Generador de lenguaje natural

Es el responsable de generar texto partiendo de una entrada con una representación simbólica proveída por el usuario, cada respuesta proveída por este generador es tratada como un sistema de

marcos, el se mapea a una oración y en base a esta se construye la siguiente salida del generador.

Los componentes de estos generadores pueden ser basados en modelos o en reglas. sin embargo, al estar basados en reglas las oraciones generadas son limitadas, adaptándose a las plantillas de respuestas que tienen.

Hay otro tipo de generadores basados en el aprendizaje automático, los cuales usan varias fuentes de entrada el cual es capaz de producir varios enunciados candidatos, los cuales se pueden tratar de diferentes maneras, como se clasifican o se selecciona con base en reglas.

2.4. Servicios de la nube

El computo en la nube consta de una serie de recursos configurables que son ofrecidos al público en general, estos servicios se proveen vía internet de manera remota y tratan de asignar los recursos que el cliente requiere lo antes posible.

Las características esenciales del computo en la nube son:

Autoservicio bajo demanda El cliente debe poder obtener una extensión de recursos a pedido y sin la necesidad de interacción humana.

Amplio acceso a la red El acceso a la red debe constar de una gran disponibilidad para cualquier tipo de dispositivo que use el cliente.

Conmutación de recursos Un servicio en la nube determinado debería poder servir a múltiples usuarios simultáneamente, con recursos físicos y virtuales asignados y reasignados dinámicamente de acuerdo con la demanda del cliente.

Rápida elasticidad Los recursos que solicite el cliente deben de ser elásticamente escalables tanto horizontal como verticalmente de acuerdo con la demanda, sea cual sea la cantidad de recursos requeridos, y en cualquier momento

Servicio medido Los clientes se atienden a pagar por cada servicio que usan, dependiendo del proveedor de la nube, se les asignaran tarifas de acuerdo a lo que ellos consideren correspondiente del uso de su servicio.

2.4.1. AWS

Ofrecida oficialmente por Amazon en el año 2006, Amazon Web Services (de ahora en adelante AWS) lanzo al público en general sus servicios de computo en la nube. Si bien es cierto, no fue la primera empresa en ofrecer este tipo de servicios, su modelo de negocios hacen posible el pago por uso de recursos como redes, informática, almacenamiento y servidores de aplicaciones datos a un costo escalable conforme a las necesidades de los usuarios.

Hoy en día, Amazon Web Services es líder en el sector de informática en la nube y los principales beneficios que ofrece son:

Bajo costo: AWS se maneja bajo una economía de escala en la cual nos ofrecen mayor potencia de recursos de informática a un menor costo operativo.

Disponibilidad: AWS tiene presencia en mas de 180 países, con 69 zonas de disponibilidad en 22 regiones geográficas de todo el mundo.

Seguridad: Los servicios y centros de datos disponen de múltiples capas de seguridad operativa y física para asegurar la integridad y seguridad de los datos.

Rendimiento: Los servicios ofrecidos constan de baja latencia y alto rendimiento con capacidades prácticamente ilimitadas la cual nos ayuda a recibir los cambios de las necesidades sin perder el rendimiento.

2.4.2. Micro servicios

La arquitectura basada en micro servicios tiene como fin el separar el sistema en varios servicios de tal manera que cada uno de estos funja como una aplicación.

Un servicio es una colección de componentes distribuidos que proveen de una funcionalidad a una aplicación o sistema, estos pequeños servicios son capaces de correr en distintos procesos y se comunican entre si haciendo comúnmente uso de las APIs. Existen dos tipos de comunicación entre los servicios: síncrona y asíncrona. En el caso de la comunicación síncrona, un servicio **x** manda una solicitud a otro servicio **y**, así, **x** se queda a la espera de la respuesta de **y**. En el caso de las llamadas asíncronas, el servicio **x** manda la solicitud a **y**, pero en lugar de quedarse a la espera de una respuesta, este continua su ejecución. Dependiendo de la frecuencia de comunicación entre servicios se determinara el grado de independencia y autonomía que tiene el servicio.

El objetivo de llevar un sistema a este grado de modulación es poder escalar hacia arriba o abajo el proyecto de acuerdo a las necesidades del mismo para así poderlo orquestarlo de forma exitosa. Esto permite gestionar de manera rápida y eficiente cada uno de los servicios, de tal manera que las tareas más comunes como el montaje, inicialización, chequeo de salud, comunicación, exposición de puertos, escalamiento, etc., ya no necesiten de intervención humana y sea posible la automatización del despliegue del sistema o aplicación, para ello simplemente definimos la arquitectura de servicios que ocupa nuestro sistema en una lista de procesos.

2.4.3. Contenedor

Es un paquete en el cual se incluyen todas las configuraciones, dependencias y software, de tal manera que se pueda usar como un ejecutable, las principales características son su seguridad, su flexibilidad al correr sobre cualquier sistema operativo que soporte un gestor de contenedores y la ligereza al momento de estar en ejecución. la diferencia con las maquinas virtuales es que los contenedores virtualizan un sistema operativo ligero y no todo el hardware.

2.4.4. Tipos de servicios

Software como servicio: Son programas o aplicaciones ofrecidos como servicios, los cuales corren sobre la infraestructura de un proveedor de servicios de la nube, esto implica no tener que administrar ni controlar los recursos ocupados por el programa como la red, el sistema operativo, almacenamiento, entre otros. Solo se permite configurar ciertos aspectos del servicio a ocupar de acuerdo a las necesidades del programa. Es necesario proveer al cliente de alguna interfaz con la cual pueda interactuar desde diferentes dispositivos para consumir el servicio de manera remota

Plataforma como servicio: Es una serie de servicios desplegadas dentro de un entorno los cuales corren sobre la infraestructura de un proveedor de servicios de la nube, esto implica no tener que administrar ni controlar los recursos ocupados por el programa como la red, el sistema operativo, almacenamiento, entre otros, únicamente sobre las aplicaciones desplegadas y algunas de las configuraciones de entorno del alojamiento de esas aplicaciones.

Infraestructura como servicio: Son una forma de ofrecerle al cliente recursos informáticos, almacenamiento y redes dentro de las características de nuestros servicios, al igual que los otros dos tipos de servicios, no podemos administrar los recursos sin embargo se tiene una mayor capacidad de configuración y cierto control sobre estos.

Capítulo 3

Análisis del prototipo

3.1. Reglas de negocio

Número	Descripción
1	Prototipo de agente conversacional que será utilizado con fines informativos.
2	Gestión escolar proveerá una lista de preguntas y respuestas sobre los trámites del ámbito educativo.
3	Se obtendrá los mensajes de por lo menos un grupo o página de Facebook oficial de la escuela.
4	El administrador del grupo o página en cuestión proveerá de las credenciales necesarias para obtener sus conversaciones.
5	El prototipo de agente conversacional responderá preguntas de la comunidad realizadas por medio del servicio de Messenger de la pagina de Facebook asociada.
6	Solo se responderá a los mensajes cuyo contenido sea únicamente texto.
7	El uso inadecuado de la tecnología propuesta será causa de una suspensión temporal del servicio al usuario.

Tabla 3.1: Reglas de negocio.

3.2. Requerimientos funcionales

Número	Requerimiento	Descripción
1	Servicio a proveer	El prototipo de agente conversacional se encargará de responder preguntas frecuentes sobre temas de gestión escolar.
2	Almacenamiento de mensajes no estructurados	La información no estructurada obtenida por las reglas de negocio 2 y 3 se almacenan en una base de datos no relacional.
3	Preprocesado	Los datos almacenados de acuerdo al requerimiento funcional dos serán preprocesados para poder hacer un análisis de texto.
4	Almacenamiento de mensajes estructurados	Los datos preprocesados de acuerdo al requerimiento funcional 3 serán almacenados en una base de datos relacional.
5	<i>Topic analysis</i>	De los datos estructurados obtenidos por el requerimiento funcional 4, se realizara un análisis con el algoritmo LDA para encontrar los temas más frecuentes.
6	Entrenamiento	La información alojada en la base de datos relacional de acuerdo al requerimiento funcional 4 servirán para entrenar el modelo de inteligencia artificial.
7	Validación de seguridad	Se verificarán los mensajes provenientes de la API de Facebook para saber si no es algún tipo de ataque DDos a nuestro prototipo.
8	Recepción de mensajes	La API de Facebook provee de un <i>end-point</i> de mensajes escritos por los usuarios para su obtención, procesamiento y validación.

Continuación de la tabla: 3.23		
Número	Nombre	Descripción
9	Envío de mensajes	Se devolverá la respuesta a la API de Facebook la cual se encargará de entregar esta al <i>chat</i> correspondiente del usuario en cuestión.

Tabla 3.2: Requerimientos funcionales.

3.3. Requerimientos no funcionales

Número	Requerimiento	Descripción
1	Rendimiento	Se ofrece un servicio con tiempos de respuesta cortos con capacidad de responder en paralelo múltiples peticiones sin afectar su rendimiento.
2	Alta disponibilidad	El prototipo de agente conversacional asegura una continuidad operacional por largos periodos de tiempo de tal forma que el servicio siempre este disponible.
3	Auto administrado	No requiere de intervención humana en la parte operativa para su correcto funcionamiento.
4	Robustes	Cuenta con esta característica debido a su modulación de código de acuerdo a su tipo de arquitectura, buenas practicas implementadas y su tolerancia a fallos.
5	Bajo costo operacional	Las tecnologías utilizadas serán adecuadas para el despliegue del proyecto pensando en la optimización de recursos, solo se pagará por lo que se usa.

Continuación de la tabla: 3.23		
Número	Nombre	Descripción
6	Escalabilidad	El prototipo será capaz de atender a todos los mensajes ajustando automáticamente la cantidad de recursos que necesita para llegar a cabo todas las tareas.
7	Fiabilidad	Los mensajes que lleguen al <i>chat</i> de la página de Facebook asociada al proyecto serán respondidas en forma ordenada.
8	Seguridad	Las tecnologías ocupadas para este prototipo cuentan con las normas ISO/IEC 27001:2013, 27017:2015, 27018:2014 y 9001:2015. La comunicación entre el prototipo y Facebook Graph API se dará mediante el protocolo SSL.
9	Concurrencia de usuarios	Será capaz de responder en paralelo a una gran cantidad preguntas enviadas a nuestro prototipo por medio de Facebook Graph API.
10	Implementación	El proyecto sera desplegado haciendo uso de la infraestructura y servicios proveídos por AWS.
11	Arquitectura del software	El proyecto estará compuesto por una serie de micro servicios que se comunicaran entre ellos.
12	Lenguaje de programación	Se ocupará el lenguaje de programación Python
13	Versionamiento de código	El código utilizado para el desarrollo de este proyecto sera gestionado por un controlador de versiones

Tabla 3.3: Requerimientos no funcionales.

3.4. Análisis de factibilidad económica

Objetivos:

- Estimar el esfuerzo requerido para realizar el proyecto
- Estimar la duración del desarrollo del proyecto
- Estimar el costo del proyecto

3.4.1. Análisis de punto de función

Asignación de puntos :

1. Entrada externa - Pantallas donde se ingresan los datos
2. Salida externa - Pantalla donde se reciben los datos
3. Consulta externa - Recuperación de datos del cliente

Almacenamiento Función de datos :

1. Archivo lógico interno - Datos guardados del cliente
2. Archivo de interfaz externo - Datos compartidos entre sistemas externos

Identificación de funciones y asignación de tipo :

Nombre	Tipo de punto de función
Extracción de mensajes	Archivo de interfaz externo
Almacenamiento de datos crudos	Archivo lógico interno
Análisis y preprocesamiento de datos crudos	Archivo lógico interno
Gestión de datos estructurados preprocesados	Archivo lógico interno
Envío de mensaje de Facebook API	Archivo de interfaz externo
Envío de respuesta a Facebook API	Archivo de interfaz externo
Validación del mensaje	Archivo lógico interno
Análisis del mensaje	Archivo lógico interno
Búsqueda entre documentos	Archivo lógico interno

Tabla 3.4: Identificación de funciones y asignación de tipo.

Catalogo de valores de los tipos de puntos de función sin ajustar según su complejidad :

Tipo	Baja	Media	Alta
Entrada externa	3	4	6
Salida externa	4	5	7
Consulta externa	3	4	6
Archivo lógico interno	7	10	15
Archivo de interfaz externo	5	7	10

Tabla 3.5: Valores estándar de la *International Function Point Users Group*.**Obtención de los puntos de función sin ajustar :**

Tipo	Baja	Media	Alta	Total
Archivo lógico interno	Alta	15	6	90
Archivo de interfaz externo	Alta	10	3	30
Total de puntos de función sin ajustar				120

Tabla 3.6: Puntos de función sin ajustar.**Obtención del factor de ajuste :**

Factor de ajuste	Puntaje
Comunicación de Datos	5
Función Distribuida	5
Rendimiento	4
Configuración utilizada masivamente	4
Tasas de Transacción	4
Entrada On-Line de datos	5
Diseño para la eficiencia de usuario final	5
Actualización On-Line	2
Complejidad del procesamiento	5
Utilizable en otras aplicaciones	5
Facilidad de Instalación	5
Facilidad de Operación	5
Puestos Múltiples	3
Facilidad de Cambio	5
Requerimientos de otras Aplicaciones	3
Seguridad, Privacidad y Auditoría	2
Uso directo de otras empresas	4
Documentación	5

Continuación de la tabla: 3.23	
Factor de ajuste	Puntaje
Total	76

Tabla 3.7: Tabla de factores de ajuste .**Ajuste de punto de función :**

$$PFA = PFSA * [0.65 + (0.01 * FA)] \quad (3.1)$$

Sea:

- PFSA : Puntos de función sin ajustar
- PFA : Puntos de función ajustado
- FA : Factor de ajuste

Donde:

- PFSA = 120
- FA = 76

Basados en los resultados de las tablas: 3.6 y 3.11 respectivamente, entonces:

$$PFA = 120 * [0.65 + (0.01 * 76)] \quad (3.2)$$

Por lo tanto:

$$PFA = 169.2$$

Estimación de esfuerzo en Horas Hombre (HH) :

$$HH = PFA / CI * HDI \quad (3.3)$$

Sea:

- HH : Horas Hombre
- PFA : Puntos de función ajustado
- CI: Cantidad de integrantes
- HDI: Promedio de horas diarias invertidas

Donde:

- Punto función ajustado = 169.2
- Cantidad de integrantes = 2

- Promedio de horas = 8

Entonces:

$$HH = 169.2/2 * 8 \quad (3.4)$$

Por lo tanto:

$$HH = 676.8$$

Duración del desarrollo del proyecto :

$$DP = HH/HDI * DMI \quad (3.5)$$

Sea:

- HH : Horas hombre
- HDI : Promedio de horas diarias invertidas
- DMI: Promedio de días invertidos al mes
- DP: Duración del proyecto en meses

Donde:

- DMI = 22
- HDI = 4

Entonces:

$$DP = 676.8/4/22 \quad (3.6)$$

Por lo tanto:

$$DP = 7.69 \text{ meses} \approx 7 \text{ meses } 21 \text{ días}$$

Distribución genérica del esfuerzo :

Actividad	Porcentaje	Tiempo
Entendimiento del negocio	20	1 mes 17 días
Adquisición y entendimiento de los datos	30	2 meses 9 días
Modelado	25	1 mes 28 días
Despliegue del servicio	15	1 mes 4 días
Pruebas	10	23 días

Tabla 3.8: Distribución del esfuerzo

Costo del proyecto :**Costo de desarrollo :**

Recurso	Horas	Sueldo por hora	Cantidad	Total
Data scientist	696	\$190	2	\$264,480
Devops Engineer	136	\$213	2	\$57,936
Tester	92	\$144	2	\$26,496
Total 3.23				\$348,912

Tabla 3.9: Costo humano**Costos materiales :**

Recurso	Costo
Hardware	
1 computadora Macbook pro	\$ 20,000
1 computadora Dell serie 5000	\$ 19,300
Software	
MacOS	\$0.0
Linux ubuntu	\$0.0
Python	\$0.0
Docker	\$0.0
AWS capa gratuita	\$0.0
*AWS	\$20000
MongoDB	\$0.0
MySQL GPL	\$0.0
Total	\$59,300

Tabla 3.10: Costos materiales

Computo en la nube : En la tabla 3.10 hablamos sobre los costos materiales previstos para el proyecto, a continuación especificaremos cada servicio que ocuparemos, en la siguiente tabla se muestran los limites mensuales que ofrece AWS en su capa gratuita.

Servicio	Recurso	Cantidad
Almacenamiento		
S3	Almacenamiento estándar	5GB
	Solicitudes GET	20 000
	Solicitudes PUT, COPY, POST o LIST	2 000
	Transferencia de datos de salida	15 GB
Machine learning		
Sage Maker	Cuaderno de notas con instancias t2.medium o t3.medium	250 Hrs
	Entrenamiento con maquinas m4.xlarge o m5.xlarge	50 Hrs
	Implementacion del modelo con maquinas m4.xlarge o m5.xlarge	125 Hrs
Computación		
Lambda	Solicitudes	1 000 000
	Tiempo de informatica	2 200 Hrs
Soluciones móviles		
API Gateway	Llamadas al API	1 000 000
	Mensajes	1 000 000
	Minutos de conexión	3 125 Hrs
Integración de aplicaciones		
SQS	Solicitudes	1 000 000

Tabla 3.11: Servicios de la capa gratuita que ocuparemos .

Costos de los servicios al sobrepasar la capa gratuita : En la tabla 3.10 se contemplan \$20,000 por si la capa gratuita es insuficiente de acuerdo a las necesidades del proyecto. Se empezara a cobrar por el tipo de servicio que haya alcanzado el limite al sobre pasar los que nos ofrece la capa gratuita de AWS, se cobra bajo demanda y únicamente por lo que se usa.

Número de solicitudes (por mes)	Precio (por millón)
Primeros 333 millones	3,50 USD
Próximos 667 millones	2,80 USD
Próximos 19 mil millones	2,38 USD
Más de 20 mil millones	1,51 USD

Tabla 3.12: AWS API gateway - precios.

Memoria (MB)	Segundos de la capa gratuita al mes	Precio por 100 ms (USD)
128	3 200 000	0,000000208
192	2 133 333	0,000000313
...
2 944	139 130	0,000004793
3 008	136 170	0,000004897

Tabla 3.13: AWS Lambda - precios.

Tipo de encolador	Precios por cada millón de solicitudes después de la capa gratuita (mensual)
Cola estándar	0,40 USD
Cola FIFO	0,50 USD

Tabla 3.14: AWS SQS tipo de encolador- precios.

Almacenamiento estándar	Precio
Primeros 50 TB/mes	0,023 USD por GB
Siguientes 450 TB/mes	0,022 USD por GB
Más de 500 TB/mes	0,021 USD por GB

Tabla 3.15: AWS S3 almacenamiento - precios.

Solicitudes al almacenamiento estándar	Precio
Datos devueltos por S3 Select	0,0007 USD por GB
Datos escaneados por S3 Select	0,002 USD por GB
Solicitudes PUT, COPY, POST o LIST	0,005 USD por cada 1000 solicitudes
GET, SELECT y el resto de las solicitudes	0,0004 USD por cada 1000 solicitudes

Tabla 3.16: AWS S3 solicitudes - precios.

Transferencia ENTRANTE de datos	Precio
Todas las transferencias entrantes de datos	0,00 USD por GB

Tabla 3.17: AWS tarifas de transferencia de datos entrante- precios.

Transferencia SALIENTE de datos	Precio
Hasta 1 GB/mes	0,00 USD por GB
Siguientes 9,999 TB/mes	0,09 USD por GB
Siguientes 40 TB/mes	0,085 USD por GB
Siguientes 100 TB/mes	0,07 USD por GB
Más de 150 TB/mes	0,05 USD por GB

Tabla 3.18: AWS tarifas de transferencia de datos saliente- precios.

Tipo de instancia	Precio
Cuaderno de notas	
t2.medium	0,0464 USD
t3.medium	0,0582 USD
Entrenamiento e implementación	
m4.xlarge	0,28 USD
m5.xlarge	0,269 USD

Tabla 3.19: AWS SageMaker tarifas por hora de tipos de instancias contempladas en la capa gratuita

Procesamiento de datos	Precio
Datos procesados ENTRANTES	0,016 USD por GB
Datos procesados SALIENTES	0,016 USD por GB

Tabla 3.20: AWS SageMaker tarifas por GB de datos procesados**Gastos :**

Servicio	Meses	Costo	Total
Luz	8	400	\$3,200
Internet	8	450	\$3,600
Total			\$6,800

Tabla 3.21: Gastos

Total del proyecto :

$$TP = CM + CH + G \quad (3.7)$$

Sea:

- TP : Costo total del proyecto
- G : Gastos totales
- CH: Costo Humano
- CM: Costo Material

Donde:

- G = \$6,800
- CH = \$348,912
- CM = \$59,300

Entonces:

$$TP = \$59,300 + \$348,912 + \$6,800 \quad (3.8)$$

Por lo tanto:

$$TP = \$415,012$$

3.4.2. Análisis de riesgos

Su objetivo es identificar los riesgos, es decir, su categoría, probabilidad de ocurrir e impacto en el proyecto.

Gravedad	Valor	Descripción
Alto riesgo	3	Provocaría el fracaso del proyecto.
Mediano riesgo	2	Mermaría la funcionalidad del proyecto.
bajo riesgo	1	Provocaría impactos mínimos en el proyecto.

Tabla 3.22: Identificadores de riesgo.

ID	RIESGO	CATEGORÍA	PROBABILIDAD	IMPACTO
01	Inexperiencia en las tecnologías para el desarrollo del sistema.	Técnico	40 %	1
02	Desintegración del equipo	Organizativo	5 %	2
03	Falta de coordinación entre los integrantes del equipo	Organizativo	80 %	3
04	Retraso de los recursos lingüísticos	Organizativo	40 %	2
05	Falta de los recursos lingüísticos	Organizativo	90 %	3
06	Ambigüedad de los recursos lingüísticos	Organizativo	50 %	2
07	Documentación e información escasa o redundante	Externo	20 %	1
09	Priorizar temas ajenos al proyecto	Organizativo	90 %	3

Tabla 3.23: Análisis de riesgo.

3.4.3. Planes de contingencia

De los riesgos que se describieron previamente, es adecuado seguir un plan de contingencia y prevención en el sentido de que los riesgos no lleguen a ser obstáculo en el desarrollo del sistema y si llegan a ocurrir tener una forma de cómo disminuir el impacto que pudieran tener.

Hoja de información del riesgo
ID: 01.
Descripción: Los integrantes del equipo no cuentan con experiencia o conocimiento mínimo en las tecnologías para el desarrollo del proyecto.
Contexto: Debido a la gran variedad de tecnologías que se ocuparan a lo largo del desarrollo del proyecto, algunas cuya integración es indispensable, es necesario contar con cierto grado de conocimiento o experiencia para el manejo de la misma.
Mitigación: <ul style="list-style-type: none"> ■ Revisar documentación. ■ Pruebas de concepto. ■ Ejemplos de uso.

Tabla 3.24: Plan de contingencia correspondiente al riesgo 01.

Hoja de información del riesgo
ID: 02.
Descripción: Algún integrante del equipo abandona el proyecto.
Contexto: Sea un factor personal u externo, un miembro del equipo decide abandonar el proyecto dando como lugar a una reorganización de las tareas.
Mitigación: <ul style="list-style-type: none"> ■ El integrante debe seguir las buenas practicas de desarrollo necesarias para que algún otro miembro pueda continuar con sus tareas. ■ Re-organización adecuada.

Tabla 3.25: Plan de contingencia correspondiente al riesgo 02.

Hoja de información del riesgo
ID: 03.
Descripción: Mala coordinación entre los integrantes del proyecto para las sesiones de trabajo
Contexto: La mala coordinación puede derivar en una serie de mal entendidos, entorpecimiento de tareas, retrasos considerables de tiempo, etc.
Mitigación: <ul style="list-style-type: none"> ■ Acordar un espacio de tiempo para resolver dudas. ■ Dar a conocer las actividades que cada integrante esta realizando. ■ Mayor comunicación entre los integrantes.

Tabla 3.26: Plan de contingencia correspondiente al riesgo 03.

Hoja de información del riesgo
ID: 04.
Descripción: Retraso en la obtención de recursos lingüísticos necesarios para poder llevar a cabo el proyecto.
Contexto: Pudiera darse el caso en que los permisos para poder obtener los recursos lingüísticos necesarios llegaran a retrasarse.
Mitigación: <ul style="list-style-type: none"> ■ Buscar un <i>corpus</i> externo con licencia para su uso.

Tabla 3.27: Plan de contingencia correspondiente al riesgo 04.

Hoja de información del riesgo
ID: 05.
Descripción: Falta de recursos lingüísticos
Contexto: Se da el caso de que la información obtenida de una fuente de datos no sea suficiente.
Mitigación: <ul style="list-style-type: none"> ■ Buscar un <i>corpus</i> externo con licencia para su uso. ■ Utilizar herramientas como BERT.

Tabla 3.28: Plan de contingencia correspondiente al riesgo 05.

Hoja de información del riesgo
ID: 06.
Descripción: Ambigüedad de los recursos lingüísticos obtenidos.
Contexto: Los recursos lingüísticos obtenidos son ambiguos y dificultan la correcta funcionalidad del proyecto.
Mitigación: <ul style="list-style-type: none"> ■ Buscar un <i>corpus</i> externo con licencia para su uso. ■ Utilizar herramientas como BERT.

Tabla 3.29: Plan de contingencia correspondiente al riesgo 06.

Hoja de información del riesgo
ID: 07.
Descripción: Documentación e información escasa
Contexto: Debido a la variedad tecnológica, librerías y la complejidad que este proyecto demanda, será de indispensable necesidad la adecuada documentación por parte de los desarrolladores de las tecnologías a utilizar
Mitigación: <ul style="list-style-type: none"> ■ Revisar a detalle la documentación oficial proveída por el desarrollador de la tecnología ■ Revisar los ejemplos de la implementación de la tecnología ■ Seguir las buenas practicas

Tabla 3.30: Plan de contingencia correspondiente al riesgo 07.

Hoja de información del riesgo
ID: 09.
Descripción: Priorizar temas ajenos al proyecto.
Contexto: Sean temas académicos, personales o profesionales, los integrantes del equipo no le dan la importancia necesaria al desarrollo del sistema.
Mitigación: <ul style="list-style-type: none"> ■ Respetar los tiempos de desarrollo. ■ Mantener informados a los miembros del equipo.

Tabla 3.31: Plan de contingencia correspondiente al riesgo 09.

Capítulo 4

Diseño del prototipo

4.1. Metodología

La metodología ágil *Team Data Science Process* es adecuada para el prototipo a desarrollar, la cual provee los siguientes *milestones* para desarrollar el modelo del agente conversacional:

1. Una fase enfocada al entendimiento del negocio.
2. Una fase enfocada a la adquisición de los recursos y el entendimiento de estos.
3. Una fase enfocada al modelado empleando los datos obtenidos en la fase anterior.
4. Una fase enfocada al despliegue en sus respectivos ambientes.

Entendimiento del negocio: Esta fase se trata de identificar las variables clave para definir los objetivos del modelo. La correcta identificación de estas impactaran directamente en el éxito que pueda obtener el prototipo, tanto en su uso, viabilidad y desempeño. Para alcanzar este objetivo es indispensable:

Definir objetivos: Identificar las variables objetivo del modelo a implementar, basándose en las métricas para medir el grado de éxito ya sea por el área de oportunidad debido a la ausencia o mala implantación de una solución o por un análisis factible del negocio para así, desarrollar un plan de hitos los cuales se les asignará a miembros del equipo adjudicándoles ciertas responsabilidades.

Identificar las fuentes de datos: Pueden ser numerosas y variables. En el caso de este agente conversacional su principal fuente de datos serán las conversaciones de la página oficial de Facebook de la Escuela Superior de Cómputo **ESCOM IPN MX**.

Adquisición de recursos: Habiendo identificando las fuentes de información contempladas en la fase de *Entendimiento del negocio* y de acuerdo a la regla de negocio número cuatro, se obtendrán los recursos lingüísticos los cuales se les aplicara un proceso de transformación

llamado “normalización” para obtener datos limpios y de calidad, dando lugar a un conjunto de datos relacionados con nuestras variables objetivo, estos datos serán almacenados y puestos a disposición para la siguiente fase.

Modelado: Una vez obtenidos todos los recursos lingüísticos gracias a la fase de *Adquisición de recursos*, se separará el conjunto en dos partes a los que llamaremos: “conjunto de entrenamiento” y “conjunto de prueba”, con una distribución del conjunto original del 80 % y 20 % respectivamente, para construir el modelo con base en los datos de entrenamiento haciendo uso de algoritmos y técnicas de aprendizaje automático y procesamiento del lenguaje natural, para así evaluar el modelo resultante con los datos de prueba. Si la precisión del modelo es suficiente se dará continuidad a la fase de *Despliegue*, en caso contrario se identificará la fase cuyo resultado haya influenciado negativamente en el modelo resultante y se remontará a la fase en cuestión para realizar los cambios pertinentes.

Despliegue: Se proveerá de una infraestructura para que nuestro modelo resultante pueda ser consumido por los servicios de *Facebook*.

4.2. Tecnologías

Python: Diseñado por Guido van Rossum en 1991, es un lenguaje de programación interpretado, dinámico y multiplataforma. Soporta programación orientada a objetos, programación imperativa y programación funcional.

NTKL: Es un conjunto de bibliotecas y programas enfocadas al procesamiento del lenguaje natural.

GENSIM: Es una biblioteca para el modelado de temas no supervisados y el procesamiento del lenguaje natural, que utiliza el aprendizaje automático moderno de estadística.

Docker: Desarrollado por Solomon Hykes en marzo del 2013, es un software que ayuda a empaquetar aplicaciones dentro de un contenedor en el que se especifican las configuraciones y sus dependencias necesarias para que la aplicación embebida pueda correr, ofreciendo portabilidad, eficiencia, optimización de tiempo y recursos.

Docker compose: Es un gestor de contenedores Docker con la característica de poder desplegar todo un entorno partir de un archivo YALM en el que se especifica paso por paso como conectar y configurar estos servicios.

MySQL: Creada por David Axmark, Allan Larsson y Michael Widenius en 1995, es un gestor de bases de datos relacional.

MongoDB: Desarrollado en el 2007 por la empresa *10gen Inc*, es una base de datos distribuida de tipo NoSQL orientada a documentos en la cual se almacenan datos no estructurados en formato BSON.

GitHub: Creado en el 2010, es un repositorio de código que utiliza el sistema de control de versiones GIT.

Git: Creado por Linus Torvalds, es un software de control de versiones en el se lleva un registro de los cambios efectuados por los desarrolladores a los archivos de código fuente y así, llevar un mayor control en las versiones sobre las que se trabaja cada rama del proyecto.

L^AT_EX: Escrito por Leslie Lamport en 1984, está formado por un gran conjunto de macros de Tex el cual fue creado por Donald Knuth en 1978. Es un sistema de composición de textos, esta enfocado a la creación de documentos de gran calidad tipográfica.

Graph API: es el núcleo de la pagina de Facebook y cuenta con las funciones de poder leer y escribir datos en Facebook, permite la conexión de aplicaciones externas que quieran consumir sus distintos servicios.

AWS: plataforma cloud con servicios tipo *IaaS*, *PaaS* y *SaaS*.

4.3. Modelado

El propósito de estos diagramas es para tener un mayor entendimiento de la funcionalidad y el comportamiento de nuestro sistema. A continuación de presentan los casos de uso, diagramas de procesos, diagramas de clases, el diagrama entidad-relación y finalmente los diagramas de infraestructura que conforman el de prototipo de agente conversacional.

4.3.1. Diagramas de casos de uso

Los diagramas de casos de uso sirven para especificar el comportamiento y comunicación del sistema mediante su interacción con los usuarios y otros sistemas. Nuestro agente conversacional a lo largo de su ciclo de vida tendrá interacción tanto con usuarios como con sistemas externos para lograr su perfecta ejecución. La figura 4.1 describe como el usuario administrador ejecuta el micro servicio que lleva por nombre *Scraper*, el cual se emplea como un modulo de extracción de mensajes, este se conecta con *Graph API* gracias a un *token* de acceso, permitiendo descargar los mensajes para después almacenarlos en el sistema local de archivos de la computadora o en una base de datos no relacional.

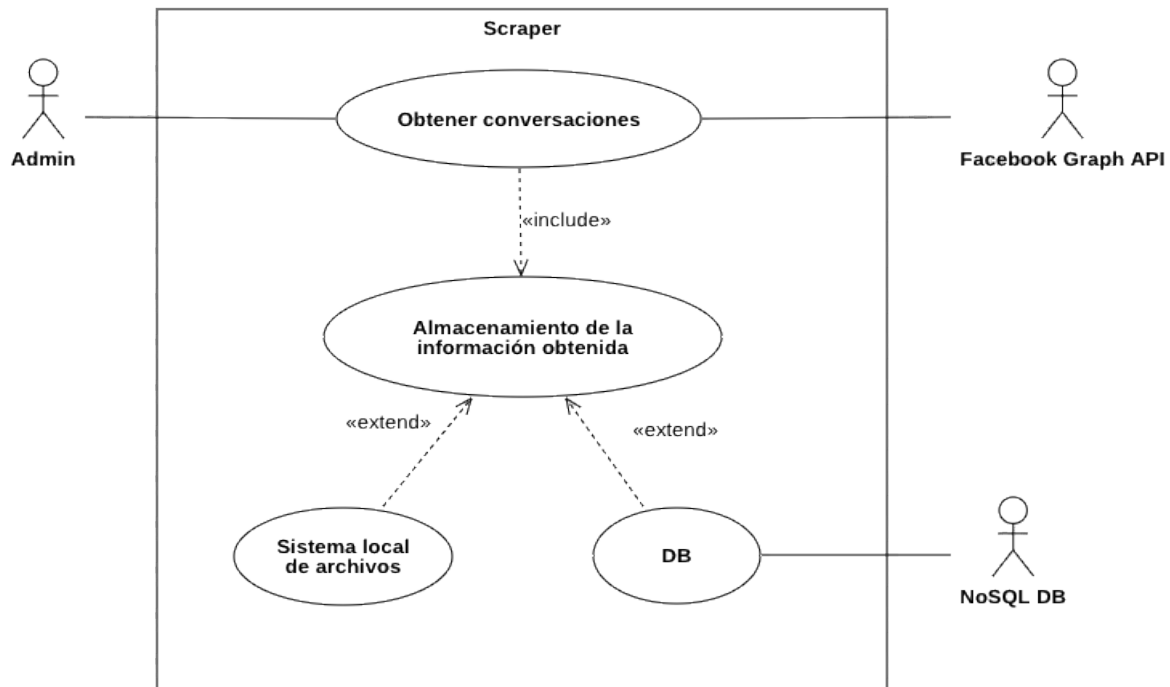


Figura 4.1: Diagrama de casos de uso para el modulo de extracción denominado Scraper.

La figura 4.2 describe una de las funcionalidades del modulo de procesamiento de lenguaje natural, en esta parte del proceso se trata de establecer una conexión con la base de datos no relacional donde los datos crudos obtenidos por *Scraper* han sido almacenados, una vez que se han recuperados estos datos se realiza un paso denominado estandarización de texto, en el cual con ayuda de diccionarios de lemas y *stopwords* se limpian los documentos dejando únicamente aquellas palabras relevantes, este proceso tiene muchas fases y cada una de estas son almacenadas en una base de datos relacional (véase la figura 4.7).

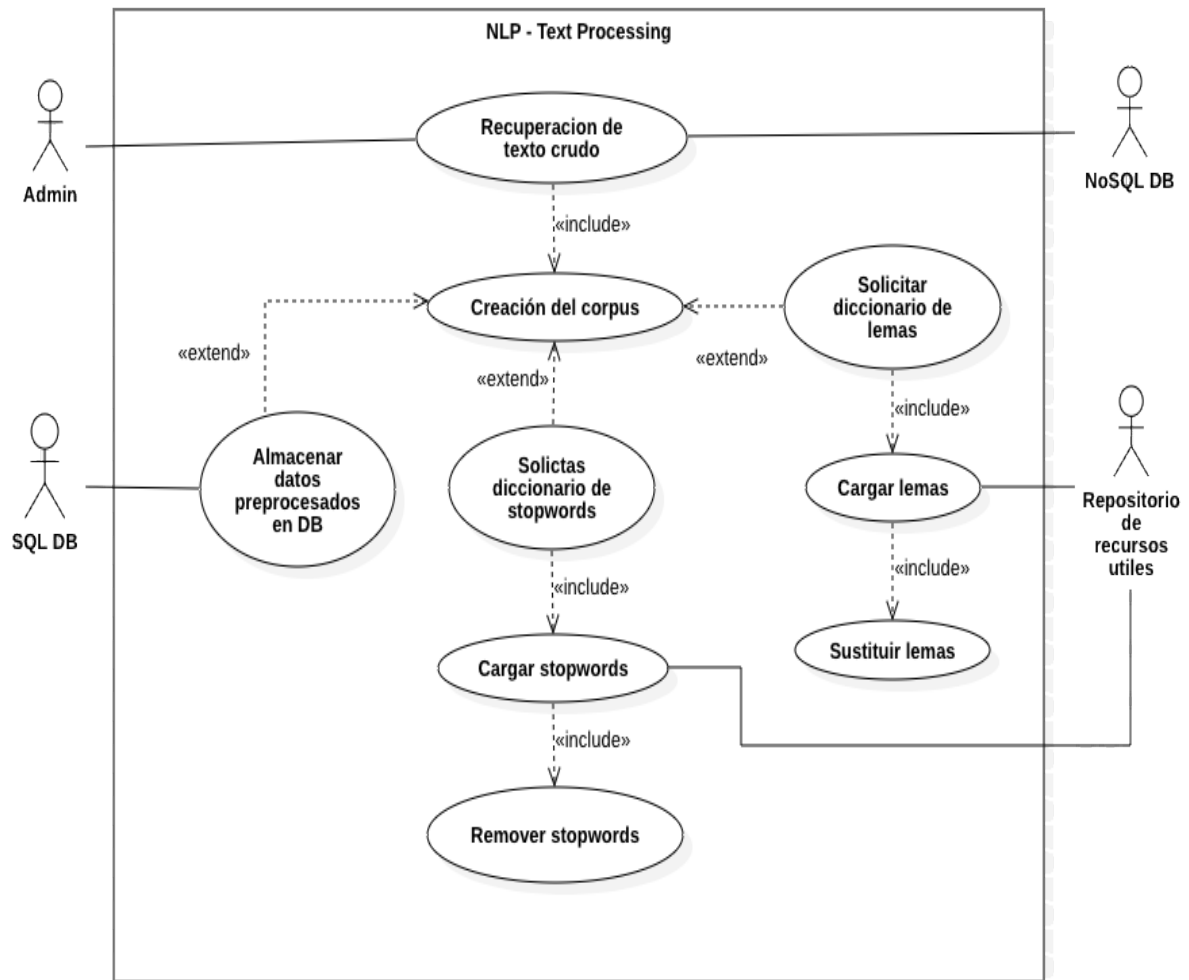


Figura 4.2: Diagrama de casos de uso para el modulo de pre-procesamiento de texto.

La figura 4.3 describe otra de las funcionalidades del modulo de procesamiento de lenguaje natural, en el cual se obtienen los datos preprocesados de la base de datos relacional para aplicar el algoritmo LDA y poder extraer los temas frecuentes tratados dentro del texto.

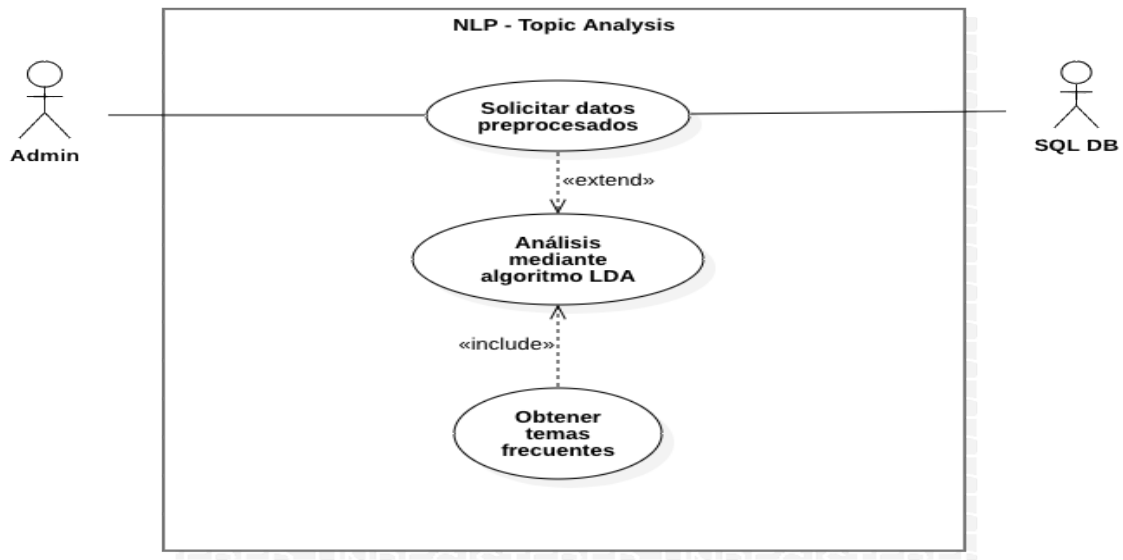


Figura 4.3: Diagrama de casos de uso para el modulo de *topic analysis*.

La figura 4.4 describe la funcionalidad de graficación del modulo de procesamiento de lenguaje natural, el cual toma los datos que el algoritmo **LDA** arroja para graficarlos y que el usuario administrador pueda realizar un análisis visual de estos datos.

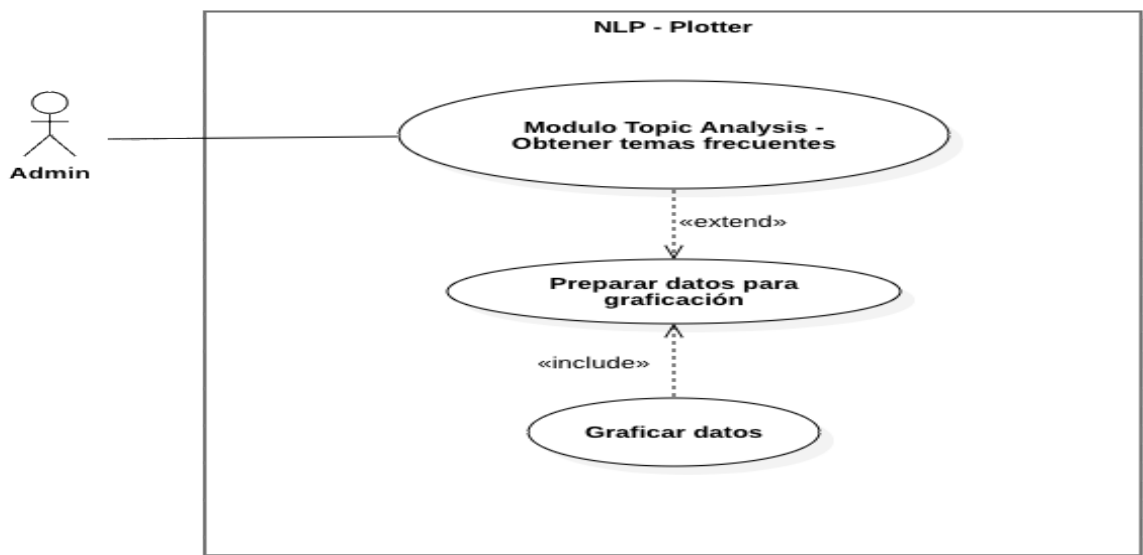


Figura 4.4: Diagrama de casos de uso para el modulo de graficación.

4.3.2. Diagramas de procesos

Estos diagramas sirven para representar de manera gráfica un algoritmo o proceso. La figura 4.5 muestra con mas detalle la funcionalidad del modulo *Scraper* o de extracción. En primer lugar, este modulo espera recibir una serie de variables de entorno y banderas de entrada por parte del usuario, si la bandera de “ayuda” es percibida se muestra un mensaje de como usar el programa y se termina la ejecución, si no, entonces comienza el mapeo del resto de banderas, cabe destacar que estas banderas tienen un valor por defecto si el usuario no ingresa alguna. Una vez hecho esto entonces se busca por la bandera “base de datos”, la cual si se encuentra en “verdadero” intentara establecer una conexión con la base de datos para almacenar ahí los mensajes, en dado caso de que la conexión no sea exitosa esta bandera automáticamente se cambia a “falso” para prevenir futuros fallos y este registro se guarda en el sistema de *logs*. Posteriormente se intenta establecer una conexión con *Graph API*, si el *token* es invalido se muestra un mensaje de error personalizado, se termina la ejecución y este error es almacenado en el sistema de *logs*, en caso de una conexión exitosa comienza la descarga de mensajes, por cada mensaje se pregunta por la bandera de *base de datos* para que, en caso de estar en “verdadero”, el mensaje sea almacenado en la base de datos no relacional, de lo contrario el mensaje será almacenado en el sistema local de archivos. Una vez terminada la descarga de mensajes se termina la ejecución de este módulo.

La figura 4.6 muestra el flujo del módulo de procesamiento de lenguaje natural. En primer lugar, este modulo espera recibir una serie de variables de entorno y banderas de entrada por parte del usuario, si la bandera de “ayuda” percibida se muestra un mensaje de como usar el programa y se termina la ejecución, si no, entonces comienza el mapeo del resto de banderas, cabe destacar que estas banderas tienen un valor por defecto si el usuario no ingresa alguna. Posteriormente se intenta conectar con la base de datos relacional, en caso de una conexión fallida se muestra un mensaje de error personalizado, el registro se guarda en el sistema de *logs* y se termina la ejecución, de lo contrario, busca que la bandera de *pre-procesamiento* se encuentre en “verdadero”, de ser así se intenta establecer una conexión con la base de datos no relacional, dándose el caso de una conexión fallida se muestra un mensaje de error personalizado, el registro se guarda en el sistema de *logs* y se termina la ejecución, de lo contrario, se descargan los mensajes en su formato crudo para que por medio de diccionarios de lemas y *stopwords* estos datos sean normalizados, cada uno de estos mensajes será guardado en la base de datos relacional. Luego, si la bandera de *análisis LDA* se encuentra en “verdadero” y dada la columna de la base de datos relacional (véase la figura 4.7) especificada en las opciones de entrada, se procederá a descargar estos datos para realizar el análisis de tópicos, de lo contrario se finalizara la ejecución. Una vez realizado el análisis con el algoritmo **LDA** se pregunta si la bandera de graficación esta en “verdadero”, de ser así, entonces cada tópico propuesto por el algoritmo **LDA** sera graficado para realizar un análisis visual, en caso contrario los tópicos descubiertos serán almacenados en un archivo local para su futuro análisis, con este paso se termina la ejecución del modulo de PLN.

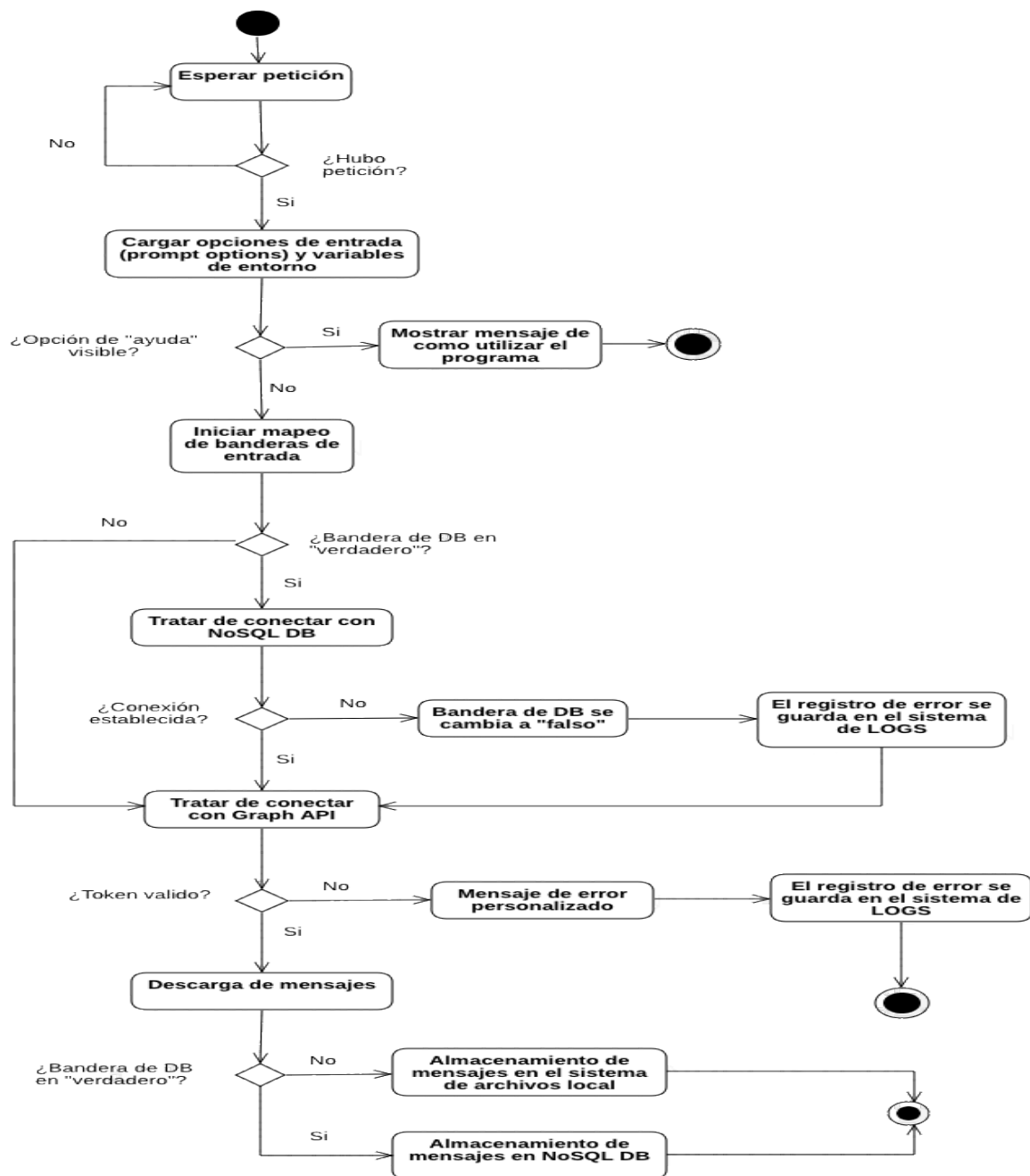


Figura 4.5: Diagrama de procesos del módulo de extracción.

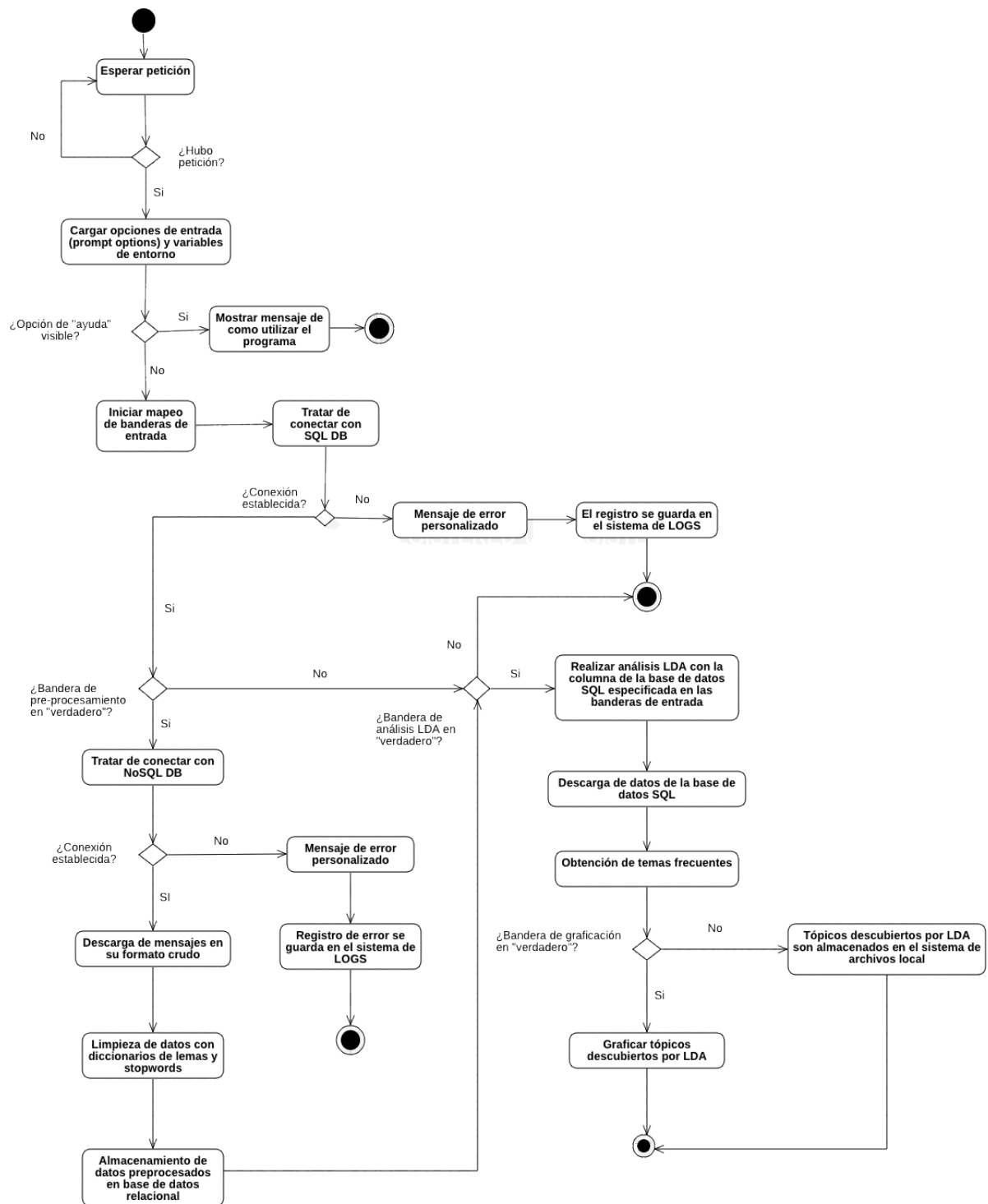


Figura 4.6: Diagrama de procesos del módulo de procesamiento de lenguaje natural.

4.3.3. Diagramas Entidad - Relación

El siguiente diagrama entidad-relación describe la base de datos relacional que será utilizada por el modulo de procesamiento de lenguaje natural, dicha base de datos cuenta únicamente con 2 tablas; *Conversation*, que guarda una referencia de la conversación a la cual cada mensaje pertenece y *Message* que almacena el mensaje en su formato crudo así como en distintos niveles de normalización.

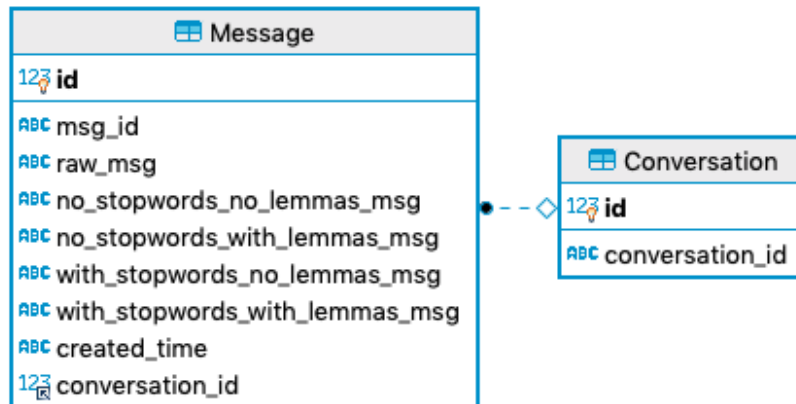


Figura 4.7: Diagrama entidad-relación de la base de datos relacional.

4.3.4. Diagramas de clases

Los diagramas de clases mostrados en las figuras 4.8 y 4.9, nos ayudan a entender de manera visual las clases que los módulos de *extracción (scraper)* y *procesamiento de lenguaje natural* utilizan, sus atributos, operaciones y las relaciones entre los objetos.

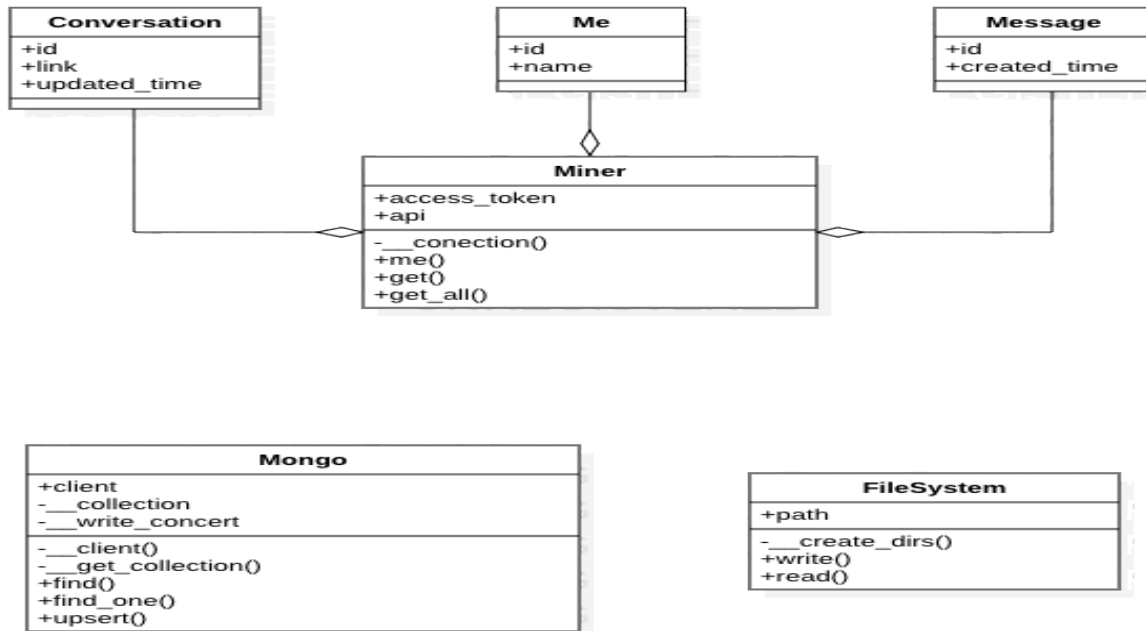


Figura 4.8: Diagrama de clases del módulo *Scraper* o de extracción.

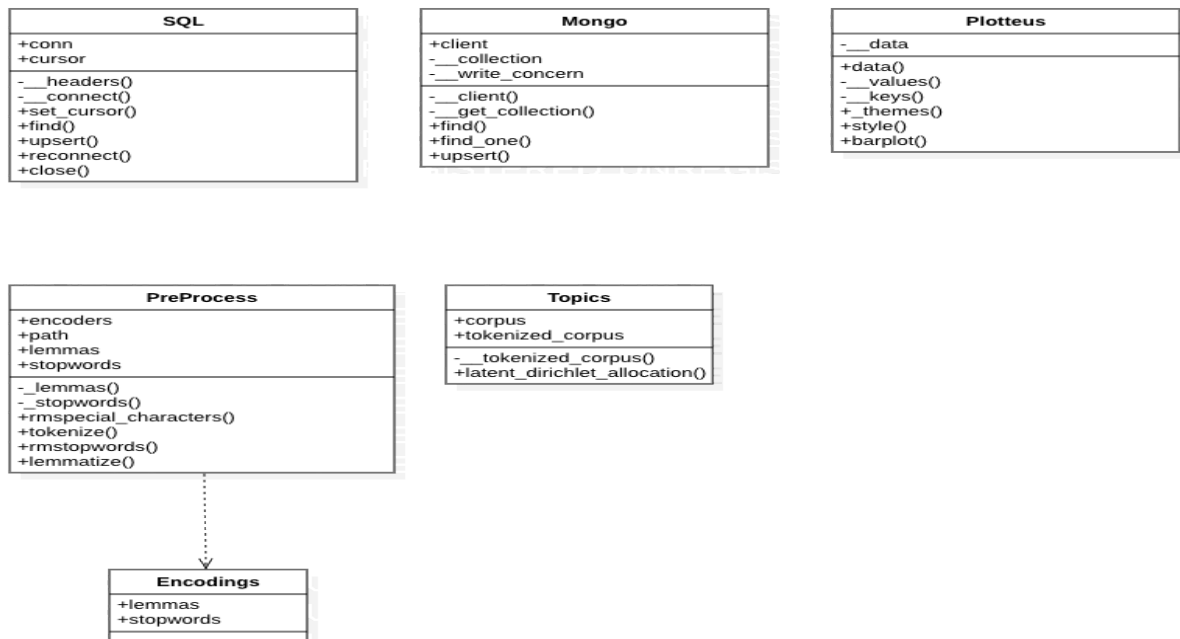


Figura 4.9: Diagrama de clases del módulo de procesamiento de lenguaje natural.

4.3.5. Diagramas de secuencia

El diagrama de secuencia de la figura 4.10 se muestra el proceso por el cual viaja una petición desde Facebook *Graph API* hasta la infraestructura y como es tratado el dato:

1. La petición entra a una frontera *AWS ACL* en el cual revisara que el no usuario se encuentre dentro de una black list
 - a) si esta: no permitirá el paso de trafico proveniente de ese usuario
 - b) si no esta: redireccionará el trafico a *API gateway*
 - 1) Llega a la frontera *API gateway* el cual encolara la función control *AWS Lambda* "StateMachine" dentro de la entidad *AWS SQS*
 - 2) La entidad *AWS SQS* desencolara las anteriores peticiones de funciones control *AWS Lambda* una por una de manera cronológica, "StateMachine" será desplegado al llegar al principio del encolador.
 - 3) La función control *AWS Lambda* "StateMachine" encolara dentro de una entidad *AWS SQS* la función correspondiente al estado de la pregunta.
 - a' si el usuario es nuevo o nueva pregunta: la función a encolar será "Welcome"
 - b' La entidad *AWS SQS* siguiendo el mismo principio del punto 3 invocará la funcion control *AWS Lambda* "Welcome"

-
- c'* si el mensaje contenía una pregunta, se analizará y guardará el contexto dentro de la entidad *AWS Elasticache*
 - d'* La entidad *AWS Elasticache* devolverá el estado de la petición a la funcion control *AWS Lambda "Welcome"* tras haber guardado el contexto
 - e'* La funcion control *AWS Lambda "Welcome"*.^{en}viará la respuesta a la frontera *API gateway*
 - f'* si es sobre preguntas pasadas: la función a encolar será *"FAQs"*
 - g'* La entidad *AWS SQS* siguiendo el mismo principio del punto 3 invocará la funcion control *AWS Lambda "FAQs"*
 - h'* La funcion control *AWS Lambda "FAQs"* solicitará a la entidad *AWS Elasticache* el contexto
 - i'* La entidad *AWS Elasticache* devolverá el contexto a la función control *AWS Lambda "FAQs"*
 - j'* La funcion control *AWS Lambda "FAQs"* tras analizar la pregunta guardara el contexto dentro de la entidad *AWS Elasticache*
 - k'* si el mensaje contenía una pregunta, se analizará y guardará el contexto dentro de la entidad *AWS Elasticache*
 - l'* La entidad *AWS Elasticache* devolverá el estado de la petición a la funcion control *AWS Lambda "FAQs"* tras haber guardado el contexto
 - m'* La funcion control *AWS Lambda "FAQs"*.^{en}viará la respuesta a la frontera *API gateway*
- 4) La frontera *API gateway* actualizara las ACL de la frontera *AWS ACL*
 - 5) La frontera *API ACL* se comunicará con *Facebook Graph API* y retornara el resultado del análisis.
 - 6) El paso 1.a.1 se repite hasta que la función control *AWS Lambda "StateMachine"* pasa al estado *"Fairwell"*
 - 7) La petición entra a una frontera *AWS ACL* siguiendo la logica del paso 1
 - 8) Llega a la frontera *API gateway* el cual encolara la función control *AWS Lambda "StateMachine"* dentro de la entidad *AWS SQS*
 - 9) La entidad *AWS SQS* siguiendo el mismo principio del punto 3 invocará la funcion control *AWS Lambda "StateMachine"*
 - 10) La función control *AWS Lambda "StateMachine"*.^{en}colara dentro de una entidad *AWS SQS* la función *"Fairwell"*
 - 11) Llega a la frontera *API gateway* el cual encolara la función control *AWS Lambda "Fairwell"* dentro de la entidad *AWS SQS*
 - 12) La funcion control *AWS Lambda "Fairwell"*.^{en}viará la respuesta a la frontera *API gateway*
 - 13) La frontera *API gateway* actualizara las ACL de la frontera *AWS ACL*
 - 14) La frontera *API ACL* se comunicará con *Facebook Graph API* y retornara la respuesta
-

2. La entidad *AWS SQS* desencolara los mensajes de forma ordenada lanzando otra función control *AWS Lambda* para cada una de estas.
3. La función control *AWS Lambda* analizara la petición y buscará en la entidad *AWS Elacti-cache* un contexto de preguntas pasadas.
4. La entidad *AWS Elasticache* buscara si existe un contexto previo, si existe entonces retornará este, en caso contrario solo mandará una notificación.
5. La función control *AWS Lambda* analizara el mensaje obtenido con base en el resultado de la entidad *AWS Elacti-cache* y devolverá el resultado a la frontera *API gateway*.
6. La frontera *API gateway* se comunicará con *Facebook Graph API* y retornara el resultado del análisis.

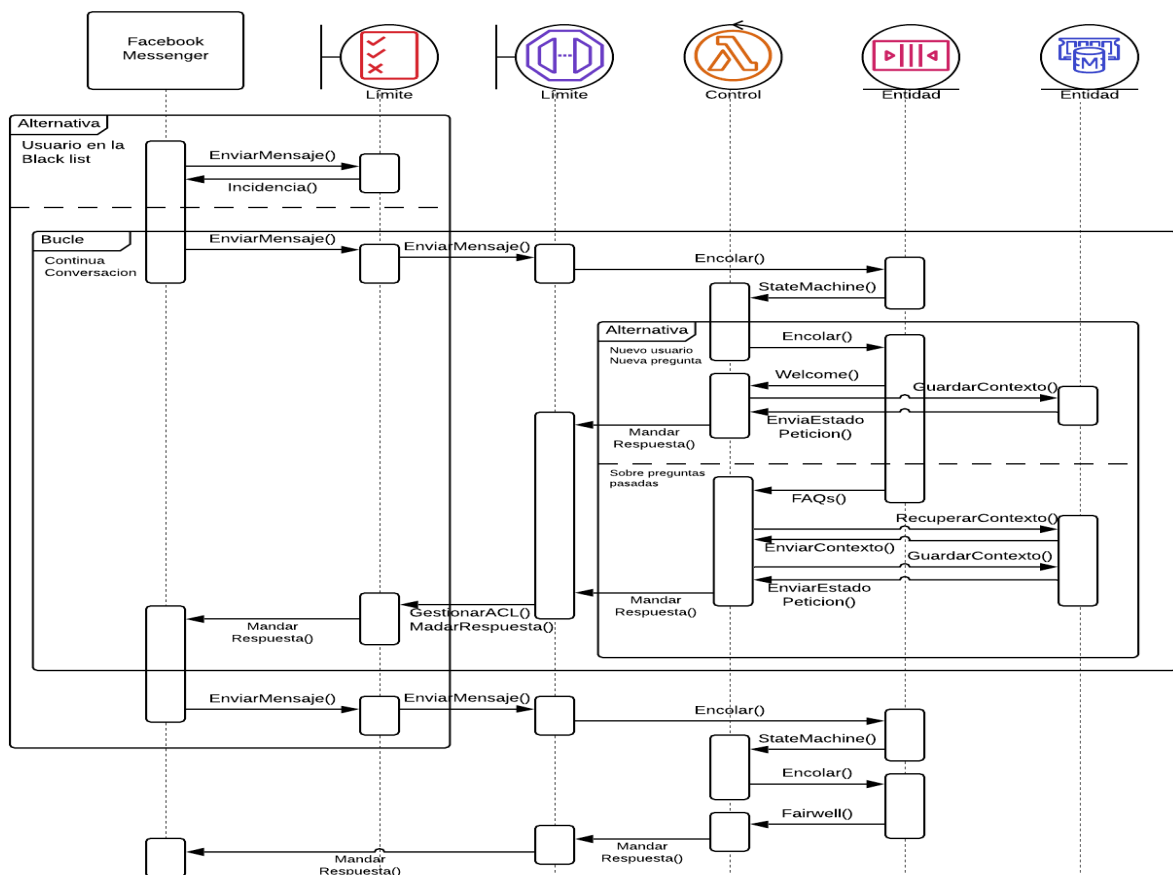


Figura 4.10: Secuencia por la que viaja una pregunta dentro de las tecnologías de AWS.

4.3.6. Diagramas de infraestructura

El diagrama de infraestructura de la figura 4.11 se muestra la infraestructura sobre la cual se ejecutará el prototipo de agente conversacional, la implementación de este será sobre el proveedor de computo en la nube AWS.

AWS Route 53 proveerá de los certificados necesarios para poder comunicar la tecnología *AWS API Gateway* con *Facebook Graph API* por medio de un canal bidireccional con las funciones de *AWS Lambda* para mandar y recibir peticiones. *AWS Lambda* ejecutará bloques de código cuando sea invocado retornando un resultado, *AWS SQS* proveerá un sistema de encolamiento de mensajes de las peticiones entrantes y se definirá la acción de invocar una función de *AWS Lambda* a la hora de desencolar estos. *AWS ElastiCache* proveerá una memoria temporal definida por el administrador, en el que se almacenan objetos de alta transaccionalidad de tipo clave valor. En el ambiente de despliegue, si se realiza un cambio sobre el código de una función de *AWS Lambda* está se actualizará y se desplegará de tal manera que las nuevas funciones de *AWS Lambda* corran con estos cambios.

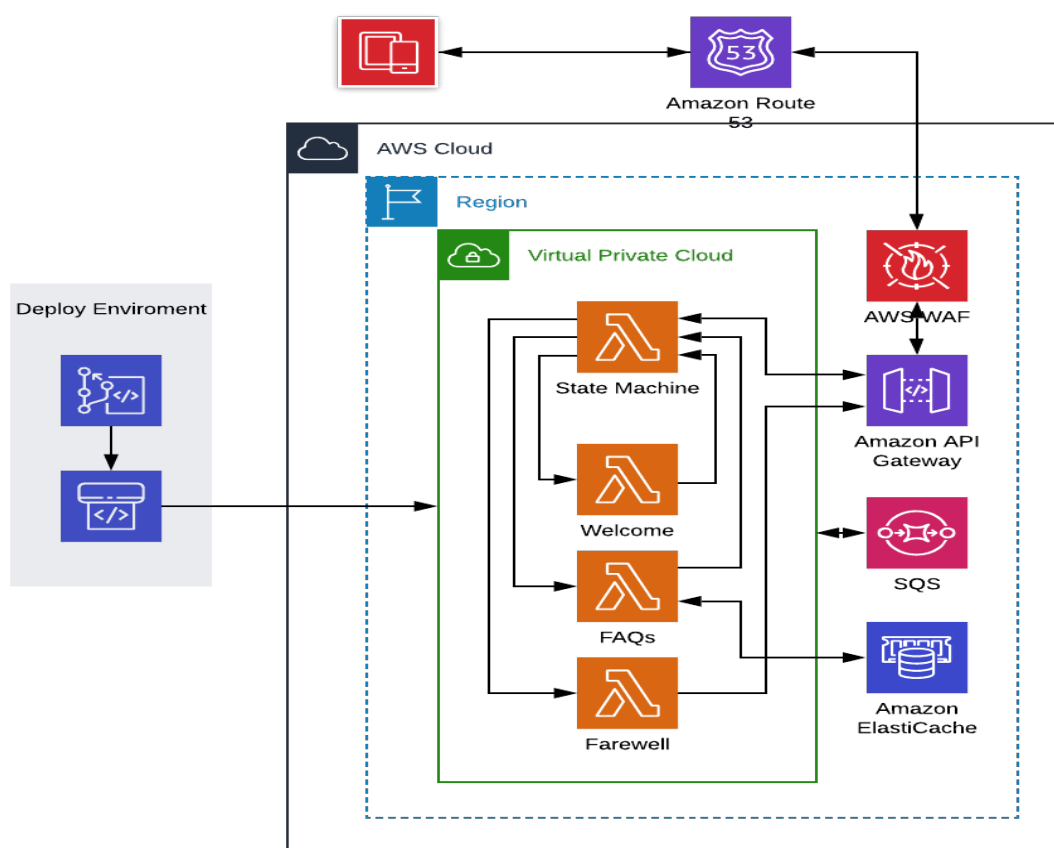


Figura 4.11: Infraestructura del agente conversacional

El diagrama de infraestructura de la figura 4.16 muestra dos módulos del prototipo de agente

conversacional: Extracción y procesamiento de lenguaje natural. En el módulo de extracción se hace uso de un equipo de cómputo con salida a internet, se comunicará con *Graph API* y obtendrá los datos pertinentes, sobre este equipo se ejecutará un contenedor de Docker con persistencia en el que correrá una base de datos NoSQL en una subnet privada con comunicación de tipo puente con el equipo de cómputo y se almacenarán los datos obtenidos. En el módulo de NLP, se hace uso de un equipo de cómputo en el que se ejecutarán dos contenedores con persistencia en una subnet privada con comunicación de tipo puente con el equipo de cómputo, los datos almacenados en la base de datos NoSQL pasaran por un submódulo de preprocesado y serán almacenados en la base de datos SQL. Con base a estos datos se ejecutará el submódulo de análisis de tópicos, al finalizar se comunicara internamente con el equipo y este ejecutará el submódulo de graficación.

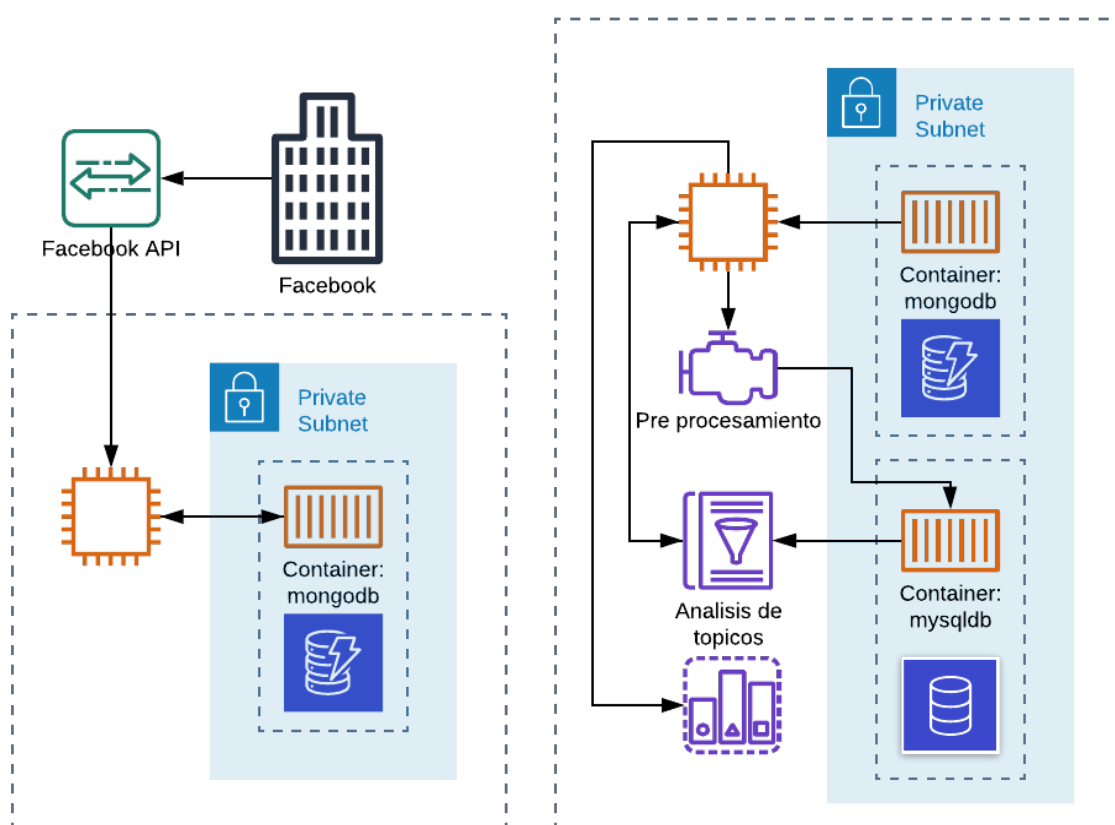


Figura 4.12: Infraestructura de los modulo de extracción y NLP respectivamente

4.3.7. Construcción del modelo

Es importante remarcar el hecho de que nuestro agente conversacional estará basado en un dominio de espacio cerrado enfocado a la resolución de preguntas frecuentes correspondientes

al área de administración educativa, las respuestas obtenidas por este, estarán basadas en un modelo generativo. Se plantea el uso de mecanismos de atención el cual se encargará de dirigir la conversación hacia una pregunta específica.

La arquitectura interna del modelo esta basada en *Seg2Seg* el cual cuenta con tres capas de redes neuronales recurrentes con diferentes conjuntos de parámetros, una capa se encargará de codificar la secuencia de la fuente y la otra en decodificar la secuencia codificada por la anterior red neuronal

Codificador: Se encarga de obtener los datos de la pregunta y se entrenará con ellos, pasando la información a cada una de sus neuronas por medio de una cinta de memoria, al llegar al final, se mandará el estado junto con una cadena de inicio a la siguiente red neuronal, a partir de estos datos se generaran las respuestas.

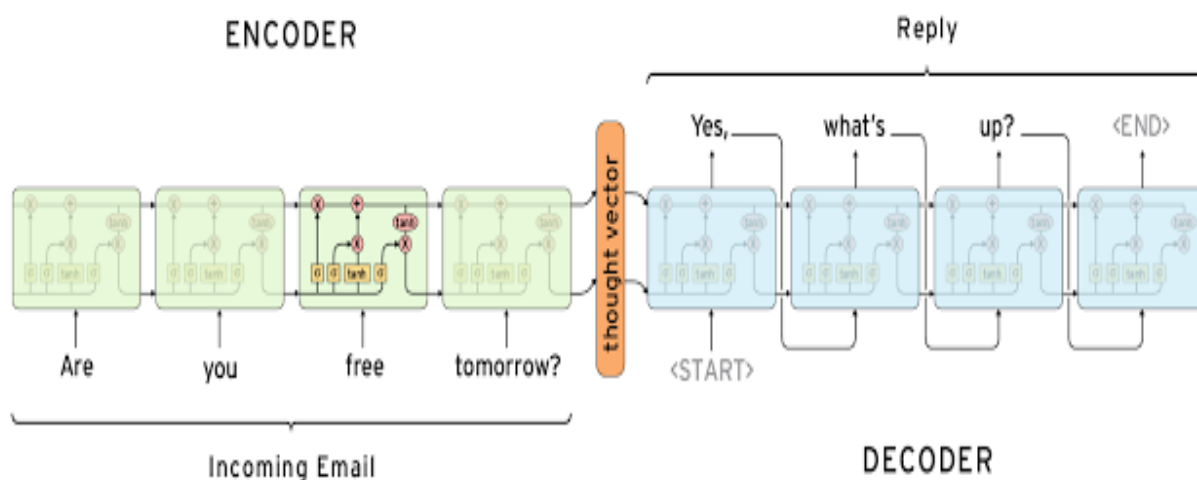
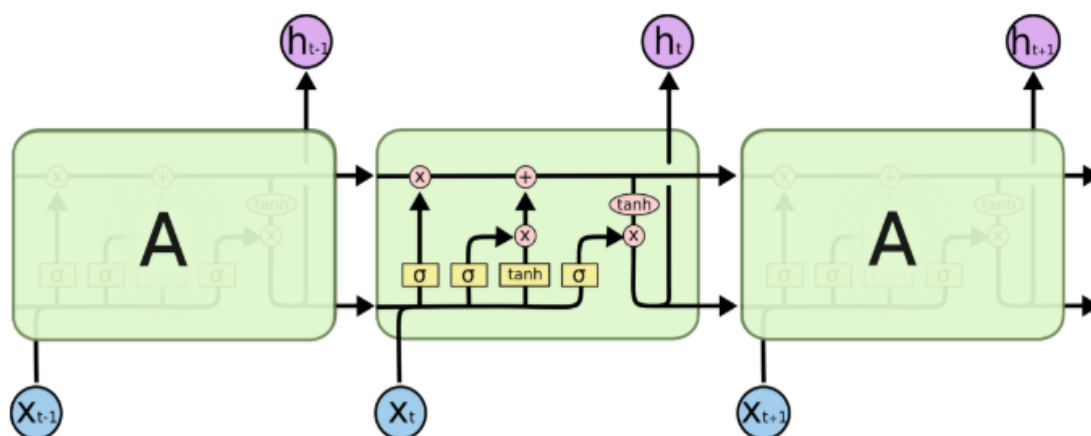


Figura 4.13: Arquitectura Seg2Seg

Las redes neuronales dentro de estas dos capas son del tipo LSTM los cuales son una variante de las RRN, cuentan con una célula de memoria el cual esta compuesta por 3 principales compuertas:

Forget gate: se encarga de analizar la entrada junto al estado de la LSTM, y pasar estos por medio de una fusión de activación de tipo sigmoidal, se eliminaran los datos que no serán usados en adelante.

Output gate: después de aplicar las transformaciones de la cinta con la salida de la Forget gate y transformando esta con el resultado de Update gate, transformaremos el resultado con una función Tanh a la cinta de memoria y la pasaremos por una compuerta XOR con la entrada junto al estado de la LSTM, ya pasados por la fusión de activación de tipo sigmoidal, generaremos así la salida del nuevo estado.



The repeating module in an LSTM contains four interacting layers.

Figura 4.14: Arquitectura de la LSTM

La tercera capa es un mecanismo de atención, enfocado en generar respuestas recomendadas con base al contexto de los estados de cada neurona del codificador junto a la intención obtenida por el modulo de LDA.

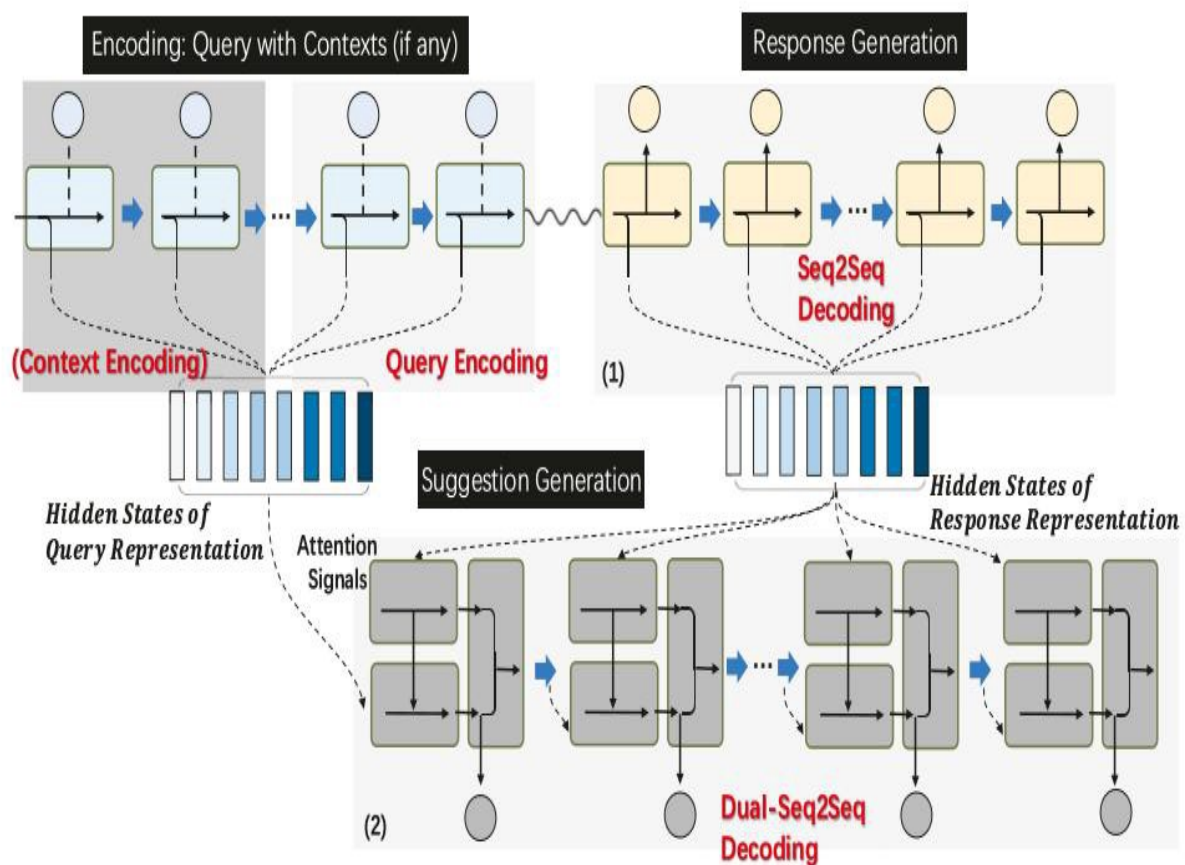


Figura 4.15: Arquitectura del mecanismo de atención

4.3.8. Resultados esperados

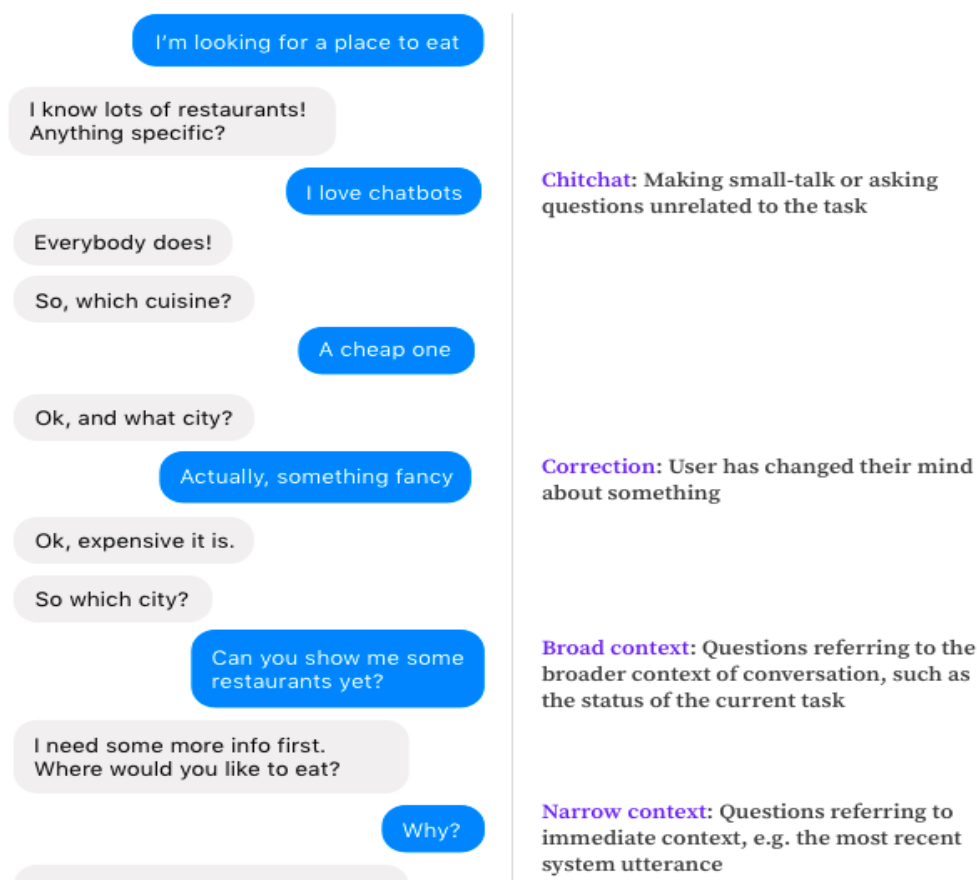


Figura 4.16: Resultados esperados usando la técnica de Seg2Seg con mecanismos de atención

Cabe destacar que este modelo estará delimitado por una cierta cantidad de preguntas a resolver, como ya se mencionó, será un sistema cerrado enfocándose principalmente en 3 tipos de peticiones:

- Convocatorias de inscripción, tanto semestrales como Trabajos Terminales, protocolo de TT, electiva, optativa y postgrados.
- Información para la baja de unidades de aprendizaje así como baja temporal o definitiva.
- Información general para alumnos de nuevo ingreso sobre la Escuela Superior de Cómputo.

Capítulo 5

Resultados

Con el fin de demostrar la correcta funcionalidad de los módulos descritos en el capítulo 4, en el presente apartado se exponen los diversos resultados obtenidos a lo largo del desarrollo del prototipo. Por discreción, los datos sensibles fueron enmascarados.

5.1. Módulo de extracción (*Scraper*)

Este módulo basa su funcionalidad en la conectividad exitosa con Graph API de Facebook para poder dar lugar a la descarga de conversaciones que serán empleadas por el agente conversacional después de haber sido normalizadas. Como se menciona en la sección 4, al descargar las conversaciones se busca que exista una base de datos no relacional disponible, de ser así entonces las conversaciones serán almacenadas en dicha base de datos (figura 5.1), en caso contrario en el sistema local de archivos (figura 5.2, 5.3 y 5.4).



Figura 5.1: Datos crudos obtenidos de *Graph API* almacenados en base de datos *NoSQL*.

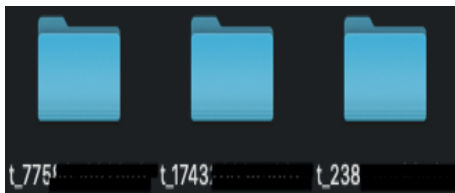


Figura 5.2: Datos crudos obtenidos de *Graph API* almacenados en el sistema local de archivos por conversación.



Figura 5.3: Datos crudos obtenidos de *Graph API* almacenados en el sistema local de archivos por mensaje.

```

1  {
2    "message": "Buenas tardes.\n\u00bfLa pl\u00e9tica de Facebook el 24 de octubre va dirigida a alumnos de...",
3    "from": {
4      "name": "Allan ...",
5      "email": "...@facebook.com",
6      "id": "..."
7    },
8    "created_time": "2019-10-15T15:20:02+0000",
9    "to": {
10     "data": [
11       {
12         "name": "ESCOM IPN MX",
13         "email": "...@facebook.com",
14         "id": "..."
15       }
16     ]
17   },
18   "id": "...",
19   "conversation_id": "..."
20 }

```

Figura 5.4: Datos crudos obtenidos de *Graph API* almacenados en el sistema local de archivos visto desde un editor de texto.

5.2. Módulo de procesamiento de lenguaje natural

Una vez que se tiene una base de datos *NoSQL* disponible y además, alimentada con los datos crudos gracias a *Graph API*, el módulo de procesamiento de lenguaje natural hace disposición de estos datos para limpiarlos y almacenarlos en una base de datos relacional (véase la figura 4.7) en cada una de sus etapas de normalización como se puede ver en la figura 5.5, es decir:

- *Mensaje crudo:* Mensaje tal cual es extraído de *Graph API*.
- *Mensaje con stopwords y sin lemas:* Mensaje sin cualquier carácter que no pertenezca al abecedario latino.
- *Mensaje sin stopwords y sin lemas:* Mensaje sin cualquier carácter que no pertenezca al abecedario latino y sin stopwords.
- *Mensaje sin stopwords y con lemas:* Mensaje sin cualquier carácter que no pertenezca al abecedario latino, sin stopwords y lematizado.
- *Mensaje con stopwords y con lemas:* Mensaje sin cualquier carácter que no pertenezca al abecedario latino y lematizado.

5.2 Módulo de procesamiento de lenguaje natural

123	abc msg_id	abc raw_msg	abc no_stopwords_no	abc no_stopwords_with_len	abc with_stopwords_no	abc with_stopwords_with	abc created_time	123 cd
17		Buenas tardes, me inte	tardes interesa sale infc	interesar salir periodo enero	Buenas tardes me interesa s	buena tarde yo interesar	2019-11-03T21:3	11
18		Hola buenas tardes, ha	tardes tiempo leí escue	tiempo leer escuela trabajar	Hola buenas tardes hace tier	hola buena tarde hacer ti	2019-11-03T21:01	12
19		Buenas tardes! Soy un	tardes soy estudiante te	estudiante terminar estudio	Buenas tardes Soy un estudi	buena tarde ser un estud	2019-11-02T21:24	13
20		si es de 10 semestres	semestres promedio cu	semestre promedio cuánto r	si es de semestres el prome	si ser de semestre el pro	2019-11-02T01:41	14
21		porque en Wikipedia d	wikipedia dura semestr	durar semestre encontrar pl	porque en Wikipedia dice qu	porque en decir que a pa	2019-11-02T01:40	14
22		Cuánto dura la carrera	cuánto dura carrera ing	cuánto durar carrera sistem	Cuánto dura la carrera de in	cuánto durar la carrera d	2019-10-28T20:43	14
23		Quiero más informació	quiero información escu	información escuela	Quiero más información sob	querer más información s	2019-10-28T20:42	14
24		Quiero más informació	quiero información escu	información escuela	Quiero más información sob	querer más información s	2019-10-31T03:25	16
25		Hola, anteriormente es	anteriormente estudiab	anteriormente estudiar prob	Hola anteriormente estudiab	hola anteriormente estud	2019-10-29T22:52	17

Figura 5.5: Datos en diferentes niveles de normalización almacenados en la base de datos relacional.

123 id	abc conversation_id
1	t_1454
2	t_2728
3	t_2591
4	t_2448
5	t_1021
6	t_2417
7	t_2500
8	t_9576
9	t_1316
10	t_4646
11	t_7667

Figura 5.6: Tabla de conversaciones.

Una vez realizada la normalización, se extraen los datos de cada una de las columnas descritas anteriormente para poder realizar un análisis de tópicos, a continuación se muestran los resultados.

- De los mensajes con stopwords y sin lemas no se pudo obtener gran información puesto que las palabras que predominan son palabras vacías como artículos, pronombres, preposiciones, etc.

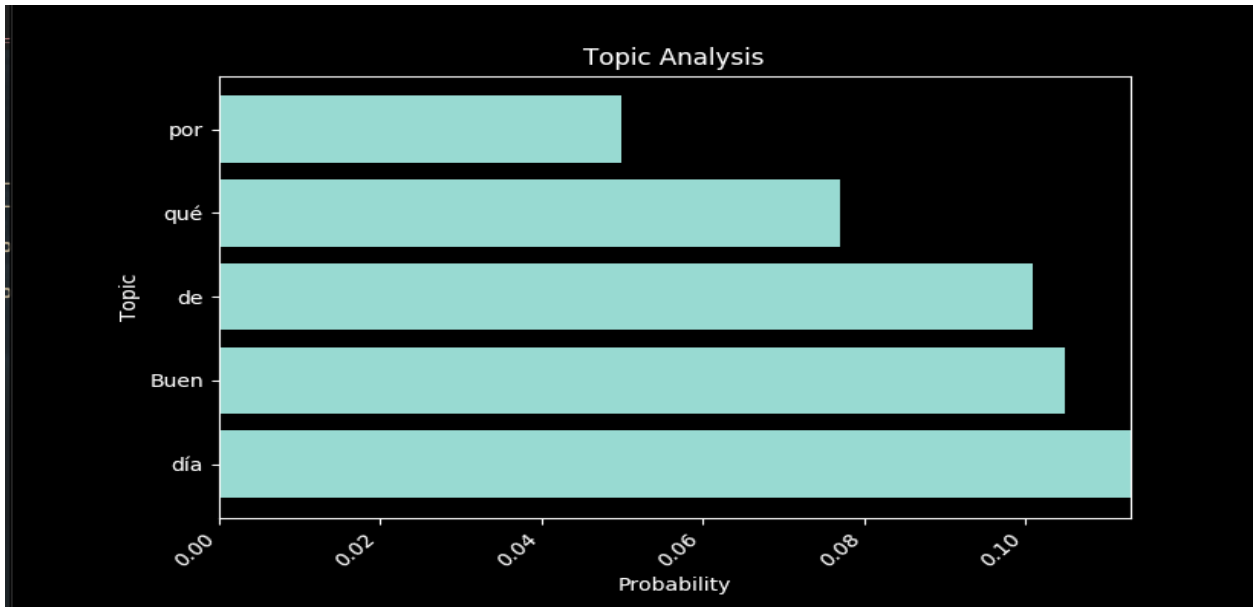


Figura 5.7: Resultado LDA con documentos con *stopwords* y sin lemas (a).

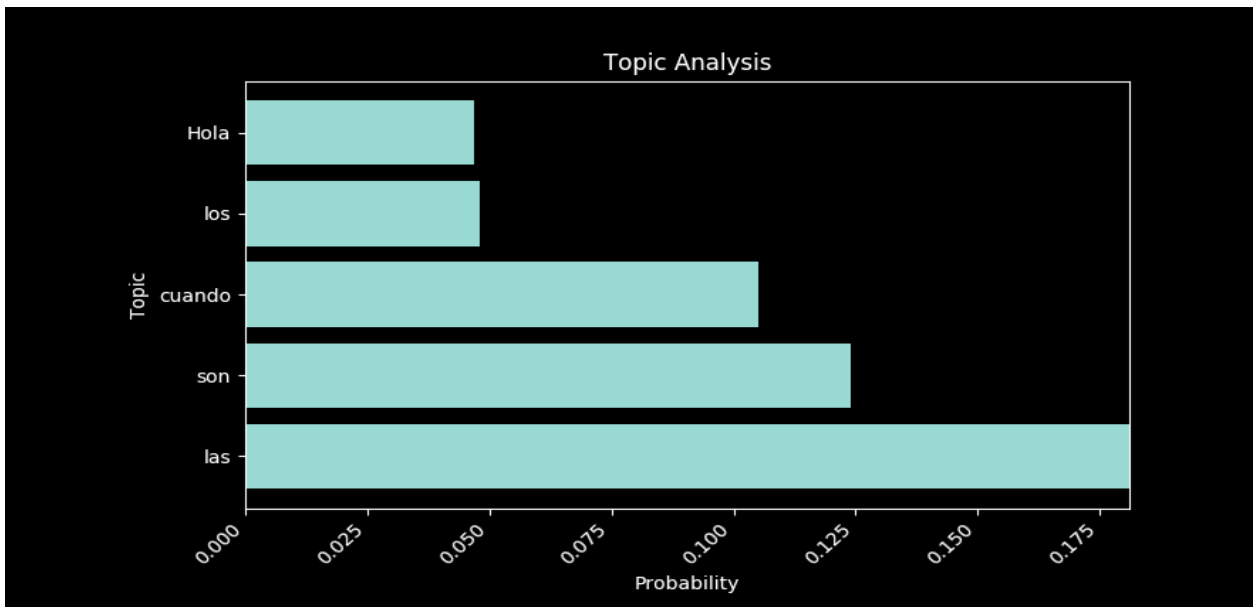


Figura 5.8: Resultado LDA con documentos con *stopwords* y sin lemas (b).

- De los mensajes con stopwords y con lemas no se pudo obtener gran información puesto que las palabras que predominan siguen siendo palabras vacías como artículos, pronombres, preposiciones, etc., a pesar de que el texto se encuentre lematizado.

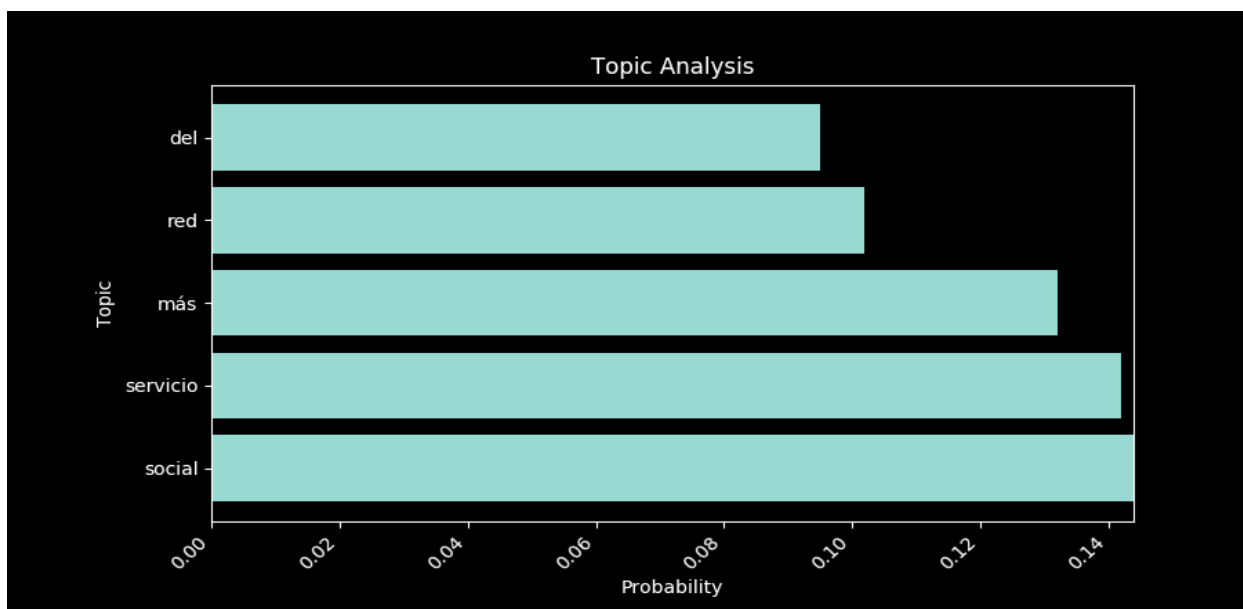


Figura 5.9: Resultado LDA con documentos con *stopwords* y con lemas (a).

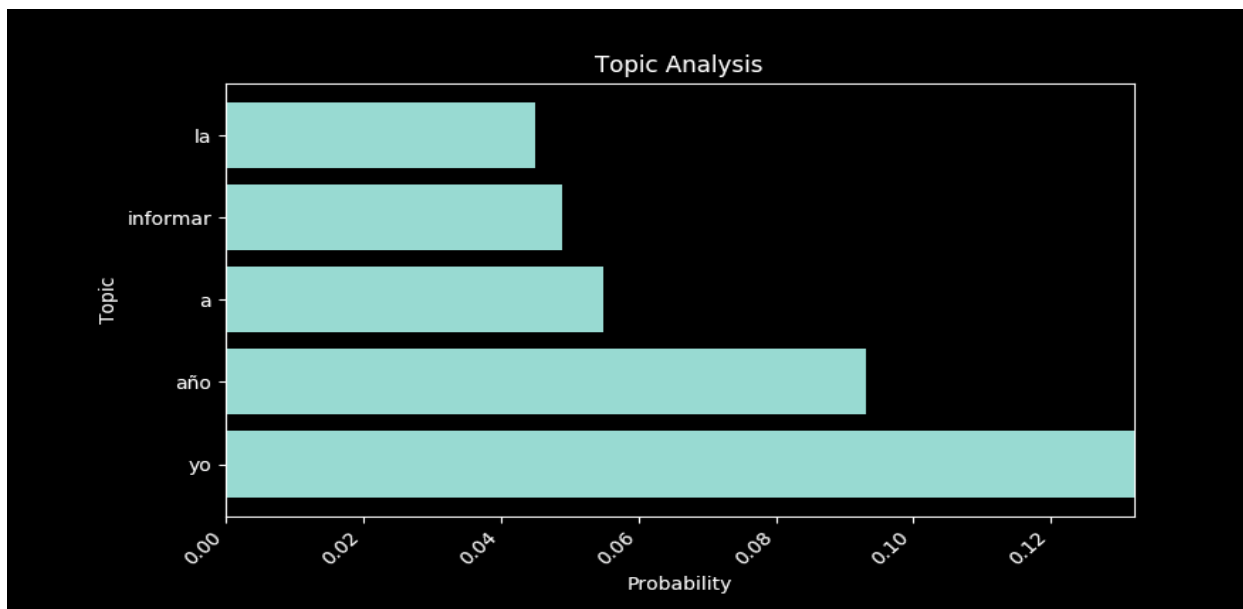


Figura 5.10: Resultado LDA con documentos con *stopwords* y con lemas (b).

- *De los mensajes sin stopwords y sin lemas los resultados obtenidos dieron lugar un avance significativo en los resultados, puesto que se pudieron encontrar temas relevantes de interés entre los estudiantes como: tramite de baja temporal, convocatorias de exámenes de admisión, inscripciones, información sobre el plantel e incluso ofertas laborales.*

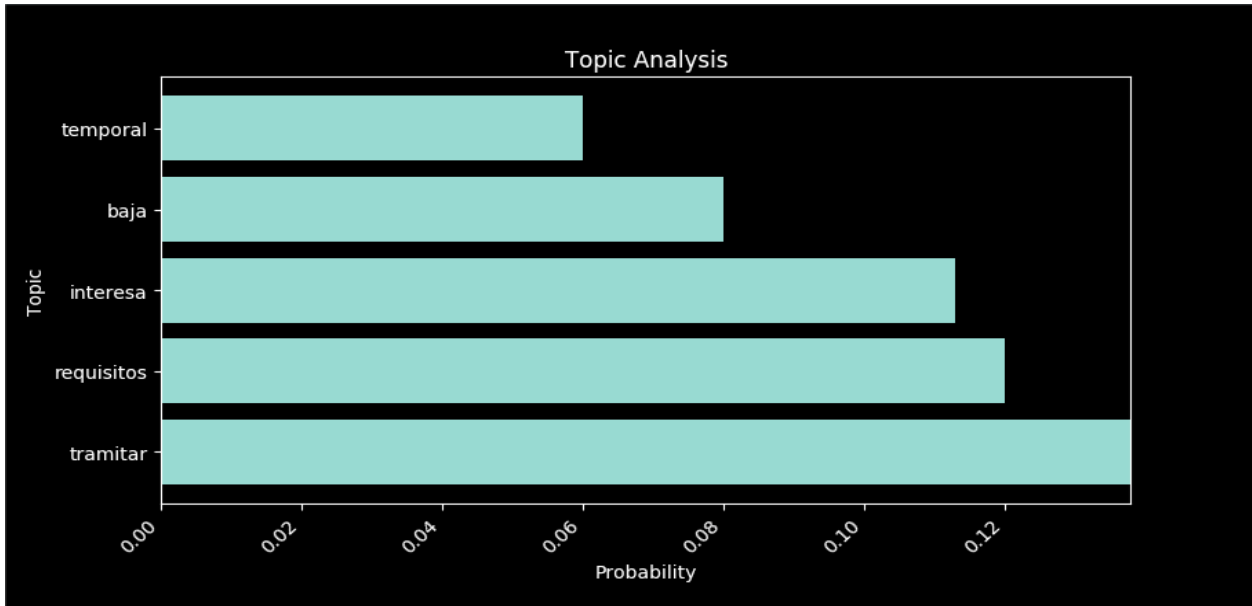


Figura 5.11: Resultado LDA con documentos sin *stopwords* y sin lemas (a).

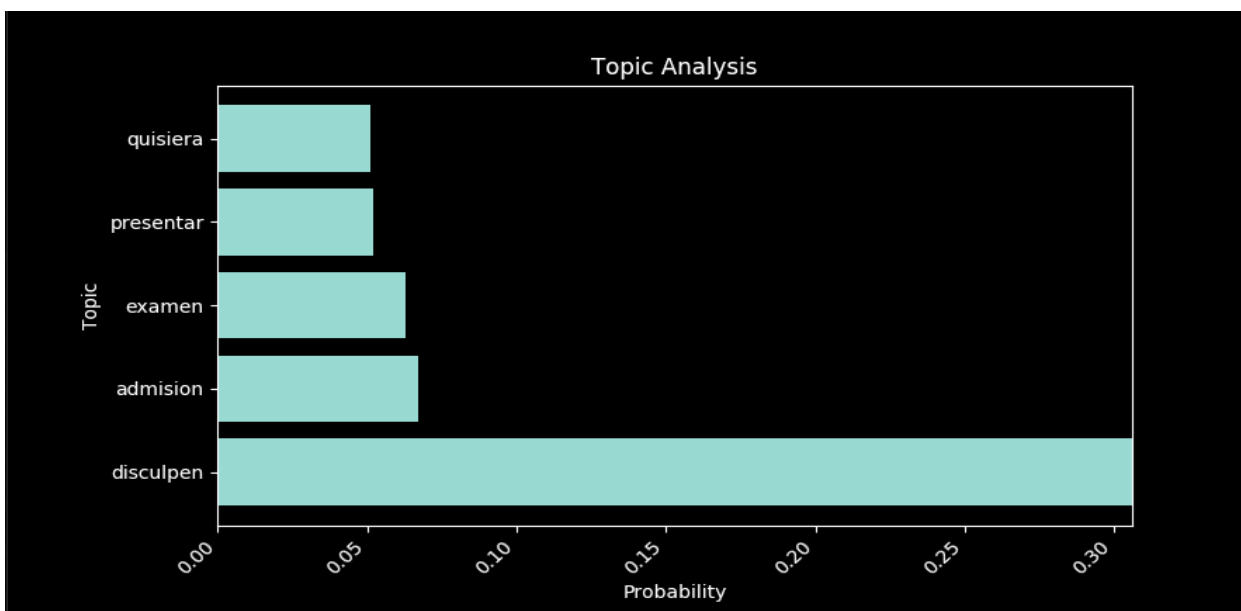


Figura 5.12: Resultado LDA con documentos sin *stopwords* y sin lemas (b).

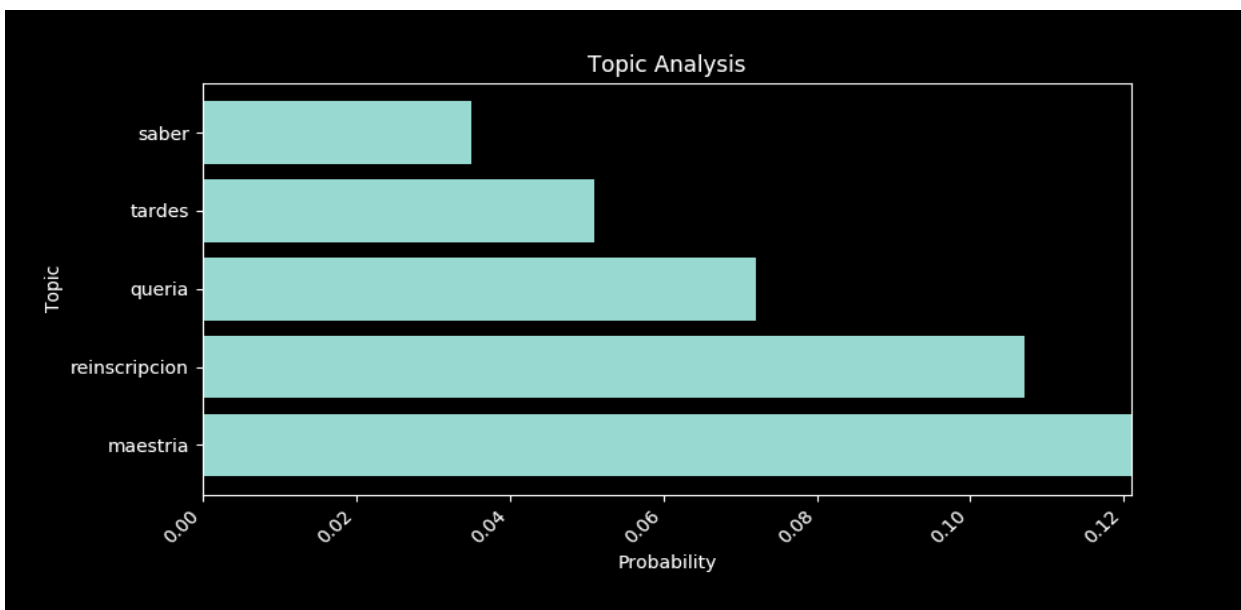


Figura 5.13: Resultado LDA con documentos sin *stopwords* y sin lemas (c).

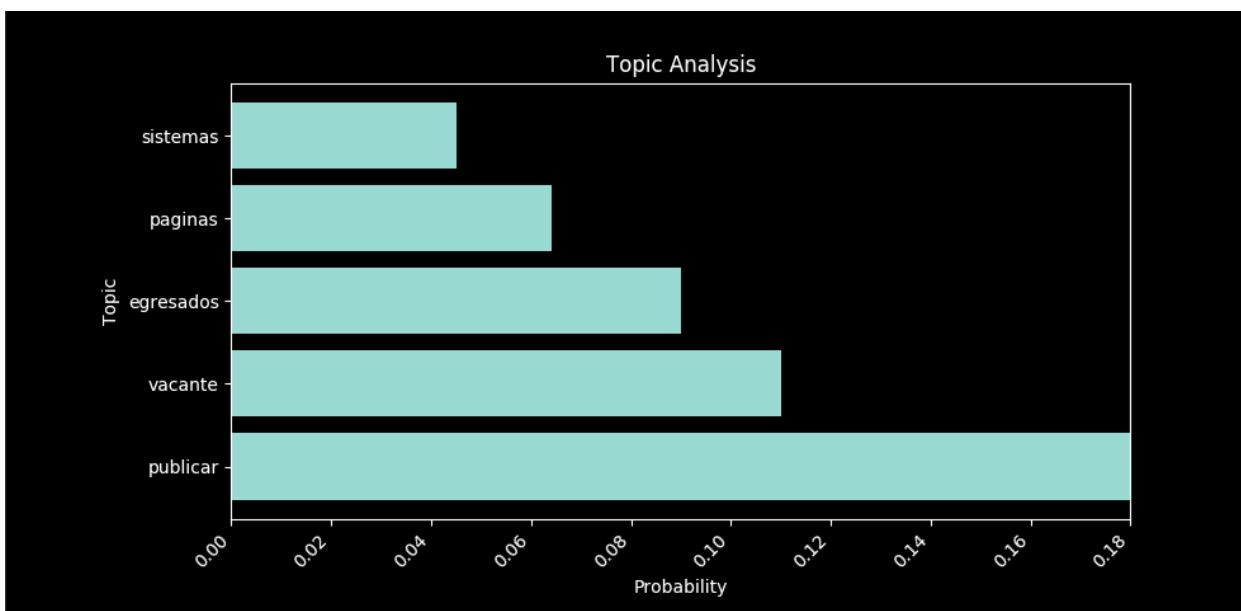


Figura 5.14: Resultado LDA con documentos sin *stopwords* y sin lemas (d).

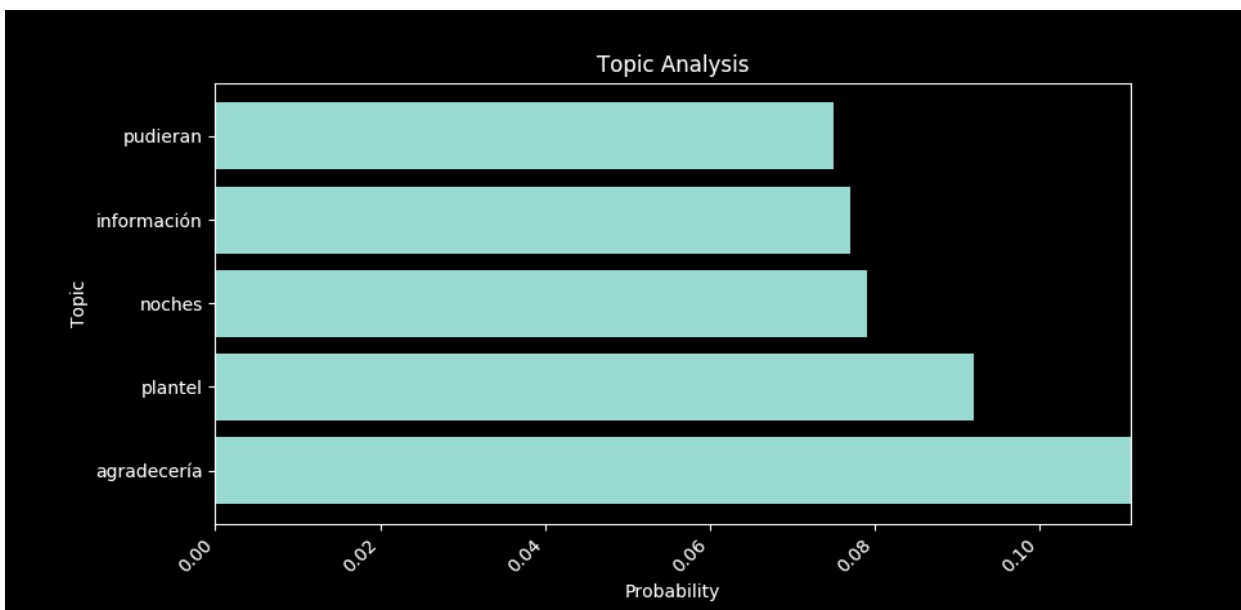


Figura 5.15: Resultado LDA con documentos sin *stopwords* y sin lemas (e).

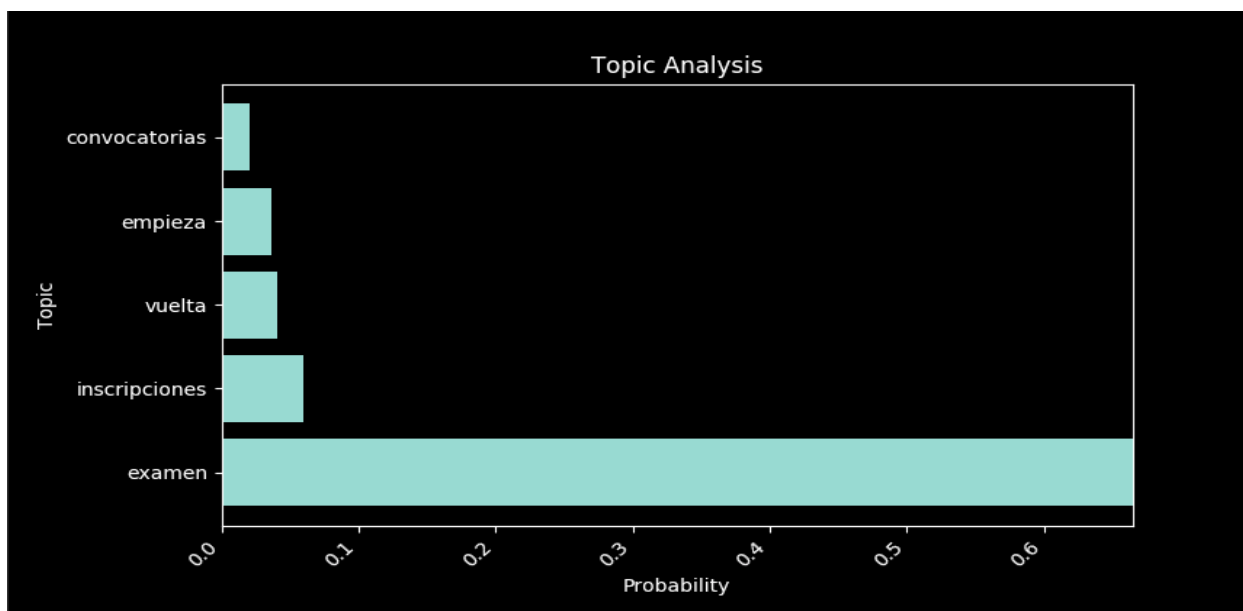


Figura 5.16: Resultado LDA con documentos sin *stopwords* y sin lemas (f).

- *De los mensajes sin stopwords y con lemas, probablemente dieron lugar a los resultados esperados ya que la mayoría de los estudiantes están interesados en los siguientes tópicos: exámenes de ingreso, cambio de carrera, convocatoria de titulación, convocatoria para inscripción a maestría, bajas temporales, taller de protocolo de trabajo terminal, servicio social, etc.*

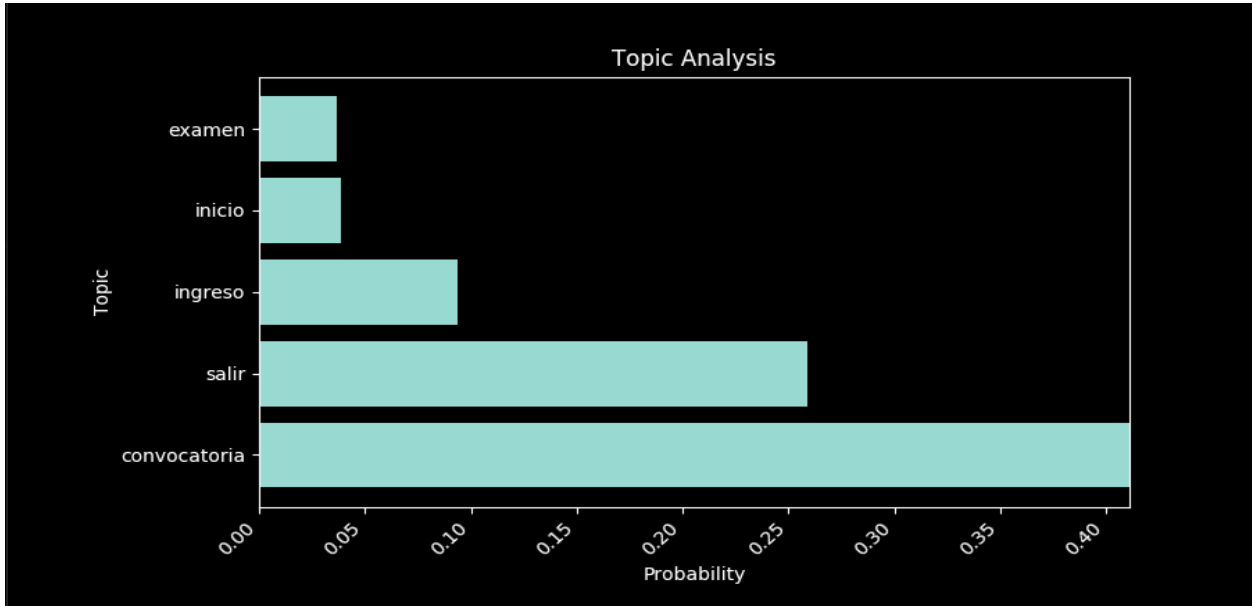


Figura 5.17: Resultado LDA con documentos sin *stopwords* y con lemas (a).

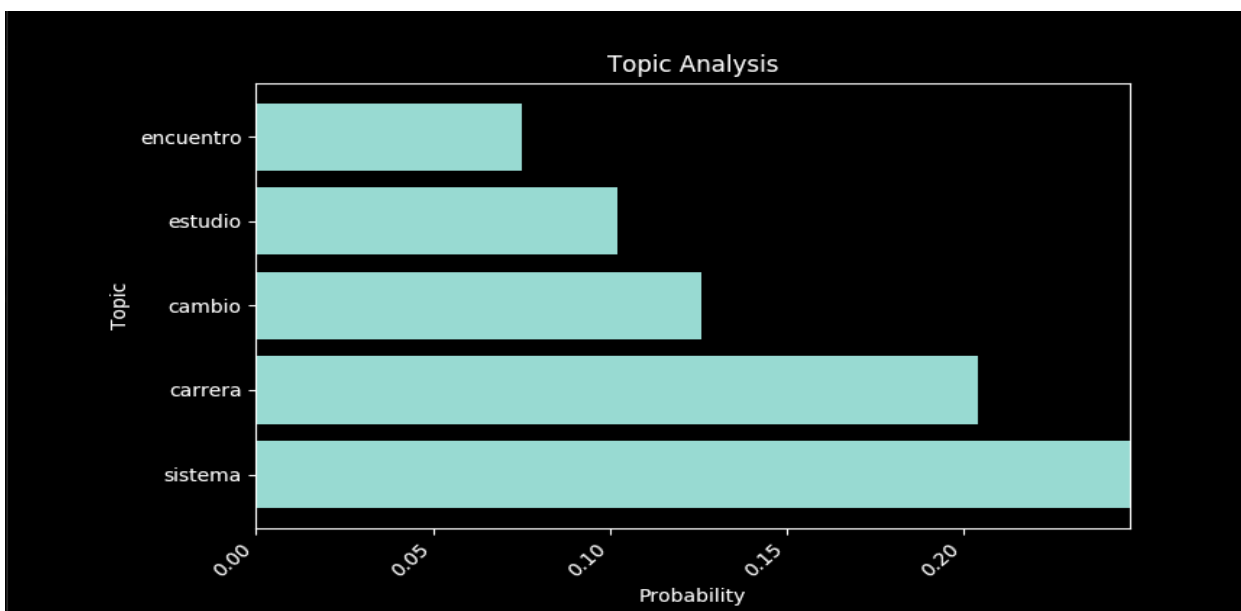


Figura 5.18: Resultado LDA con documentos sin *stopwords* y con lemas (b).

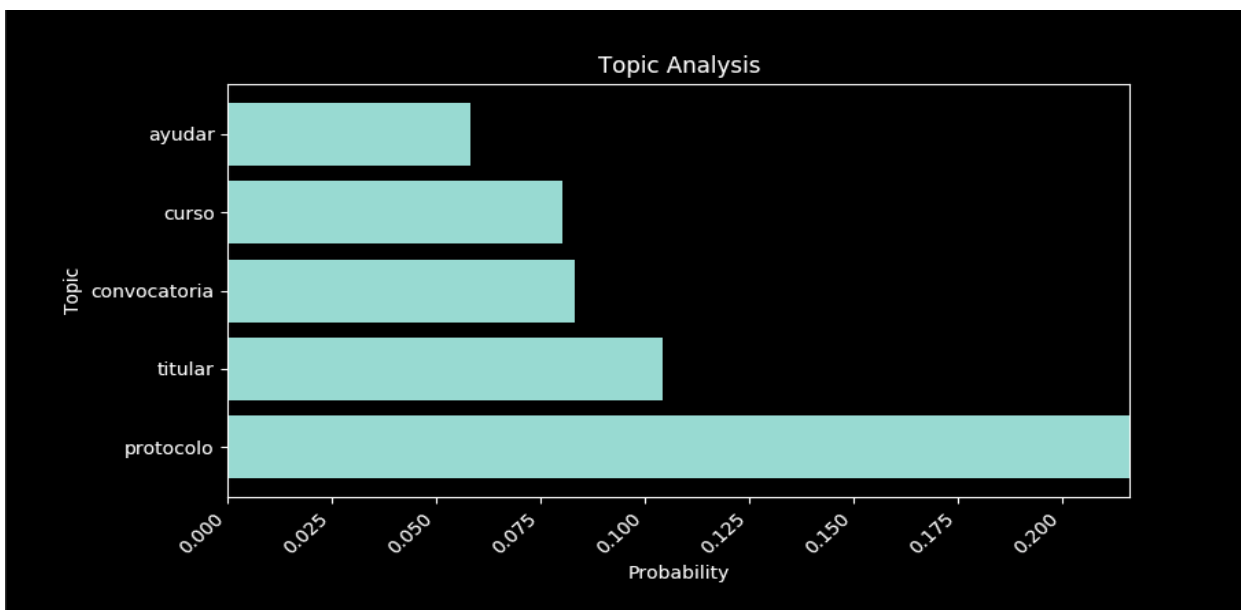


Figura 5.19: Resultado LDA con documentos sin *stopwords* y con lemas (c).

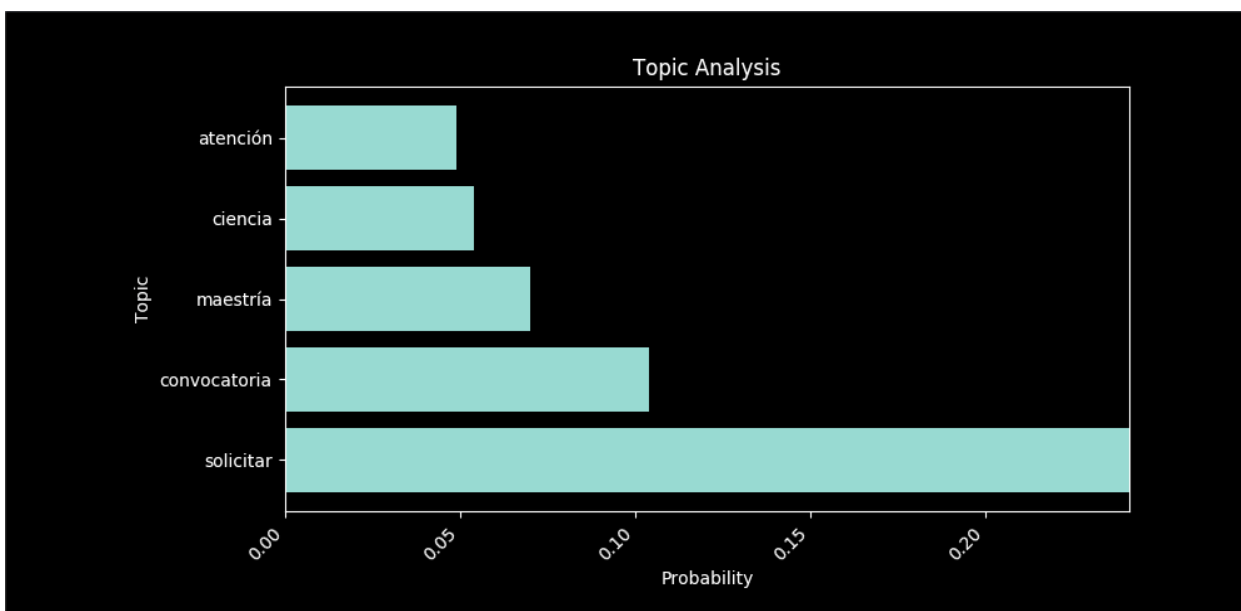


Figura 5.20: Resultado LDA con documentos sin *stopwords* y con lemas (d).

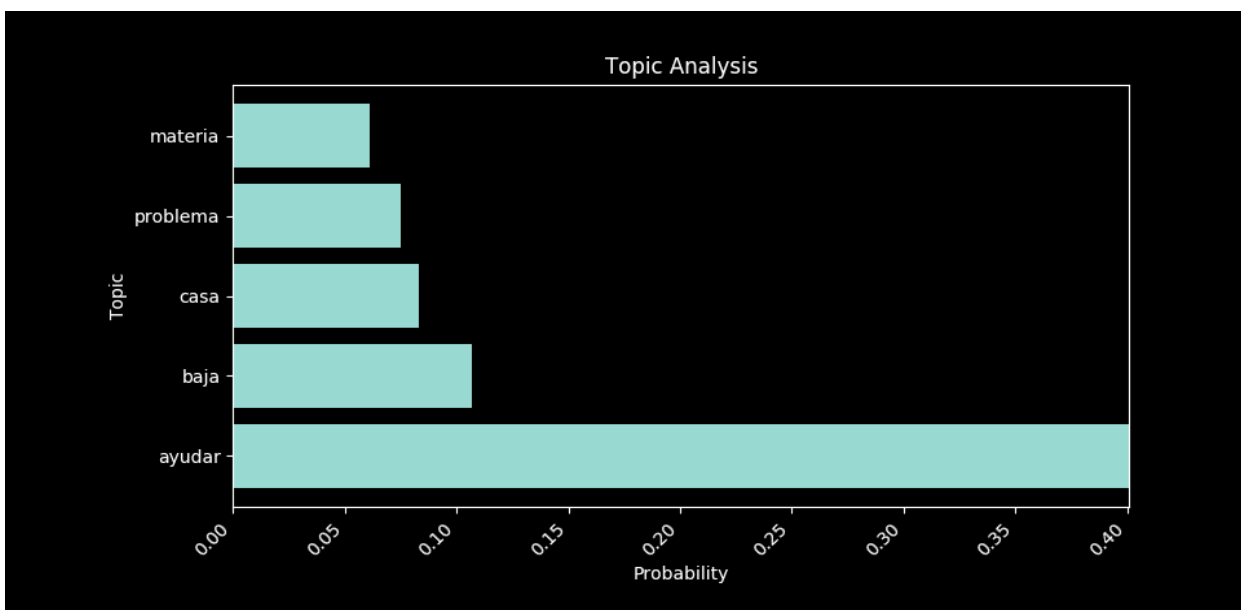


Figura 5.21: Resultado LDA con documentos sin *stopwords* y con lemas (e).

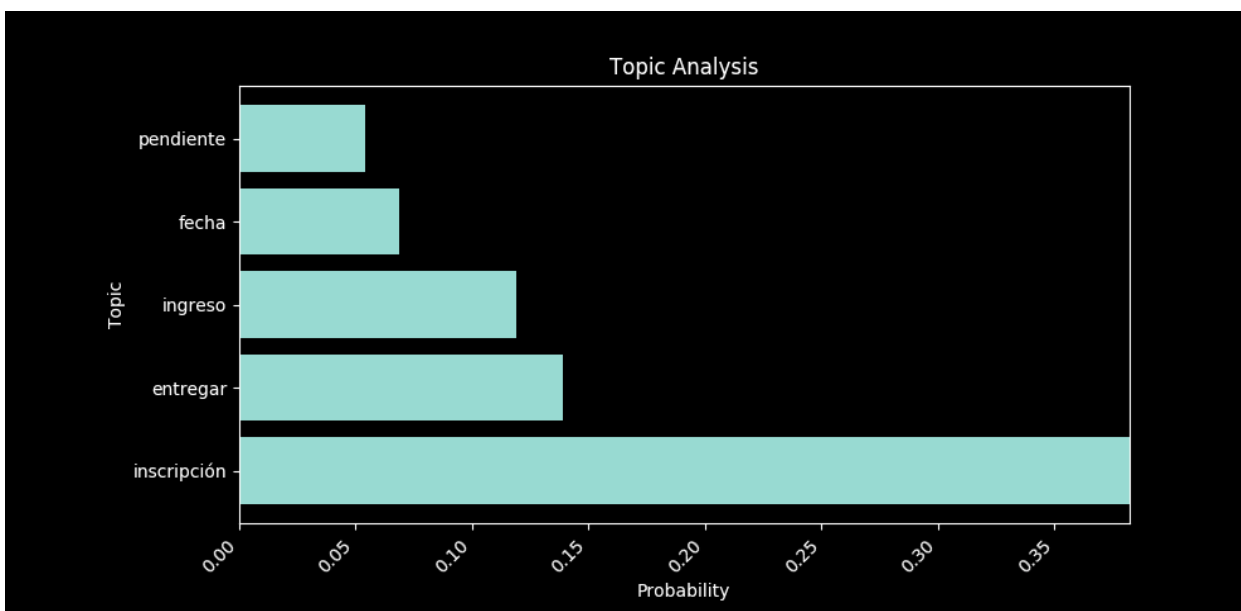


Figura 5.22: Resultado LDA con documentos sin *stopwords* y con lemas (f).

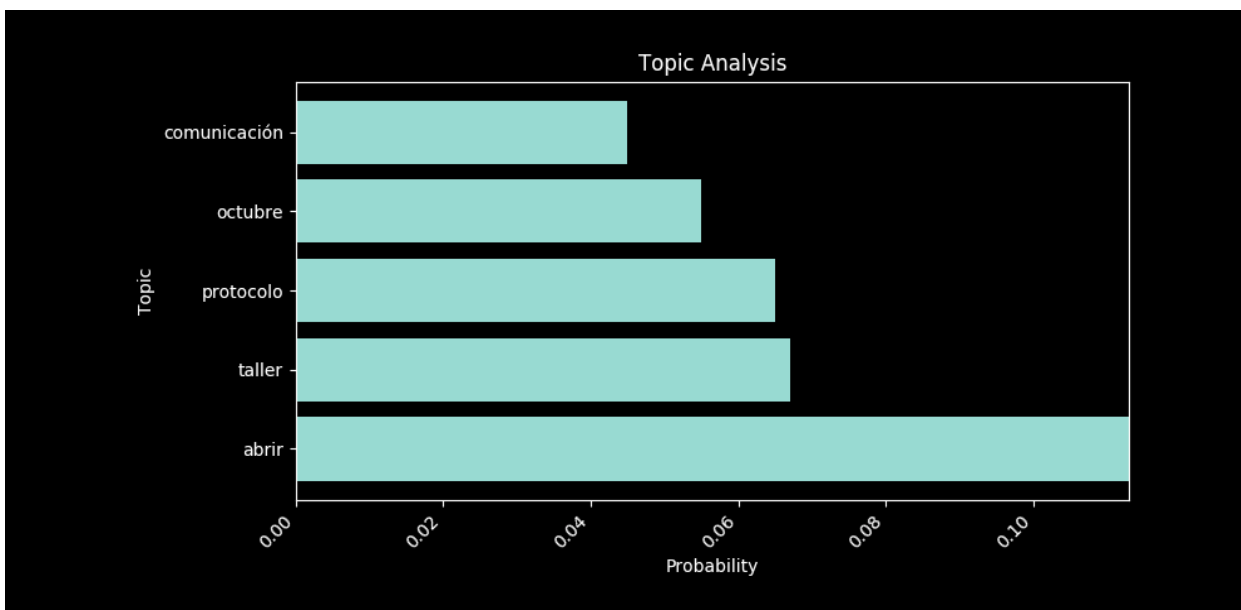


Figura 5.23: Resultado LDA con documentos sin *stopwords* y con lemas (g).

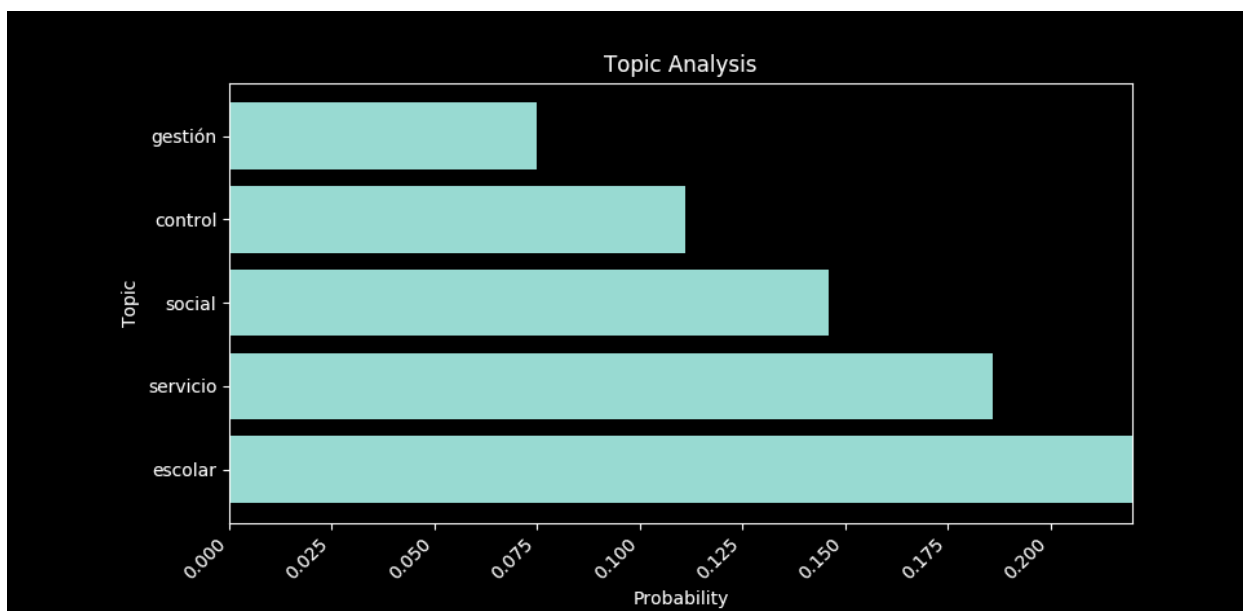


Figura 5.24: Resultado LDA con documentos sin *stopwords* y con lemas (h).

Capítulo 6

Anexos

6.1. Anexo I. Análisis de campo

6.1.1. Encuesta

Las siguientes preguntas hacen referencia a los procesos para realizar trámites del ámbito de la gestión educativa de la Escuela Superior de Cómputo.

- | | |
|---|---|
| 1. <i>¿Cómo te enteras del procedimiento para llevar a cabo tus trámites?</i> | a) <i>SI</i> |
| | b) <i>No</i> |
| a) <i>Compañeros o Amigos</i> | |
| b) <i>Página web de la escuela</i> | 3. <i>Si la respuesta fue sí, ¿Cuáles consideras que son los 3 principales impedimentos a la hora de realizar un trámite?</i> |
| c) <i>Redes sociales</i> | |
| d) <i>Semana de inducción</i> | a) <i>Falta de información</i> |
| e) <i>Atención en ventanilla</i> | b) <i>Falta de claridad en el proceso</i> |
| f) <i>Atención telefónica</i> | c) <i>Falta de documentación</i> |
| g) <i>Otro</i> | d) <i>Información no actualizada</i> |
| 2. <i>¿Has tenido problemas a la hora de realizar algún trámite?</i> | e) <i>Fuera de tiempo</i> |
| | f) <i>Otro</i> |

4. *¿Requeriste más de un medio para poder obtener la información que necesitabas?*

a) *SI*

b) *No*

5. *¿Alguna vez has intentado pedir asesoría para realizar un trámite por medio de internet?*

a) *SI*

b) *No*

Si la respuesta fue sí, contesta las preguntas 6-9:

6. *¿En qué medio buscaste la asesoría?*

a) *Página de facebook*

b) *Messenger de facebook*

c) *Twitter*

d) *Correo electrónico*

e) *Otro*

7. *¿En cuanto tiempo te obtuviste la asesoría?*

a) *En menos de una hora*

b) *Ese mismo día más tarde*

c) *Días después*

d) *No obtuve una respuesta*

8. *La información que te proveyeron fue:
(Marca las necesarias)*

a) *Completa*

b) *Oportuna*

c) *Relevante*

d) *Accesible*

e) *Entendible*

f) *Clara*

g) *Precisa*

h) *Satisfactoria*

9. *¿Como fue la experiencia a la hora de solicitar la información?*

a) *Muy satisfactoria*

b) *Buena*

c) *Regular*

d) *Mala*

10. Si las respuestas a través de Facebook Messenger a tus dudas sobre distintos trámites fueran contestadas con prontitud. ¿Qué prioridad de uso le darías a este servicio?

- a) Seria mi primera opción
- b) Estaría entre mis opciones
- c) No lo usaría

La propuesta de solución a este problema es la creación de un prototipo de agente conversacional popularmente conocido como chatbot proveído a través del aplicativo de facebook para mensajería instantánea “Messenger”, este estará enfocado a la resolución de preguntas frecuentes formuladas por el alumnado de ESCOM.

La información recabada en esta encuesta nos será de gran utilidad, por parte de los integrantes y directores del trabajo terminal 2019-A087 le damos las gracias por completar esta encuesta.

6.1.2. Resultados

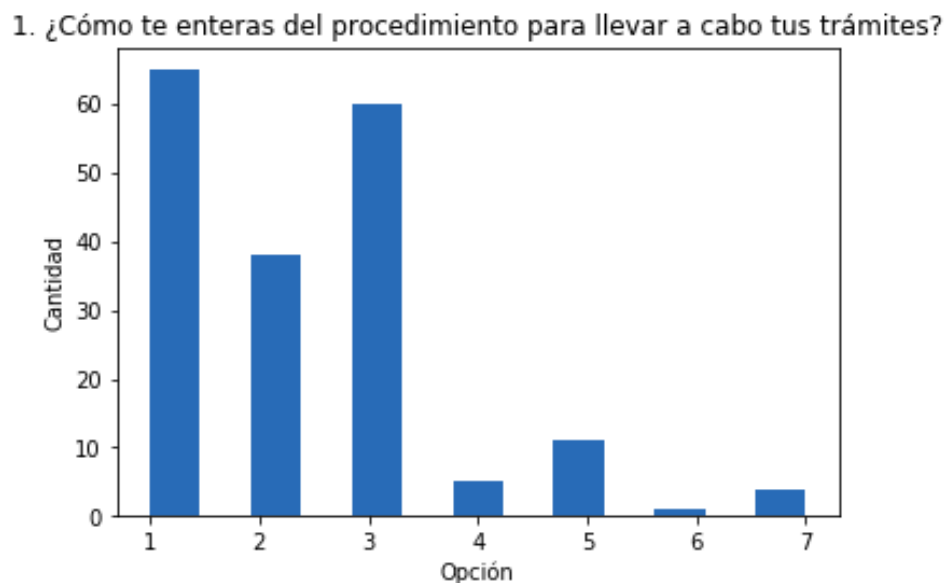
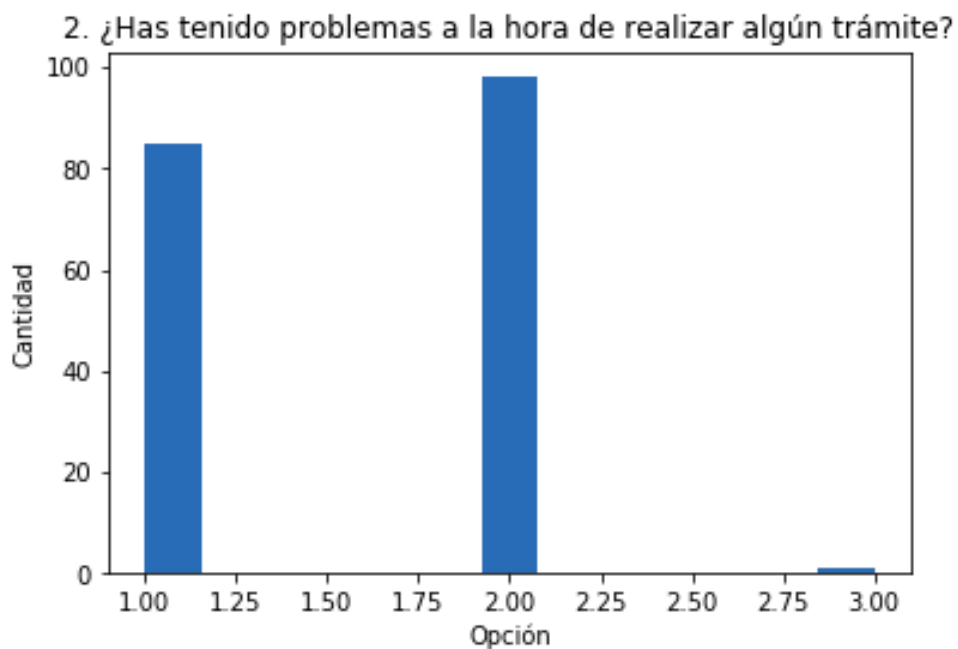
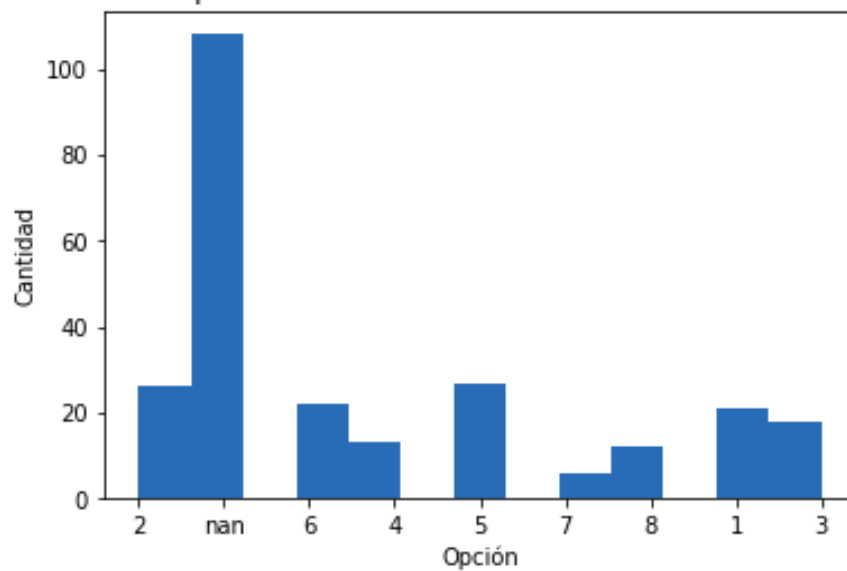
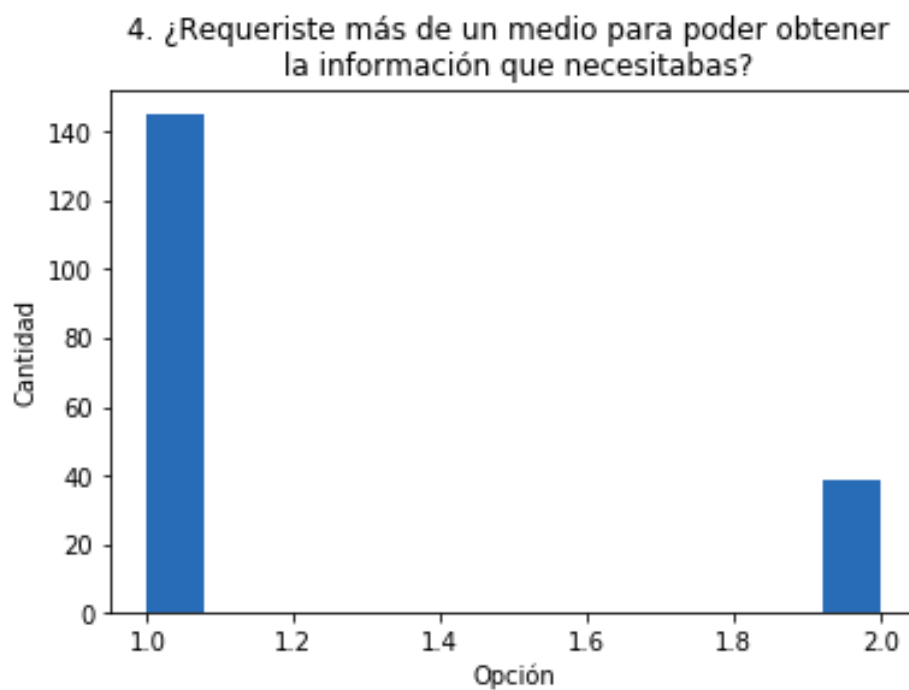
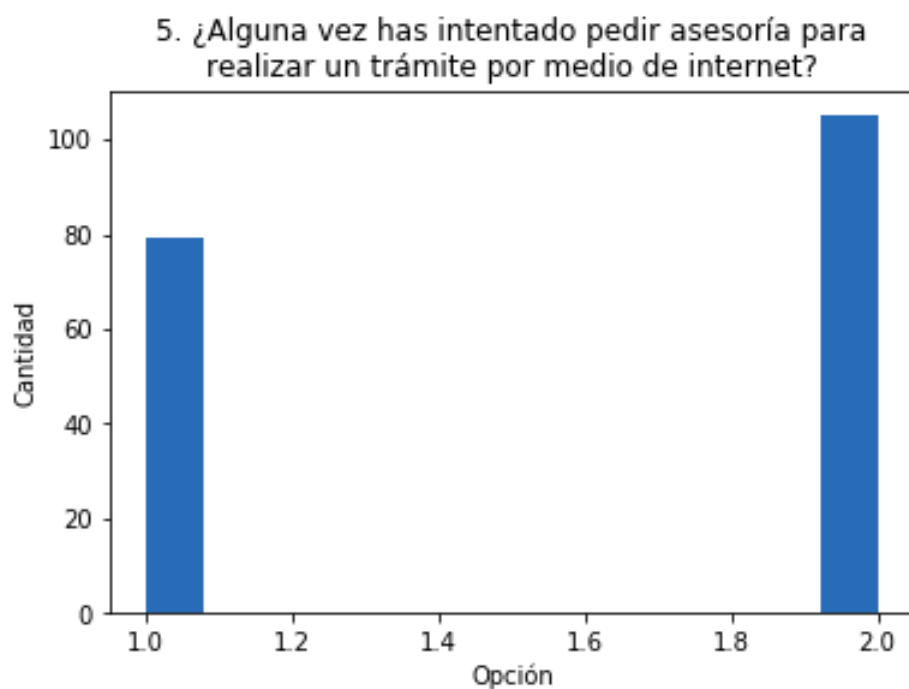


Figura 6.1: Gráfico de barras: Pregunta 1

**Figura 6.2:** Gráfico de barras: Pregunta 2

3. Si la respuesta fue sí, ¿Cuáles consideras que son los 3 principales impedimentos a la hora de realizar un trámite?

**Figura 6.3:** Gráfico de barras: Pregunta 3

**Figura 6.4:** Gráfico de barras: Pregunta 4**Figura 6.5:** Gráfico de barras: Pregunta 5

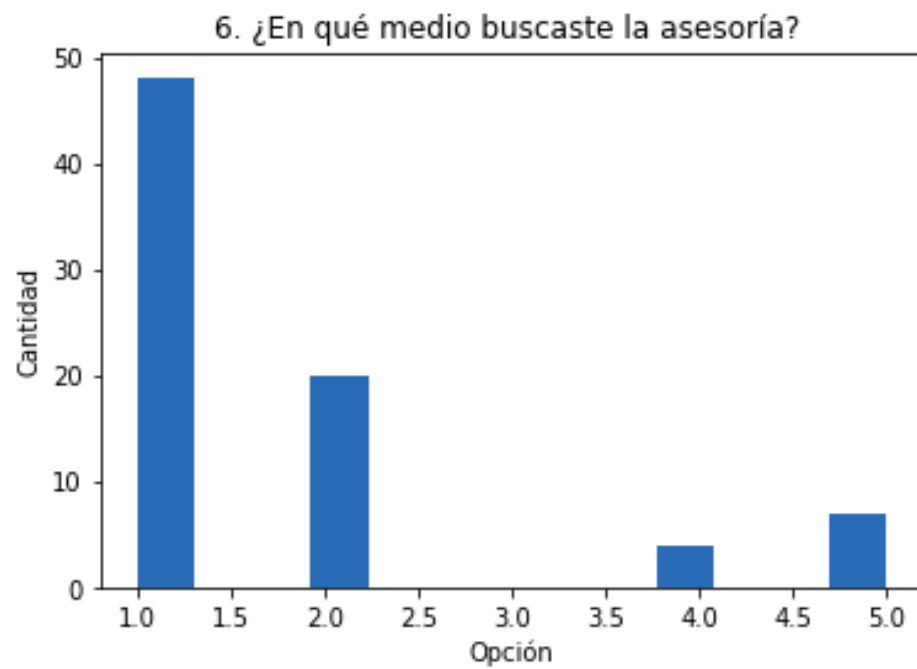


Figura 6.6: Gráfico de barras: Pregunta 6

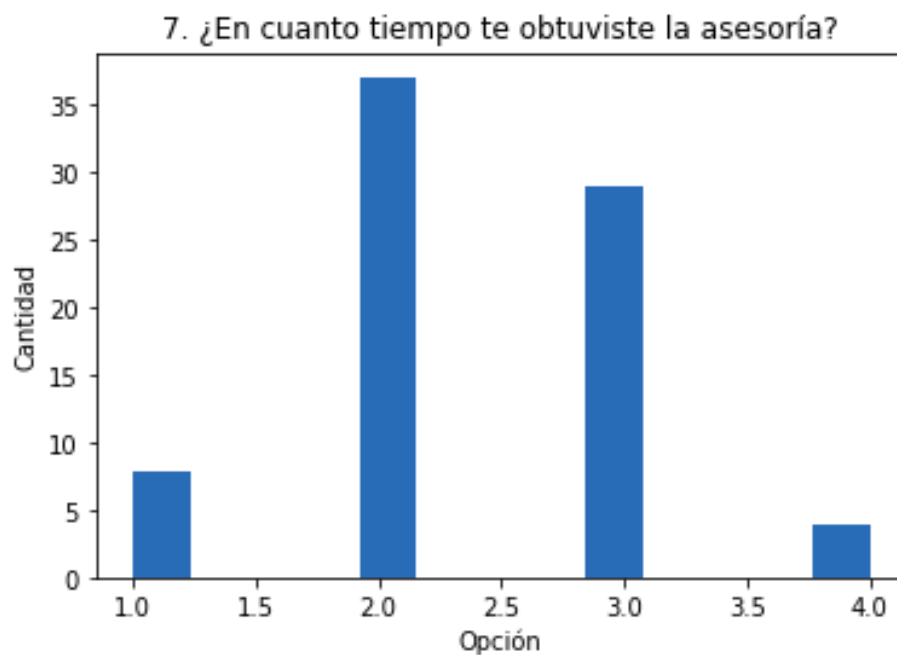
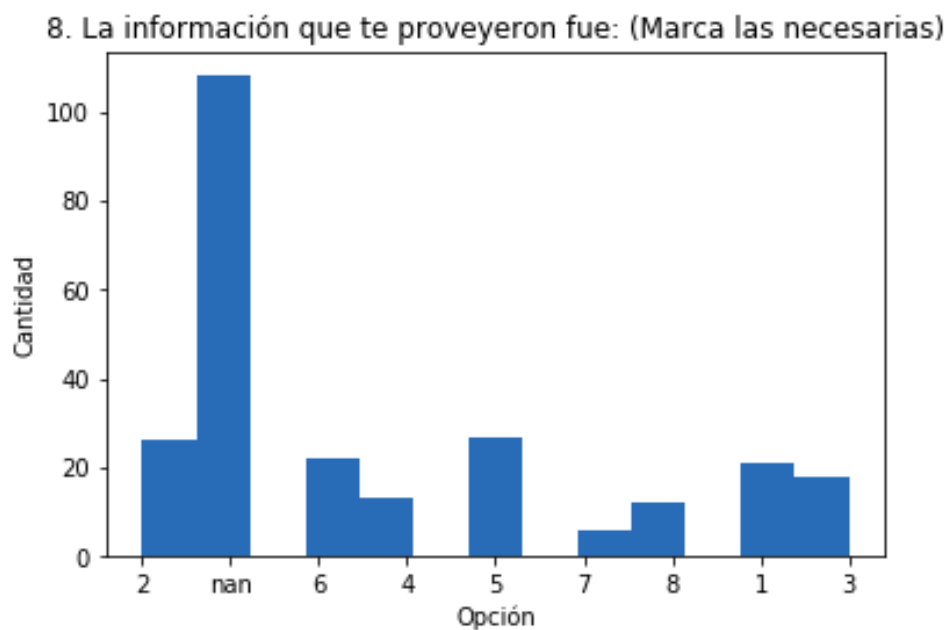
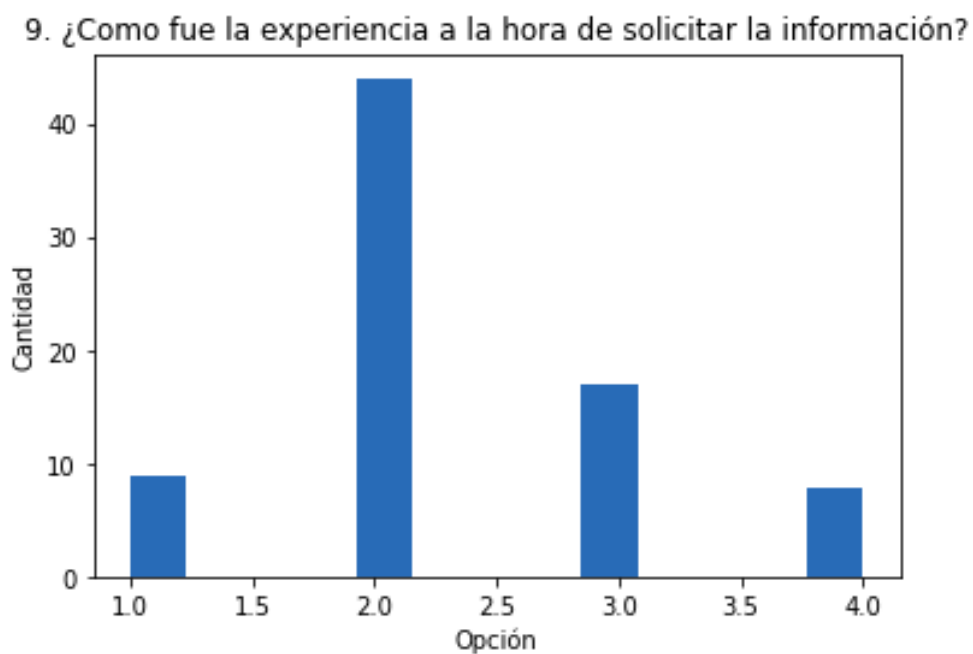


Figura 6.7: Gráfico de barras: Pregunta 7

**Figura 6.8:** Gráfico de barras: Pregunta 8**Figura 6.9:** Gráfico de barras: Pregunta 9

10. Si las respuestas a través de Facebook Messenger a tus dudas sobre distintos trámites fueran contestadas con prontitud. ¿Qué prioridad de uso le darías a este servicio?

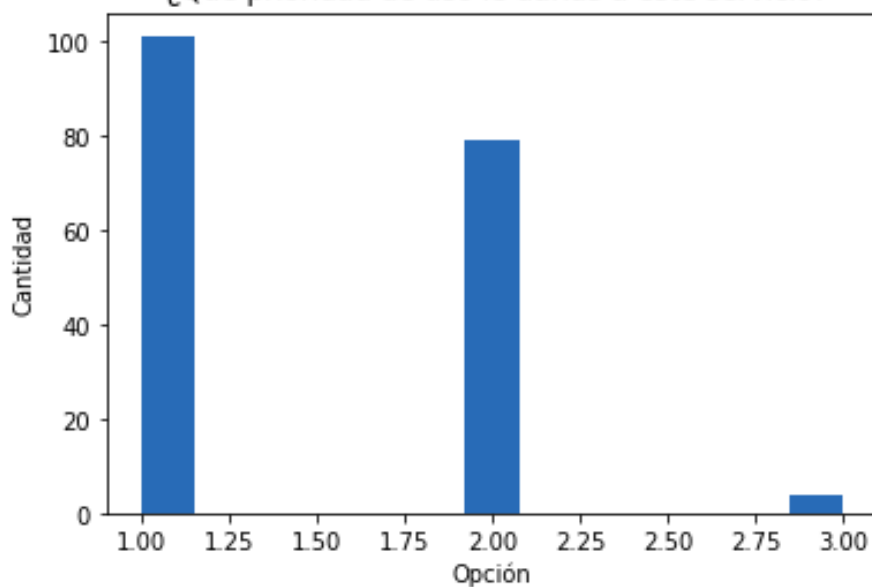


Figura 6.10: Gráfico de barras: Pregunta 10

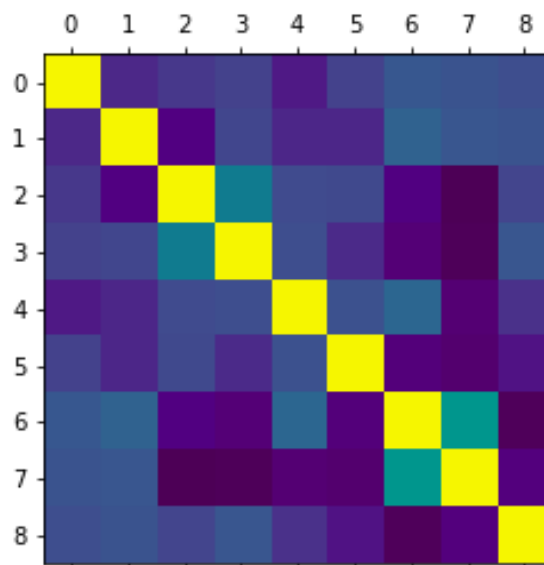


Figura 6.11: Mapa de correlación de variables ente las preguntas de la encuesta

6.2. Anexo II. Entrevista

6.3. Anexo III. Métricas del ajuste de la completitud técnica

1. Comunicación de Datos

0: *Aplicación es batch exclusivamente*

1-2 *Impresión o entrada de datos remota*

3-5 *Teleproceso (TP) interactivo*

3 *TP interface a un proceso batch*

5 *La aplicación se interactiva*

2. Función Distribuída. "Distribuída" significa que los componentes de la aplicación están distribuídos en dos o más procesadores diferentes.

0: *La aplicación no ayuda a la transferencia de datos o a la función de procesamiento entre los componentes del sistema*

1 *La aplicación prepara datos para el usuario final de otro procesador*

2-3 *Los datos se preparan para transferencia, se transfieren y se procesan en otro componente del sistema*

4 *Igual que 2-3, pero con realimentación al sistema inicial*

5 *Las funciones de procesamiento se realizan dinámicamente en el componente más apropiado del sistema.*

3. Rendimiento (referido a la importancia de respuesta dentro de todo el sistema)

0, 3: *Análisis y diseño de las consideraciones del rendimiento son estándar. No se requieren requerimientos especiales por parte del usuario*

4: *En la fase de diseño se incluyen tareas del análisis del rendimiento para cumplir los requerimientos del usuario*

5: *Además se utilizan herramientas de análisis del rendimiento en el diseño, desarrollo e instalación*

4. *Configuración utilizada masivamente (referente a la importancia del entorno)*

0, 3: *La aplicación corre en una máquina estándar sin restricciones de operación*

4: *Restricciones de operación requieren características específicas de la aplicación en el procesador central*

5: *Además hay restricciones específicas a la aplicación en los componentes distribuidos del sistema.*

5. *Tasas de Transacción (una alta llegada de transacciones provoca problemas más allá de los de la característica 3)*

0, 3: *Las tasas son tales que las consideraciones de análisis de rendimiento son estándares*

4: *En la fase de diseño se incluyen tareas de análisis de rendimiento para verificar las altas tasas de transacciones*

5: *Además se utilizan herramientas de análisis del rendimiento.*

6. *Entrada On-Line de datos*

0, 2: *Hasta el 15% de las transacciones tienen entrada interactiva*

3, 4: *15% al 30% tienen entrada interactiva*

5: *30% al 50% tienen entrada interactiva.*

7. *Diseño para la eficiencia de usuario final*

0: *Sistema batch*

1, 3: *No se especifican requerimientos especiales*

4: *Se incluyen tareas de diseño para la consideración de factores humanos*

5: *Además se utilizan herramientas especiales o de prototipado para promover la eficiencia.*

8. *Actualización On-Line*

0: *Nada*

1, 2: *Actualización on line de los ficheros de control. El volumen de actualización es bajo y la recuperación fácil.*

3: *Actualización on line de la mayoría de los ficheros internos lógicos*

4: *Además es esencial la protección contra la pérdida de datos*

5: *Además se considera el coste de recuperación de volúmenes elevados.*

9. *Complejidad del procesamiento (esto es complejidad interna más allá de las convenciones de cuenta de entidades de MkII FPA). ¿Qué características tiene la aplicación?*

- *mucho procesamiento matemático y/o lógico*
- *muchas excepciones de procesamiento, muchas transacciones incompletas y mucho reprocesamiento de las transacciones*
- *procesamiento de seguridad y/o control sensitivo*

0: *No se aplica nada de esto*

1, 3: *Se aplica alguna cosa*

4: *Se aplican dos cosas*

5: *Se aplica todo.*

10. Utilizable en otras aplicaciones (el código se diseña para que sea compartido o utilizable por otras aplicaciones -no confundir con 13).

0, 1: *Una aplicación local que responde a las necesidades de una organización usuaria*

2, 3: *La aplicación utiliza o produce módulos comunes que consideran más necesidades que las del usuario*

4, 5: *Además, la aplicación se empaquetó y documentó con el propósito de fácil reutilización*

11. Facilidad de Instalación

0, 1: *No se requieren por parte del usuario facilidades especiales de conversión e instalación*

2, 3: *Los requerimientos de conversión e instalación fueron descritos por el usuario y se proporcionaron guías de conversión e instalación*

4, 5: *Además se proporcionaron y probaron herramientas de conversión e instalación*

12. Facilidad de Operación

0: *No se especifican por parte del usuario consideraciones específicas de operación*

1, 2: *Se requieren, proporcionan y prueban procesos específicos de arranque, backup y recuperación*

3, 4: *Además la aplicación minimiza la necesidad de actividades manuales, tales como instalación de cintas y papel*

5: *La aplicación se diseña para operación sin atención*

13. Puestos Múltiples. Añadir puntos por cada uno de los siguientes factores

0: *El usuario no requiere la consideración de más de un puesto*

1: *De uno a cuatro puestos*

2: *Cinco o más puestos*

1: *Se proporciona documentación y plan de apoyo para soportar la aplicación en varios lugares*

2: *Los puestos están en países diferentes*

14. Facilidad de Cambio (esfuerzo específico de diseño para facilitar cambios futuros). Añadir puntos por cada uno de los siguientes factores

0: *No hay requerimientos especiales del usuario para minimizar o facilitar el cambio*

1, 3: *Se proporciona capacidad de consulta flexible*

4, 5: *Datos importantes de control se mantienen en tablas que son actualizadas por el usuario a través de procesos on-line interactivos.*

15. Requerimientos de otras Aplicaciones

0: *El sistema es stand-alone*

1, 4: *Requerimientos del sistema para interfaces o compartición de datos deben ser sincronizados con otras aplicaciones*

5: *Se deben sincronizar los requerimientos del sistema con cuatro o más aplicaciones*

16. Seguridad, Privacidad y Auditoría. Añadir puntos por cada uno de los siguientes factores

1: *Si el sistema debe cumplir requerimientos personales (incluso legales) de privacidad*

1: *Si el sistema debe cumplir requerimientos especiales de auditoría*

1, 2: *Si el sistema debe cumplir requerimientos excepcionales de seguridad para prevenir pérdidas de naturaleza financiera o militar*

1: *Si se requiere el criptográfico de los datos de las comunicaciones*

17. Necesidad de Adiestramiento al Usuario

0: *Si no es necesario material ni cursos*

1, 4: *Si se requiere entrenamiento especial o se proporcionan facilidades de ayuda on-line*

5: *Se requiere un sistema separado de entrenamiento o simulador*

18. Uso directo de otras empresas

0: *No existe otra empresa conectada al sistema*

1: *Los datos se envían o reciben de empresas conocidas*

2: *Empresas conocidas están conectadas al sistema en modo de lectura solamente*

3, 4: *Empresas conocidas están conectadas directamente al sistema con capacidad de actualización on-line*

5: *Empresas desconocidas (público en general o un grupo demasiado extenso como para ser adiestrado) pueden acceder al sistema*

19. Documentación. Contar 1 por cada tipo de documento listado a continuación que se entregue al final del proyecto.

- *Especificación Funcional del Sistema (datos y procesos)*
- *Especificación Técnica del Sistema*
- *Documentación del programa (al menos diagramas de flujo)*
- *Librería de Elementos de Datos*

- *Elemento de Datos/ Registro/ X-referencia del programa*
- *Manual de usuario*
- *Folleto de información general del sistema*
- *Librería de datos de prueba*
- *Material de curso de adiestramiento al usuario*
- *Documentos de coste/beneficio del sistema*
- *Informe de petición de cambios y errores*

Se calcula el grado de influencia utilizando la siguiente tabla

- a) *si 0-2 tipos de documento*
- b) *si 3-4*
- c) *si 5-6*
- d) *si 7-8*
- e) *si 9-10*
- f) *si 11-12 tipos de documento*